

A

Abrupt Climate Change Modeling

GERRIT LOHMANN

Alfred Wegener Institute for Polar and Marine Research,
Bremerhaven, Germany

Article Outline

[Glossary](#)

[Definition of the Subject](#)

[Introduction](#)

[A Mathematical Definition](#)

[Earth System Modeling and Analysis](#)

[Example: Glacial-Interglacial Transitions](#)

[Example: Cenozoic Climate Cooling](#)

[Examples: Transient Growth](#)

[Future Directions](#)

[Bibliography](#)

Glossary

Atmosphere The atmosphere is involved in many processes of abrupt climate change, providing a strong non-linearity in the climate system and propagating the influence of any climate forcing from one part of the globe to another. Atmospheric temperature, composition, humidity, cloudiness, and wind determine the Earth's energy fluxes. Wind affects the ocean's surface circulation and upwelling patterns. Atmospheric moisture transport determines the freshwater balance for the oceans, overall water circulation, and the dynamics of glaciers.

Oceans Because water has enormous heat capacity, oceans typically store 10–100 times more heat than equivalent land surfaces. The oceans exert a profound influence on climate through their ability to transport heat from one location to another. Changes in ocean circulation have been implicated in abrupt climate change of the past. Deglacial meltwater has fresh-

ened the North Atlantic and reduced the ability of the water to sink, inducing long-term coolings.

Land surface The reflective capacity of the land can change greatly, with snow or ice sheets reflecting up to 90% of the sunlight while dense forests absorb more than 90%. Changes in surface characteristics can also affect solar heating, cloud formation, rainfall, and surface-water flow to the oceans, thus feeding back strongly on climate.

Cryosphere The portion of the Earth covered with ice and snow, the cryosphere, greatly affects temperature. When sea ice forms, it increases the planetary reflective capacity, thereby enhancing cooling. Sea ice also insulates the atmosphere from the relatively warm ocean, allowing winter air temperatures to steeply decline and reduce the supply of moisture to the atmosphere. Glaciers and snow cover on land can also provide abrupt-change mechanisms. The water frozen in a glacier can melt if warmed sufficiently, leading to possibly rapid discharge, with consequent effects on sea level and ocean circulation. Meanwhile, snow-covered lands of all types maintain cold conditions because of their high reflectivity and because surface temperatures cannot rise above freezing until the snow completely melts.

External factors Phenomena external to the climate system can also be agents of abrupt climate change. For example, the orbital parameters of the Earth vary over time, affecting the latitudinal distribution of solar energy. Furthermore, fluctuations in solar output, prompted by sunspot activity or the effects of solar wind, as well as volcanoes may cause climate fluctuations.

Climate time scales The climate system is a composite system consisting of five major interactive components: the atmosphere, the hydrosphere, including the oceans, the cryosphere, the lithosphere, and the biosphere. All subsystems are open and non-isolated, as the atmosphere, hydrosphere, cryosphere and biosphere act as cascading systems linked by complex

feedback processes. Climate refers to the average conditions in the Earth system that generally occur over periods of time, usually several decades or longer. This time scale is longer than the typical response time of the atmosphere. Parts of the other components of the Earth system (ice, ocean, continents) have much slower response times (decadal to millennial).

Climate variables and forcing State variables are temperature, rainfall, wind, ocean currents, and many other variables in the Earth system. In our notation, the variables are described by a finite set of real variables in a vector $x(t) \in \mathbb{R}^n$. The climate system is subject to two main external forcings $F(x, t)$ that condition its behavior, solar radiation and the action of gravity. Since $F(x, t)$ has usually a spatial dependence, F is also a vector $\in \mathbb{R}^n$. Solar radiation must be regarded as the primary forcing mechanism, as it provides almost all the energy that drives the climate system. The whole climate system can be regarded as continuously evolving, as solar radiation changes on diurnal, seasonal and longer time scales, with parts of the system leading or lagging in time. Therefore, the subsystems of the climate system are not always in equilibrium with each other. Indeed, the climate system is a dissipative, highly non-linear system, with many instabilities.

Climate models are based on balances of energy, momentum, and mass, as well as radiation laws. There are several model categories, full circulation models, low-order models, and models of intermediate complexity. Climate models simulate the interactions of the atmosphere, oceans, land surface, and ice. They are used for a variety of purposes from study of the dynamics of the weather and climate system, past climate to projections of future climate.

Global climate models or General circulation models

(GCMs) The balances of energy, momentum, and mass are formulated in the framework of fluid dynamics on the rotating Earth. GCMs discretize the equations for fluid motion and energy transfer and integrate these forward in time. They also contain parametrization for processes – such as convection – that occur on scales too small to be resolved directly. The dimension of the state vector is in the order of $n \sim 10^5 - 10^8$ depending on the resolution and complexity of the model.

Model categories In addition to complex numerical climate models, it can be of great utility to reduce the system to low-order, box, and conceptual models. This complementary approach has been successfully applied to a number of questions regarding feedback mechanisms and the basic dynamical behavior,

e.g. [48,84]. In some cases, e.g. the stochastic climate model of Hasselmann [32], such models can provide a null hypothesis for the complex system. The transition from highly complex dynamical equations to a low-order description of climate is an important topic of research. In his book “Dynamical Paleoclimatology”, Saltzman [77] formulated a dynamical system approach in order to differentiate between fast-response and slow-response variables. As an alternative to this method, one can try to derive phenomenologically based concepts of climate variability, e.g. [21,43]. In between the comprehensive models and conceptual models, a wide class of “models of intermediate complexity” were defined [12].

Earth-system models of intermediate complexity

(EMICs) Depending on the nature of questions asked and the pertinent time scales, different types of models are used. There are, on the one extreme, conceptual models, and, on the other extreme, comprehensive models (GCMs) operating at a high spatial and temporal resolution. Models of intermediate complexity bridge the gap [12]. These models are successful in describing the Earth system dynamics including a large number of Earth system components. This approach is especially useful when considering long time scales where the complex models are computationally too expensive, e.g. [47]. Improvements in the development of coupled models of intermediate complexity have led to a situation where modeling a glacial cycle, even with prognostic atmospheric CO_2 is becoming possible.

Climate simulation A climate simulation is the output of a computer program that attempts to simulate the climate evolution under appropriate boundary conditions. Simulations have become a useful part of climate science to gain insight into the sensitivity of the system.

Climate variability pattern Climate variability is defined as changes in integral properties of the climate system. True understanding of climate dynamics and prediction of future changes will come only with an understanding of the Earth system as a whole, and over past and present climate. Such understanding requires identification of the patterns of climate variability and their relationships to known forcing. Examples for climate variability patterns are the North Atlantic Oscillation (NAO) or the El Niño-Southern Oscillation (ENSO).

Abrupt climate change One can define abrupt climate change in the time and frequency domain. (a) Time domain: Abrupt climate change refers to a large shift in climate that persists for years or longer, such

as marked changes in average temperature, or altered patterns of storms, floods, or droughts, over a widespread area that takes place so rapidly that the natural system has difficulty adapting to it. In the context of past abrupt climate change, “rapidly” typically means on the order of a decade. (b) Frequency domain: An abrupt change means that the characteristic periodicity changes. Also the phase relation between certain climate variables may change in a relatively short time. For both types of changes examples will be provided.

Regime shifts are defined as rapid transitions from one state to another. In the marine environment, regimes may last for several decades, and shifts often appear to be associated with changes in the climate system. If the shifts occur regularly, they are often referred to as an oscillation (e. g., Atlantic Multi-decadal Oscillation, Pacific Decadal Oscillation). Similarly, one can define a regime shift in the frequency domain.

Anthropogenic climate change Beginning with the industrial revolution in the 1850s and accelerating ever since, the human consumption of fossil fuels has elevated CO₂ levels from a concentration of ~ 280 ppm to more than 380 ppm today. These increases are projected to reach more than 560 ppm before the end of the 21st century. As an example, a concomitant shift of ocean circulation would have serious consequences for both agriculture and fishing.

Multiple equilibria Fossil evidence and computer models demonstrate that the Earth’s complex and dynamic climate system has more than one mode of operation. Each mode produces different climate patterns. The evidence of models and data analysis shows that the Earth’s climate system has sensitive thresholds. Pushed past a threshold, the system can jump from one stable operating mode to a completely different one.

Long-term climate statistics Starting with a given initial state, the solutions $x(t)$ of the equations that govern the dynamics of a non-linear system, such as the atmosphere, result in a set of long-term statistics. If all initial states ultimately lead to the same set of statistical properties, the system is ergodic or transitive. If, instead, there are two or more different sets of statistical properties, where some initial states lead to one set, while the other initial states lead to another, the system is called intransitive (one may call the different states regimes). If there are different sets of statistics that a system may assume in its evolution from different initial states through a long, but finite, period of time, the system is called almost intransitive [50,51,53]. In the transitive case, the equilibrium climate statistics

are both stable and unique. Long-term climate statistics will give a good description of the climate. In the almost intransitive case, the system in the course of its evolution will show finite periods during which distinctly different climatic regimes prevail. The almost intransitive case arises because of internal feedbacks, or instabilities involving the different components of the climatic system. The climatic record can show rapid step-like shifts in climate variability that occur over decades or less, including climatic extremes (e. g. drought) that persist for decades.

Feedbacks A perturbation in a system with a negative feedback mechanism will be reduced whereas in a system with positive feedback mechanisms, the perturbation will grow. Quite often, the system dynamics can be reduced to a low-order description. Then, the growth or decay of perturbations can be classified by the systems’ eigenvalues or the pseudospectrum. Consider the stochastic dynamical system

$$\frac{d}{dt}x(t) = f(x) + g(x)\xi + F(x, t), \quad (1)$$

where ξ is a stochastic process. The functions f, g describe the climate dynamics, in this case without explicit time dependence. The external forcing $F(x, t)$ is generally time-, variable-, and space-dependent. In his theoretical approach, Hasselmann [32] formulated a linear stochastic climate model

$$\frac{d}{dt}x(t) = Ax + \sigma\xi + F(t), \quad (2)$$

with system matrix $A \in \mathbb{R}^{n \times n}$, constant noise term σ , and stochastic process ξ . Interestingly, many features of the climate system can be well described by (2), which is analogous to the Ornstein–Uhlenbeck process in statistical physics [89]. In the climate system, linear and non-linear feedbacks are essential for abrupt climate changes.

Paleoclimate Abrupt climate change is evident in model results and in instrumental records of the climate system. Much interest in the subject is motivated by the evidence in archives of extreme changes. Proxy records of paleoclimate are central to the subject of abrupt climate change. Available paleoclimate records provide information on many environmental variables, such as temperature, moisture, wind, currents, and isotopic compositions.

Thermohaline circulation stems from the Greek words “thermos” (heat) and “halos” (salt). The ocean is driven to a large extent by surface heat and freshwater

fluxes. As the ocean is non-linear, it cannot be strictly separated from the wind-driven circulation. The expressions thermohaline circulation (THC) and meridional overturning circulation (MOC) in the ocean are quite often used as synonyms although the latter includes all effects (wind, thermal, haline forcing) and describes the ocean transport in meridional direction. Another related expression is the ocean conveyor belt. This metaphor is motivated by the fact that the North Atlantic is the source of the deep limb of a global ocean circulation system [10]. If North Atlantic surface waters did not sink, the global ocean circulation would cease, currents would weaken or be redirected. The resulting reorganization would reconfigure climate patterns, especially in the Atlantic Ocean. One fundamental aspect of this circulation is the balance of two processes: cooling of the deep ocean at high latitudes, and heating of deeper levels from the surface through vertical mixing.

Definition of the Subject

The occurrence of abrupt change of climate at various time scales has attracted a great deal of interest for its theoretical and practical significance [2,3,9]. To some extent, a definition of what constitutes an abrupt climatic change depends on the sampling interval of the data being examined [28]. For the instrumental period covering approximately the last 100 years of annually or seasonally sampled data, an abrupt change in a particular climate variable will be taken to mean a statistically highly significant difference between adjacent 10-year sample means. In the paleoclimate context (i. e. on long time scales), an abrupt climate change can be in the order of decades to thousands of years. Since the climate dynamics can be often projected onto a limited number of modes or patterns of climate variability (e. g., [21,22]), the definition of abrupt climate change is also related to spatio-temporal patterns.

The concept of abrupt change of climate is therefore applied for different time scales. For example, changes in climatic regimes were described associated with surface temperature, precipitation, atmospheric circulation in North America during the 1920s and 1960s [19,75]. Sometimes, the term “climate jump” is used instead of “abrupt climate change”, e. g. [92]. Flohn [25] expanded the concept of abrupt climate change to include both singular events and catastrophes such as the extreme El Niño of 1982/1983, as well as discontinuities in paleoclimate indices taken from ice cores and other proxy data. In the instrumental record covering the last 150 years, there is a well-documented abrupt shift of sea surface tempera-

ture and atmospheric circulation features in the Northern Hemisphere in the mid-1970s, e. g. [22,67,88]. Some of the best-known and best-studied widespread abrupt climate changes started and ended during the last deglaciation, most pronounced at high latitudes.

In his classic studies of chaotic systems, Lorenz has proposed a deterministic theory of climate change with his concept of the “almost-intransitivity” of the highly non-linear climate systems. In this set of equations, there exists the possibility of multiple stable solutions to the governing equations, even in the absence of any variations in external forcing [51]. More complex models, e. g. [11,20] also demonstrated this possibility. On the other hand, variations in external forcing, such as the changes of incoming solar radiation, volcanic activity, deglacial meltwater, and increases of greenhouse gas concentration have also been proposed to account for abrupt changes in addition to climate intransitivity [9,25,38,41,49]. A particular climate change is linked to the widespread continental glaciation of Antarctica during the Cenozoic (65 Ma to present) at about 34 Ma, e. g. [93]. It should be noted that many facets of regional climate change are abrupt changes although the global means are rather smoothly changing.

Besides abrupt climate change as described in the time domain, we can find abrupt shifts in the frequency domain. A prominent example for an abrupt climate change in the frequency domain is the mid-Pleistocene transition or revolution (MPR), which is the last major “event” in a secular trend towards more intensive global glaciation that characterizes the last few tens of millions of years. The MPR is the term used to describe the transition between 41 ky (ky = 10^3 years) and 100 ky glacial-interglacial cycles which occurred about one million years ago (see a recent review in [61]). Evidence of this is provided by high-resolution oxygen isotope data from deep sea cores, e. g. [45,83].

Another example is the possibility of greenhouse gas-driven warming leading to a change in El Niño events. Modeling studies indicate that a strong enhancement of El Niño conditions in the future is not inconceivable [85]. Such a shift would have enormous consequences for both the biosphere and humans. The apparent phase shifts during the 1970s seems unique over this time period, and may thus represent a real climate shift although the available time series is probably too short to unequivocally prove that the shift is significant [90]. The inability to resolve questions of this kind from short instrumental time series provides one of the strongest arguments for extending the instrumental record of climate variability with well-dated, temporally finely resolved and rigorously calibrated proxy data.

Introduction

One view of climate change is that the Earth's climate system has changed gradually in response to both natural and human-induced processes. Researchers became intrigued by abrupt climate change when they discovered striking evidence of large, abrupt, and widespread changes preserved in paleoclimatic archives, the history of Earth's climate recorded in tree rings, ice cores, sediments, and other sources. For example, tree rings show the frequency of droughts, sediments reveal the number and type of organisms present, and gas bubbles trapped in ice cores indicate past atmospheric conditions.

The Earth's climate system is characterized by change on all time and space scales, and some of the changes are abrupt even relative to the short time scales of relevance to human societies. Paleoclimatic records show that abrupt climate changes have affected much or all of the Earth repeatedly over the last ice-age cycle as well as earlier – and these changes sometimes have occurred in periods as short as a few years, as documented in Greenland ice cores. Perturbations at northern high latitudes were spectacularly large: some had temperature increases of up to 10–20°C and a local doubling of precipitation within decades.

In the frequency domain, abrupt climate shifts are due to changes in the dominant oscillations (as in the case of the MPR), or due to a shift in the phase between different climate signals. As an example, the phase between the Indian Monsoon and ENSO exhibits significant shifts for the past 100 years [59].

The period of regular instrumental records of global climate is relatively short (100–200 years). Even so, this record shows many climatic fluctuations, some abrupt or sudden, as well as slow drifts in climate. Climatic changes become apparent on many temporal and spatial scales. Most abrupt climate changes are regional in their spatial extent. However, regional changes can have remote impacts due to atmospheric and oceanic teleconnections. Some of these shifts may be termed abrupt or sudden in that they represent relatively rapid changes in otherwise comparatively stable conditions, but they can also be found superimposed on other much slower climatic changes.

The definition of “abrupt” or “rapid” climate changes is therefore necessarily subjective, since it depends in large measure on the sample interval used in a particular study and on the pattern of longer-term variation within which the sudden shift is embedded. It is therefore useful to avoid a too general approach, but instead to focus on different types of rapid transitions as they are detected and modeled for different time periods. Although distinctions between

types are somewhat arbitrary, together they cover a wide range of shifts in dominant climate mode on time scales ranging from the Cenozoic (the last 65 millions of years) to the recent and future climate.

A Mathematical Definition

Time Domain

Abrupt climate change is characterized by a transition of the climate system into a different state (of temperature, rainfall, and other variables) on a time scale that is faster than variations in the neighborhood (in time). Abrupt climate change could be related to a forcing or internally generated. Consider $x(t) \in \mathbb{R}^n$ as a multi-dimensional climate state variable (temperature, rainfall, and other variables). We define an abrupt climate shift of degree ϵ and amplitude B , if

$$\frac{d}{dt}x_i(t) \quad \text{can be approximated by a function} \quad \frac{B}{\pi} \frac{\epsilon}{x_i^2 + \epsilon^2} \quad (3)$$

for one $i \in \{1, \dots, n\}$ in a time interval $[t_1, t_2]$. The case $\epsilon \rightarrow 0$ is called instantaneous climate shift, i. e. $x_i(t)$ can be approximated by the Heaviside step function. The degree of approximation can be specified by a proper norm.

An alternative way of defining an abrupt climate shift is through the identification of probable breaks in a time series (e. g., the surface temperature series). The formulation of a two-phase regression (TPR) test, e. g. [55,79], describing a series $x(t)$ is given by

$$x(t) = \mu_1 + \alpha_1 t + \epsilon(t) \quad \text{for } t \leq c \quad (4)$$

$$x(t) = \mu_2 + \alpha_2 t + \epsilon(t) \quad \text{for } t > c. \quad (5)$$

Under the null hypothesis of no changepoint, the two phases of the regression should be statistically equivalent and both the difference in means $\mu_{1,2}$, and the difference in slopes, $\alpha_{1,2}$, should be close to zero for each possible changepoint c .

In a stochastic framework one may use an appropriate stochastic differential equation (Langevin equation)

$$\frac{d}{dt}x(t) = f(x) + g(x)\xi, \quad (6)$$

where ξ is a stationary stochastic process and the functions $f, g: \mathbb{R}^n \rightarrow \mathbb{R}^n$ describe the climate dynamics. Abrupt climate change can be defined as a transition of a short period of time $[t_1, t_2]$, where the probability of an event is larger than a threshold. The properties of the random force

are described through its distribution and its correlation properties at different times. In the Ornstein–Uhlenbeck process ξ is assumed to have a Gaussian distribution of zero average,

$$\langle \xi(t) \rangle = 0 \quad (7)$$

and to be δ -correlated in time,

$$\langle \xi(t)\xi(t + \tau) \rangle = \delta(\tau) . \quad (8)$$

The brackets indicate an average over realizations of the random force. For a Gaussian process only the average and second moment need to be specified since all higher moments can be expressed in terms of the first two. Note that the dependence of the correlation function on the time difference τ assumes that ξ is a stationary process.

The probability density $p(x, t)$ for the variable $x(t)$ in (6) obeys the Fokker–Planck equation

$$\partial_t p = -\frac{\partial}{\partial x} [f(x)p] + \frac{\partial}{\partial x} \left[g(x) \frac{\partial}{\partial x} \{g(x)p\} \right] . \quad (9)$$

Its stationary probability density of (6) is given by

$$p_{st}(x) = \aleph \exp \left(-2 \int_{x_0}^x \frac{f(y) - g(y)g'(y)}{g(y)^2} dy \right) , \quad (10)$$

where \aleph is a normalization constant. $g'(y)$ stands for the derivative of g with respect to its argument. The extrema x_m of the steady state density obey the equation

$$f(x_m) - g(x_m)g'(x_m) = 0 \quad (11)$$

for $g(x_m) \neq 0$. Here is the crux of the noise-induced transition phenomenon: one notes that this equation is not the same as the equation $f(x_m) = 0$ that determines the steady states of the system in the absence of multiplicative noise. As a result, the most probable states of the noisy system need not coincide with the deterministic stationary states. More importantly, new solutions may appear or existing solutions may be destabilized by the noise. These are the changes in the asymptotic behavior of the system caused by the presence of the noise, e. g. [84].

Climate Variability and Climate Change

The temporal evolution of climate can be expressed in terms of two basic modes: the forced variations which are the response of the climate system to changes in the external forcing $F(x, t)$ (mostly called climate change), and the free variations owing to internal instabilities and feedbacks leading to non-linear interactions among the various components of the climate system [68] (mostly called climate

variability). The external causes $F(x, t)$, operate mostly by causing variations in the amount of solar radiation received by or absorbed by the Earth, and comprise variations in both astronomical (e. g. orbital parameters) and terrestrial forcings (e. g. atmospheric composition, aerosol loading). For example, the diurnal and seasonal variations in climate are related to external astronomical forcings operating via solar radiation, while ice ages are related to changes in Earth orbital parameters. Volcanic eruptions are one example of a terrestrial forcing which may introduce abrupt modifications over a period of 2 or 3 years. The more rapid the forcing, the more likely it is that it will cause an abrupt change. The resulting evolution may be written as

$$\frac{d}{dt}x(t) = f(x) + g(x)\xi + F(x, t) . \quad (12)$$

$F(x, t)$ is independent of x if the forcing does not depend on climate (external forcing).

The internal free variations within the climate system are associated with both positive and negative feedback interactions between the atmosphere, oceans, cryosphere and biosphere. These feedbacks lead to instabilities or oscillations of the system on all time scales, and can either operate independently or reinforce external forcings. Investigations of the properties of systems which are far from equilibrium show that they have a number of unusual properties. In particular, as the distance from equilibrium increases, they can develop complex oscillations with both chaotic and periodic characteristics. They also may show bifurcation points where the system may switch between various regimes. Under non-equilibrium conditions, local events have repercussions throughout the whole system. These long-range correlations are at first small, but increase with distance from equilibrium, and may become essential at bifurcation points.

When applying (12), different concepts of climate change are in the literature. Quite often, the dynamics is governed by the following stochastic differential equation

$$\frac{d}{dt}x(t) = -\frac{d}{dx}U(x) + \sigma\xi + F(t) \quad (13)$$

with potential

$$U(x) = a_4x^4 + a_3x^3 + a_2x^2 + a_1x . \quad (14)$$

If the potential is quadratic and $F(t) = 0$, the Ornstein–Uhlenbeck process is retained. In contrast, a bistable non-linear system with two minima in $U(x)$ has been assumed in which shifts between the two distinctly different states are triggered randomly by stochastic forcing, e. g. [7]. In

such a system, climate variability and change in the potential can interact due to stochastic resonance [1,7]. Stochastic resonance occurs when the signal-to-noise ratio of a non-linear device is maximized for a moderate value of noise intensity σ . It often occurs in bistable and excitable systems with sub-threshold inputs. For lower noise intensities, the signal does not cause the device to cross threshold, so little signal is passed through it. For large noise intensities, the output is dominated by the noise, also leading to a low signal-to-noise ratio. For moderate intensities, the noise allows the signal to reach threshold, but the noise intensity is not so large as to swamp it.

Strictly speaking, stochastic resonance occurs in bistable systems, when a small periodic force $F(t)$ (which is external) is applied together with a large wide-band stochastic force $\sigma\xi$ (which is internal). The system response is driven by the combination of the two forces that compete/cooperate to make the system switch between the two stable states. The degree of order is related to the amount of periodic function that it shows in the system response. When the periodic force is chosen small enough in order to not make the system response switch, the presence of a non-negligible noise is required for it to happen. When the noise is small very few switches occur, mainly at random with no significant periodicity in the system response. When the noise is very strong a large number of switches occur for each period of the periodic force and the system response does not show remarkable periodicity. Quite surprisingly, between these two conditions, there exists an optimal value of the noise that cooperatively concurs with the periodic forcing in order to make almost exactly one switch per period (a maximum in the signal-to-noise ratio).

Furthermore, non-linear oscillators have been proposed where the timing of the deterministic external forcing is crucial for generating oscillations [51,77,78]. Some aspects of non-equilibrium systems can be found in the climatic system. On the climatological scale, it exhibits abrupt jumps in the long-term rate of temperature change, which are often associated with changes in circulation patterns.

Frequency Domain

In the frequency domain, there are different ways to describe abrupt climate change. A stationary process exhibits an autocovariance function of the form

$$\text{Cov}(\tau) = \langle (x(t + \tau) - \langle x \rangle)(x(t) - \langle x \rangle) \rangle \quad (15)$$

where $\langle \rangle$ denotes the expectation or the statistical mean. Normalized to the variance (i. e. the autocovariance func-

tion at $\tau = 0$) one gets the autocorrelation function $C(\tau)$:

$$C(\tau) = \text{Cov}(\tau) / \text{Cov}(0) . \quad (16)$$

Many stochastic processes in nature exhibit short-range correlations, which decay exponentially:

$$C(\tau) \sim \exp(-\tau/\tau_0) , \quad \text{for } \tau \rightarrow \infty . \quad (17)$$

These processes exhibit a typical time scale τ_0 .

As the frequency domain counterpart of the autocovariance function of a stationary process, one can define the spectrum as

$$S(\omega) = \widehat{\text{Cov}(\tau)} , \quad (18)$$

where the hat denotes the Fourier transformation. However, geophysical processes are furthermore often non-stationary. In this regard, the optimal method is continuous wavelet analysis as it intrinsically adjusts the time resolution to the analyzed scale, e. g. [16,59].

Wavelet Spectra A major question concerns the significance testing of wavelet spectra. Torrence and Compo [86] formulated pointwise significance tests against reasonable background spectra. However, Maraun and Kurths [58] pointed out a serious deficiency of pointwise significance testing: Given a realization of white noise, large patches of spurious significance are detected, making it – without further insight – impossible to judge which features of an estimated wavelet spectrum differ from background noise and which are just artifacts of multiple testing. Under these conditions, a reliable corroboration of a given hypothesis is impossible. This demonstrates the necessity to study the significance testing of continuous wavelet spectra in terms of sensitivity and specificity. Given the set of all patches with pointwise significant values, areawise significant patches are defined as the subset of additionally areawise significant wavelet spectral coefficients given as the union of all critical areas that completely lie inside the patches of pointwise significant values. Whereas the specificity of the areawise test appears to be – almost independently of the signal-to-noise ratio – close to one, that of the pointwise test decreases for high background noise, as more and more spurious patches appear [58].

Eigenvalues and Pseudospectrum Another spectral method characterizing the abruptness of climate change is related to the resonance of the linear system (1). As we will see later in the context of atmosphere and ocean instabilities, an eigenvalue analysis is inappropriate in describing the dynamics of the system (12). Inspection of many geo-

physical systems shows that most of the systems fail the normality condition

$$A A^\dagger = A^\dagger A, \quad (19)$$

where † denotes the adjoint-complex operator. If a matrix is far from normal, its eigenvalues (the spectrum) have little to do with its temporal evolution [71,87]. More about the dynamics can be learned by examining the pseudospectrum of A in the complex plane. The ϵ -pseudospectrum of operator A is defined by two equivalent formulations:

$$\begin{aligned} \Lambda_\epsilon(A) &= \{z \in \mathbb{C} : \|(zI - A)^{-1}\| \geq \epsilon^{-1}\} \\ &= \{z \in \mathbb{C} : [\text{smallest singular value of } (zI - A)] \leq \epsilon\}. \end{aligned} \quad (20)$$

This set of values z in the complex plane are defined by contour lines of the resolvent $(zI - A)^{-1}$. The resolvent determines the system's response to a forcing as supplied by external forcing $F(x, t)$, stochastic forcing $g(x)\xi$, or initial/boundary conditions. The pseudospectrum reflects the robustness of the spectrum and provides information about instability and resonance. One theorem is derived from Laplace transformation stating that transient growth is related to how far the ϵ -pseudospectrum extends into the right half plane:

$$\|\exp(A t)\| \geq \frac{1}{\epsilon} \sup_{z \in \Lambda_\epsilon(A)} \text{Real}(z). \quad (21)$$

In terms of climate theory, the pseudospectrum indicates resonant amplification. Maximal amplification is at the poles of $(zI - A)^{-1}$, characterized by the eigenfrequencies. In a system satisfying (19), the system's response is characterized solely by the proximity to the eigenfrequencies. In the non-normal case, the pseudospectrum shows large resonant amplification for frequencies which are not eigenfrequencies. This transient growth mechanism is important for both initial value and forced problems.

Earth System Modeling and Analysis

Hierarchy of Models

Modeling is necessary to produce a useful understanding of abrupt climate processes. Model analyses help to focus research on possible causes of abrupt climate change, such as human activities; on key areas where climatic thresholds might be crossed; and on fundamental uncertainties in climate-system dynamics. Improved understanding of abrupt climatic changes that occurred in the past and that

are possible in the future can be gained through climate models. A comprehensive modeling strategy designed to address abrupt climate change includes vigorous use of a hierarchy of models, from theory and conceptual models through models of intermediate complexity, to high-resolution models of components of the climate system, to fully coupled earth-system models. The simpler models are well-suited for use in developing new hypotheses for abrupt climate change. Model-data comparisons are needed to assess the quality of model predictions. It is important to note that the multiple long integrations of enhanced, fully coupled Earth system models required for this research are not possible with the computer resources available today, and thus, these resources are currently being enhanced.

Feedback

One particularly convincing example showing that the feedbacks in the climate system are important is the drying of the Sahara about 5000 years before present which is triggered by variations in the Earth's orbit around the sun. Numerous modeling studies, e.g. [31], suggest that the abruptness of the onset and termination of the early to mid-Holocene humid period across much of Africa north of the equator depends on the presence of non-linear feedbacks associated with both ocean circulation and changes in surface hydrology and vegetation, e.g. [18]. Without including these feedbacks alongside gradual insolation forcing, it is impossible for existing models to come even close to simulating the rapidity or the magnitude of climatic change associated with the extension of wetlands and plant cover in the Sahara/Sahel region prior to the onset of desiccation around 5000 years before present.

Climate Archives and Modeling

Systematic measurements of climate using modern instruments have produced records covering the last 150 years. In order to reconstruct past variations in the climate system further back in time, scientists use natural archives of climatic and environmental changes, such as ice cores, tree rings, ocean and lake sediments, corals, and historical evidence. Scientists call these records proxies because, although they are not usually direct measures of temperature or other climatic variables, they are affected by temperature, and using modern calibrations, the changes in the proxy preserved in the fossil record can be interpreted in terms of past climate.

Ice core data, coral data, ring width of a tree, or information from marine sediments are examples of a proxy for temperature, or in some cases rainfall, because the thick-

ness of the ring can be statistically related to temperature and/or rainfall in the past. The most valuable proxies are those that can be scaled to climate variables, and those where the uncertainty in the proxy can be measured. Proxies that cannot be quantified in terms of climate or environment are less useful in studying abrupt climate change because the magnitude of change cannot be determined. Quite often, the interpretation of proxy data is already a model of climate change since it involves constraints (dating, representativeness etc.). Uncertainties in the proxies, and uncertainties in the dating, are the main reasons that abrupt climate change is one of the more difficult topics in the field of paleoclimatology.

Example: Glacial-Interglacial Transitions

Astronomical Theory of Ice Ages

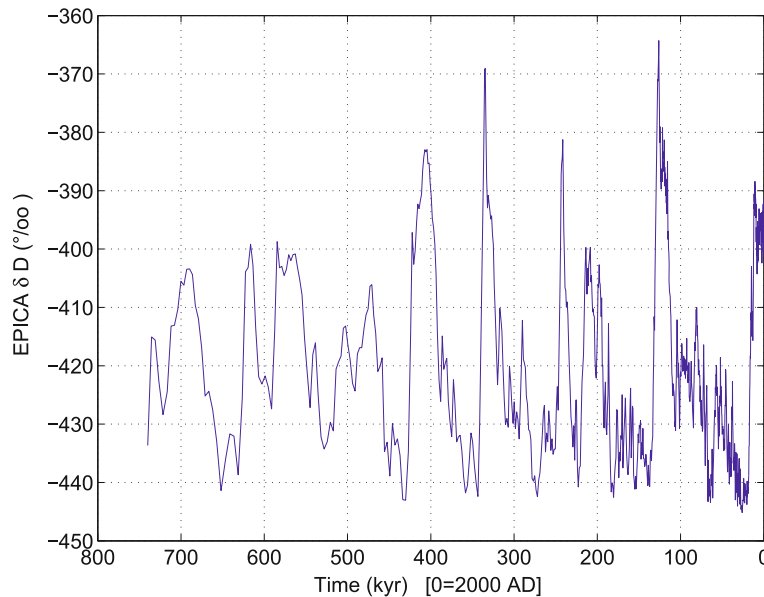
Over the past half million years, marine, polar ice core and terrestrial records all highlight the sudden and dramatic nature of glacial terminations, the shifts in global climate that occurred as the world passed from dominantly glacial to interglacial conditions, e. g. [23,69]. These climate transitions, although probably of relatively minor relevance to the prediction of potential future rapid climate change, do provide the most compelling evidence available in the historical record for the role of greenhouse gas, oceanic and biospheric feedbacks as non-linear amplifiers in the cli-

mate system. It is such evidence of the dramatic effect of non-linear feedbacks that shows relatively minor changes in climatic forcing may lead to abrupt climate response.

A salient feature of glacial-interglacial climate change is furthermore its asymmetry (Fig. 1). Warmings are rapid, usually followed by slower descent into colder climate. Given the symmetry of orbital forcings $F(t)$, the cause of rapid warming at glacial “terminations” must lie in a climate feedback [37,65]. Clearly, the asymmetric feedback is due to the albedo (reflectance) of ice and snow changing from high values under glacial climates to low values under warm climates. The albedo feedback helps explain the rapidity of deglaciations and their beginnings in spring and summer. Increased absorption of sunlight caused by lower albedo provides the energy for rapid ice melt. The build-up of snow and ice takes much longer than melting.

Many simplified climate models consist of only a few coupled ordinary differential equations controlled by carefully selected parameters. It is generally acknowledged that the “best” models will be those that contain a minimum of adjustable parameters [77] and are robust with respect to changes in those parameters. Rial [72] formulated a logistic-delayed and energy balance model to understand the saw-tooth shape in the paleoclimate record: A fast warming-slow cooling is described by

$$\frac{d}{dt}x(t) = R \left(1 - \frac{x(t-\tau)}{K(t)} \right) x(t-\tau) \quad (22)$$



Abrupt Climate Change Modeling, Figure 1

Oxygen isotope record from a southern hemisphere ice core [23] showing the glacial-interglacial changes. Note the asymmetry: the state is longer in the cold (glacials) phases than in the warm phases (interglacials)

$$C \frac{d}{dt} T(t) = Q(1 - \alpha(x)) - (A + BT) \quad (23)$$

with $x(t)$ for the normalized ice extent, τ time delay, $K(t) = 1 + e(t)T(t)$ carrying capacity, $1/R$ response time of the ice sheet, $T(t)$ global mean temperature, $\alpha(x)$ planetary albedo, external parameter $e(t)$, and $R\tau$ bifurcation parameter. A, B, C, Q are constants for the energy balance of the climate. The equation is calibrated so that for $x(t) = 1$ the albedo $\alpha(x) = 0.3$ and $T(t) = 15^\circ\text{C}$. With (23), saw-toothed waveforms and frequency modulation can be understood [72]. The delayed equation yields damped oscillations of $x(t)$ about the carrying capacity for small τ . If τ becomes long compared to the natural response time of the system, the oscillations will become strong, and will grow in amplitude, period and duration. As in the logistic equation for growth, here the product $R\tau$ is a bifurcation parameter, which when crossing the threshold value $\pi/2$ makes the solutions undergo a Hopf bifurcation and settle to a stable limit cycle with fundamental period $\sim 4\tau$ [73].

The astronomical theory of ice ages – also called Milankovitch theory [62] – gained the status of a paradigm for explaining the multi-millennial variability. A key element of this theory is that summer insolation at high latitudes of the northern hemisphere determines glacial-interglacial transitions connected with the waxing and waning of large continental ice sheets, e.g. [33,37], the dominant signal in the climate record for the last million

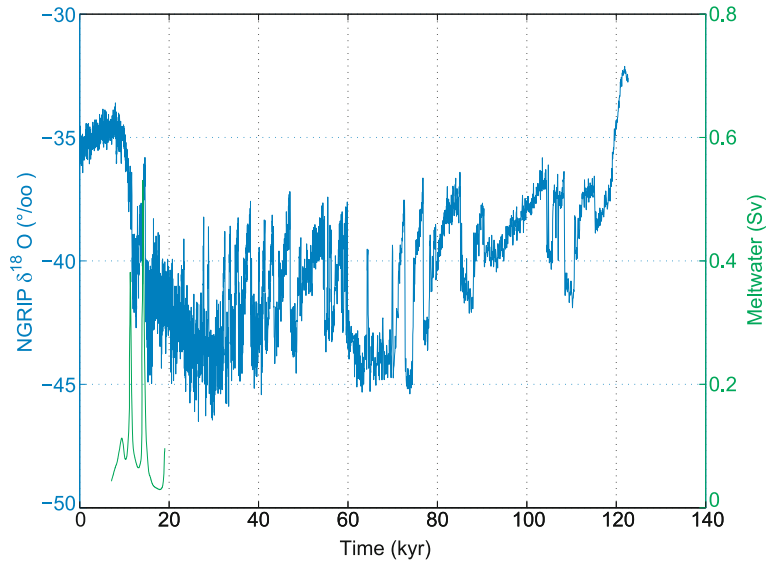
years. Climate conditions of glacials and interglacials are very different. During the Last Glacial Maximum, about 20,000 years before present, surface temperature in the north Atlantic realm was $10\text{--}20^\circ\text{C}$ lower than today [13]. A recent study of Huybers and Wunsch [36] has shown that the most simple system for the phase of ice volume $x(t)$ is given by

$$x(t+1) = x(t) + \sigma \xi \quad (24)$$

with ξ a Gaussian white noise process, but with mean $\mu = 1$, and $\sigma = 2$. ξ represents the unpredictable background weather and climate variability spanning all time scales out to the glacial/interglacial. This highly simplified model posits 1-ky steps in ice volume $x(t)$. The non-zero mean biases the Earth toward glaciation. Once $x(t)$ reaches a threshold, a termination is triggered, and ice-volume is linearly reset to zero over 10 ky. The following threshold condition for a termination makes it more likely for a termination of ice volume to occur when obliquity $\Theta(t)$ is large:

$$x(t) \geq T_0 - a\Theta(t). \quad (25)$$

$\Theta(t)$ has a frequency of about 41 ky, and is furthermore normalized to zero mean with unit variance. The other parameters are: amplitude $a = 15$, $T_0 = 105$. Furthermore, the initial ice volume at 700 ky before present is set to $x(t = -700) = 30$. Equation (24) resembles an order-one autoregressive process, similar to (2), plus the threshold



Abrupt Climate Change Modeling, Figure 2

Oxygen isotope record from a Greenland ice core record [64] using an updated time scale for this record [23]. Green: Sea-level derived rate of deglacial meltwater discharge [24] which is strong after deglacial warming

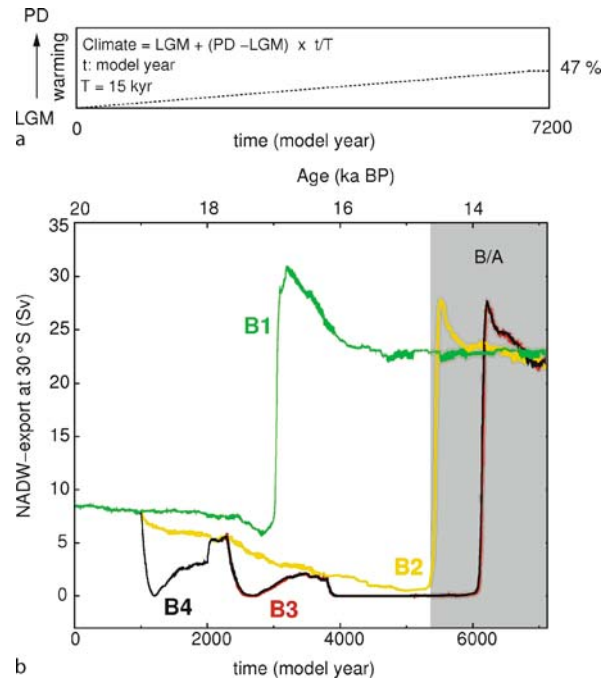
condition (25). Models like (24), (25) are not theories of climate change, but rather attempts at efficient kinematic descriptions of the data, and different mechanisms can be consistent with the limited observational records. In the next section, the process of deglaciation is modeled in a three-dimensional model including the spatial dimension.

Deglaciation

The question is what causes the abrupt warming at the onset of the Boelling as seen in the Greenland ice cores (Fig. 2). There is a clear antiphasing seen in the deglaciation interval between 20 and 10 ky ago: During the first half of this period, Antarctica steadily warmed, but little change occurred in Greenland. Then, at the time when Greenland's climate underwent an abrupt warming, the warming in Antarctica stopped. Knorr and Lohmann [42], also summarizing numerous modeling studies for deglaciation, describe how global warming (which may be induced by greenhouse gases and feedbacks) can induce a rapid intensification of the ocean circulation (Fig. 3). During the Boelling/Alleroed, a sudden increase of the northward heat transport draws more heat from the south, and leads to a strong warming in the north. This “heat piracy” from the South Atlantic has been formulated by Crowley [15]. A logical consequence of this heat piracy is the Antarctic Cold Reversal (ACR) during the Northern Hemisphere warm Boelling/Alleroed. This particular example shows that an abrupt climate change of the ocean circulation (with large climate impacts in the North Atlantic) is related to a smooth global warming. To understand the dynamical behavior of the system, the concept of hysteresis is applied, using the global warming after the last ice ages as the control parameter [42]. The system exhibits multiple steady states (Fig. 4): a weak glacial ocean circulation and a stronger circulation (which is comparable in strength to the modern mode of operation). Deglacial warming induces a transition from a weak glacial THC state to a stronger THC state, characterizing the abrupt warming during the deglaciation.

Millennial Climate Variability

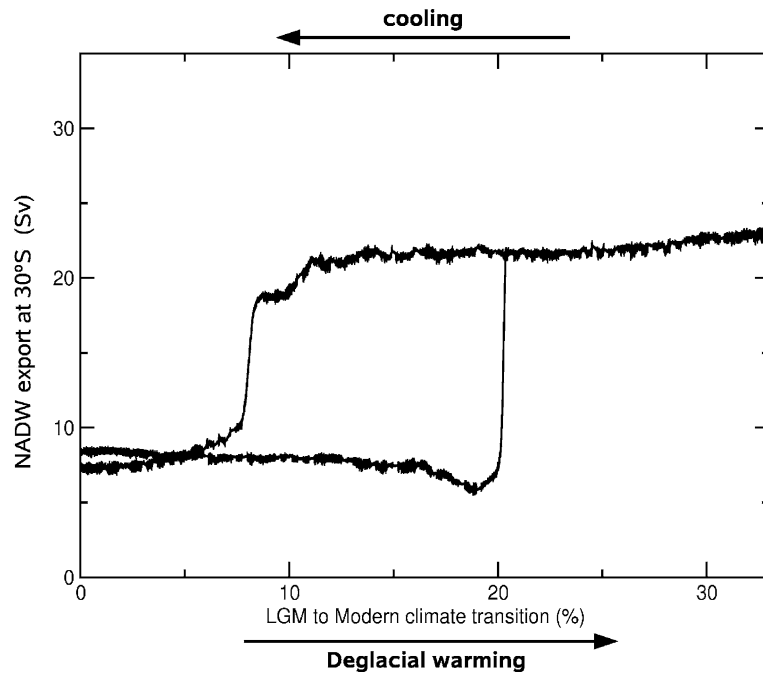
Within glacial periods, and especially well documented during the last one, spanning from around 110 to 11.6 ky ago, there are dramatic climate oscillations, including high-latitude temperature changes approaching the same magnitude as the glacial cycle itself, recorded in archives from the polar ice caps, high to middle latitude marine sediments, lake sediments and continental loess sections. These oscillations are usually referred to as the Dans-



Abrupt Climate Change Modeling, Figure 3

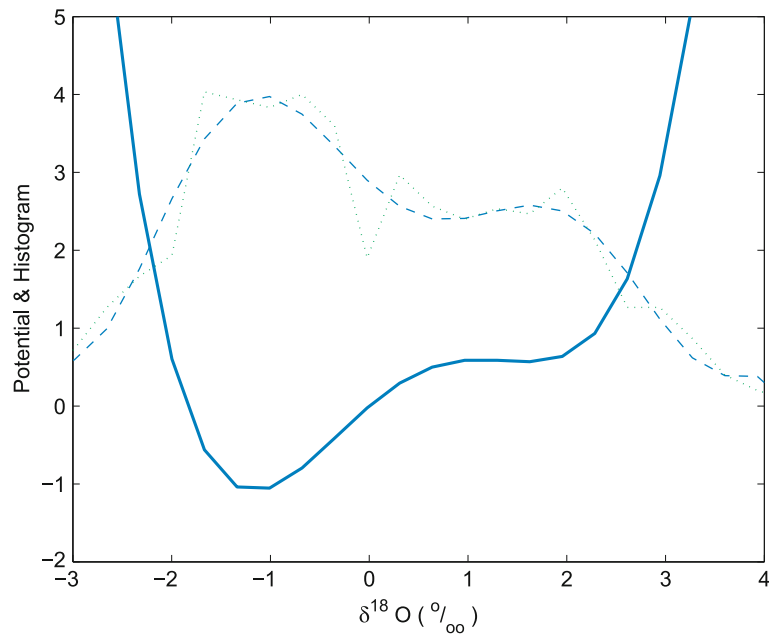
Forcing and model response of the ocean overturning rate. **a** The background climate conditions are linearly interpolated between glacial (LGM), and modern (PD), conditions. Gradual warming is stopped after 7000 model years, which is related to $\sim 47\%$ of the total warming. **b** Circulation strength (export at 30°S) versus time. The green curve B1 represents the experiment without any deglacial freshwater release to the North Atlantic. Experiments B2 (yellow curve), B3 (red curve), and B4 (black curve), exhibit different successions of deglacial meltwater pulse scenarios to the North Atlantic [42]

gaard-Oeschger Cycle and occur mostly on 1 to 2 ky time scales, e.g. [6], although regional records of these transitions can show much more rapid change. The termination of the Younger Dryas cold event, for example, is manifested in ice core records from central Greenland as a near doubling of snow accumulation rate and a temperature shift of around 10°C occurring within a decade with world-wide teleconnections. One hypothesis for explaining these climatic transitions is that the ocean thermohaline circulation flips between different modes, with warm intervals reflecting periods of strong deep water formation in the northern North Atlantic and vice versa [29]. As an alternative approach, one can estimate the underlying dynamics (13), (14) directly from data [43]. The method is based on the unscented Kalman filter, a non-linear extension of the conventional Kalman filter. This technique allows one to consistently estimate parameters in deterministic and stochastic non-linear models. The optimiza-



Abrupt Climate Change Modeling, Figure 4

Hysteresis loop of the ocean overturning strength (*black curve*) with respect to slowly varying climate background conditions. The transition values are given in % of a full glacial-interglacial transition [42]



Abrupt Climate Change Modeling, Figure 5

Potential derived from the data (*solid*) together with probability densities of the model (*dashed*) and the data (*dotted*)

tion yields for the coefficients $a_4 = 0.13 \pm 0.01$, $a_3 = -0.27 \pm 0.02$, $a_2 = -0.36 \pm 0.08$, and $a_1 = 1.09 \pm 0.23$. The dynamical noise level of the system σ is estimated to be 2.4. The potential is highly asymmetric and degenerate (that is, close to a bifurcation): there is one stable cold stadial state and one indifferently stable warm interstadial state (Fig. 5). This seems to be related to the fact that the warm intervals are relatively short-lasting.

Coming back to the ice cores and a potential linkage of the hemispheres, Stocker and Johnson [81] proposed a conceptual model linking the isotopic records from Antarctica and Greenland. The basis is an energy balance with temperatures in the North and South Atlantic Ocean, as well as a “southern heat reservoir”. It is assumed that the change in heat storage of a “southern heat reservoir” T_S is given by the temperature difference between the reservoir T_S and the Southern Ocean temperature T , with a characteristic time scale τ :

$$\frac{d}{dt} T_S(t) = \frac{1}{\tau} [T - T_S] . \quad (26)$$

T_N denotes the time-dependent temperature anomaly of the North Atlantic. The Southern Ocean temperature T is assumed to be $-T_N$ according to the bipolar seesaw (North Atlantic cold \leftrightarrow South Atlantic warm). Using Laplace transform, one can solve for T_S

$$T_S = -\frac{1}{\tau} \int_0^t T_N(t-t') \exp(-t'/\tau) dt' + T_S(0) \exp(-t/\tau) . \quad (27)$$

The reservoir temperature is therefore a convolution of the northern temperature using the time scale τ ranging from 100 to 4000 years. Equation (27) demonstrates that T_S and T_N will have entirely different time characteristics. Abrupt changes in the north appear damped and integrated in time in the southern reservoir. A sudden reduction in the thermohaline circulation causes a cooling in the North Atlantic and a warming in the South, a situation similar to the Younger Dryas period [80], see also Fig. 2.

Example: Cenozoic Climate Cooling

Antarctic Glaciation

During the Cenozoic (65 million years ago (Ma) to present), there was the widespread glaciation of the Antarctic continent at about 34 Ma, e.g. [93]. Antarctic glaciation is the first part of a climate change from relatively warm and certainly ice-free conditions to massive ice sheets in both, the southern and northern hemi-

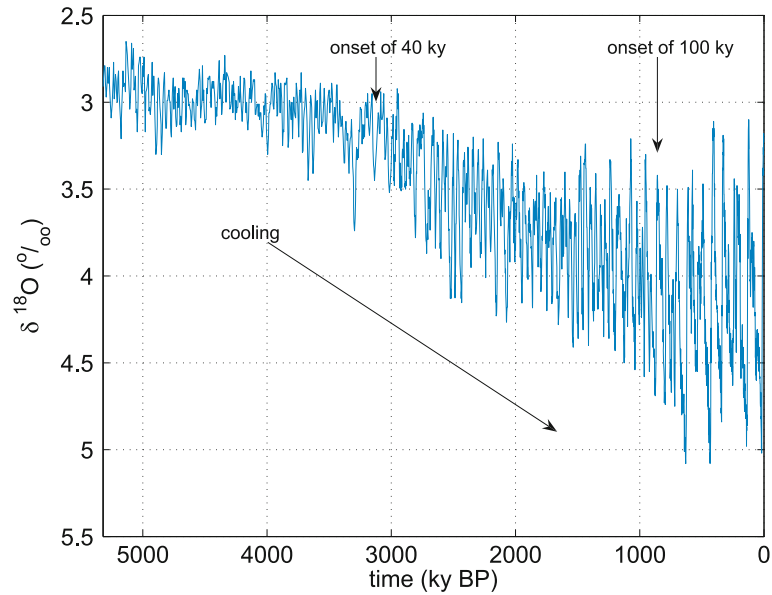
spheres [44]. Opening of circum-Antarctic seaways is one of the factors that have been ascribed as a cause for Antarctic climate change so far [40,93]. Besides gateway openings, the atmospheric carbon dioxide concentration is another important factor affecting the evolution of the Cenozoic climate [17,93]. As a third component in the long-term evolution of Antarctic glaciation, land topography is able to insert certain thresholds for abrupt ice sheet build-up. Whereas tectonics, land topography, and long-term Cenozoic CO_2 -decrease act as preconditioning for Antarctic land ice formation, the cyclicities of the Earth's orbital configuration are superimposed on shorter time scales and may have served as the ultimate trigger and pacemaker for ice-sheet growth at the Eocene-Oligocene boundary around 34 Ma [14].

DeConto and Pollard [17] varied Southern Ocean heat transport to mimic gateway opening instead of an explicit simulation of ocean dynamics. They found a predominating role of pCO_2 in the onset of glaciation instead of a dominating tectonic role for “thermal isolation”.

Mid-Pleistocene Revolution

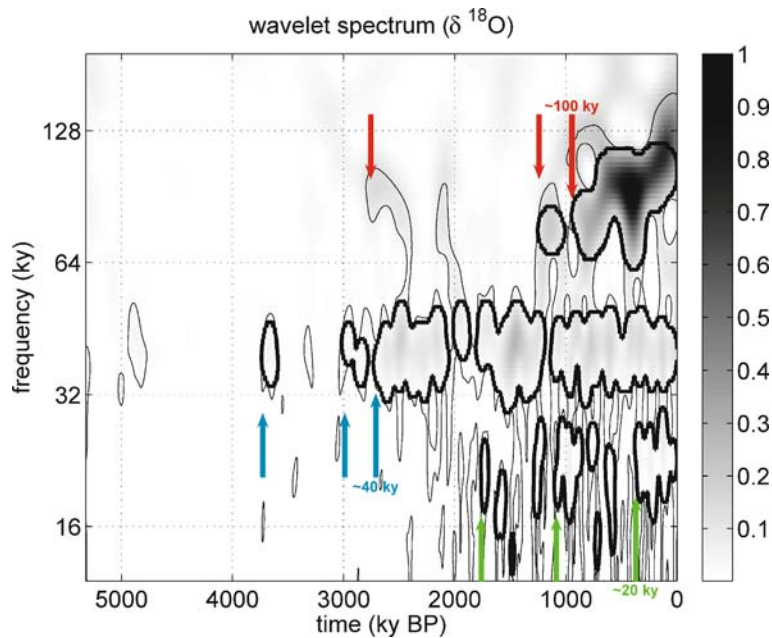
Glaciation in the Northern Hemisphere lagged behind, with the earliest recorded glaciation anywhere in the Northern Hemisphere occurring between 10 and 6 Ma and continuing through to the major increases in global ice volume around 2–3 Ma [60]. A recent compilation of 57 globally distributed records [45] is shown in Fig. 6. Let us focus now on the mid-Pleistocene transition or revolution (MPR), describing the transition from 41 ky to 100 ky glacial-interglacial cycles.

Milankovitch [62] initially suggested that the critical factor was total summer insolation at about 65°N , because for an ice sheet to grow some additional ice must survive each successive summer. In contrast, the Southern Hemisphere is limited in its response because the expansion of ice sheets is curtailed by the Southern Ocean around Antarctica. The conventional view of glaciation is thus that low summer insolation in the temperate North Hemisphere allows ice to survive summer and thus start to build up on the northern continents. If so, how then do we account for the MPR? Despite the pronounced change in Earth system response evidenced in paleoclimatic records, the frequency and amplitude characteristics of the orbital parameters which force long-term global climate change, e.g., eccentricity (~ 100 ky), obliquity (~ 41 ky) and precession (~ 21 and ~ 19 ky), do not vary during the MPR [8]. This suggests that the cause of change in response at the MPR is internal rather than external to the global climate system.



Abrupt Climate Change Modeling, Figure 6

A compilation of 57 globally distributed records by Lisiecki and Raymo [45]: The $\delta^{18}\text{O}$ record reflects mainly the climate variables temperature and ice volume



Abrupt Climate Change Modeling, Figure 7

Lisiecki and Raymo [45]: The corresponding wavelet sample spectrum calculated using Morlet wavelet with $\omega_0 = 6$. Thin and thick lines surround pointwise and areawise significant patches, respectively

The result of a wavelet spectral analysis (Fig. 7) suggests several abrupt climate changes in the frequency domain (shown as schematic arrows in the figure). These abrupt climate shifts represent major reorganizations in

the climate system. Some of them are possibly linked to the development of Northern Hemisphere ice volume. The MPR marked a prolongation to and intensification of the ~ 100 ky glacial-interglacial climate. Not only does

the periodicity of glacial-interglacial cycles increase going through the MPR, but there is also an increase in the amplitude of global ice volume variations.

It is likely that the MPR is a transition to a more intense and prolonged glacial state, and associated subsequent rapid deglaciation becomes possible. The first occurrence of continental-scale ice sheets, especially on Greenland, is recorded as ice-rafted detritus released from drifting icebergs into sediments of the mid- and high-latitude ocean. After a transient precursor event at 3.2 Ma, signals of large-scale glaciations suddenly started in the subpolar North Atlantic in two steps, at 2.9 and 2.7 Ma, e. g. [5].

The ice volume increase may in part be attributed to the prolonging of glacial periods and thus of ice accumulation. The amplitude of ice volume variation is also accentuated by the extreme warmth of many interglacial periods. Thus, a colder climate with larger ice sheets should have the possibility of a greater sudden warming [45]. The MPR therefore marks a dramatic sharpening of the contrast between warm and cold periods. Note however, that the amount of energy at the 40 ka period is hardly changed in the time after 1 Ma, and notably, one sees the addition of energy at longer periods, without any significant reduction in obliquity-band energy. After about 1 Ma, large glacial-interglacial changes begin to occur on an approximately 100 ka time scale (but not periodically) superimposed upon the variability which continues largely unchanged [91]. Why did 100 ka glacial-interglacials also become possible in addition to the ice volume variability? Lowering of global CO₂ below some critical threshold, or changes in continental configuration, or atmospheric circulation patterns, or all together, are among the conceivable possibilities, e. g. [70].

Examples: Transient Growth

The former examples show the power of the combination of models, data analysis, and interpretation for abrupt climate change. In the next two examples, it is shown how important the transient growth mechanism is for abrupt climate change.

Conceptual Model of the Ocean Circulation

In this section, a category of the non-linear models following the simple thermohaline model of Stommel [82] is analyzed. The common assumption of these box models is that the oceanic overturning rate Φ can be expressed by the meridional density difference:

$$\Phi = -c(\alpha\Delta T - \beta\Delta S), \quad (28)$$

where α and β are the thermal and haline expansion coefficients, c is a tunable parameter, and Δ denotes the meridional difference operator applied to the variables temperature T and salinity S , respectively. Stommel [82] considered a two-box ocean model where the boxes are connected by an overflow at the top and a capillary tube at the bottom, such that the capillary flow is directed from the high-density vessel to the low-density vessel following (28).

The equations for temperature T and salinity S are the heat and salt budgets using an upstream scheme for the advective transport and fluxes with the atmosphere:

$$\frac{d}{dt}T = -\frac{\Phi}{V}\Delta T - \frac{F_{oa}}{\rho_0 c_p h} \quad (29)$$

$$\frac{d}{dt}S = -\frac{\Phi}{V}\Delta S - \frac{S_0}{h}(P - E), \quad (30)$$

where V is the volume of the box with depth h , and $(P - E)$ denotes the freshwater flux (precipitation minus evaporation plus runoff). F_{oa} is the heat flux at the ocean-atmosphere interface, S_0 is a reference salinity, and $\rho_0 c_p$ denotes the heat capacity of the ocean.

Denoting furthermore $x \in \mathbb{R}^2$ for the anomalies of $(\Delta T, \Delta S)$, Lohmann and Schneider [48] have shown the evolution equation is of the following structure:

$$\frac{d}{dt}x = Ax + \langle b|x \rangle x. \quad (31)$$

The brackets $\langle | \rangle$ denote the Euclidean scalar product. This evolution Equation (31) can be transferred to a

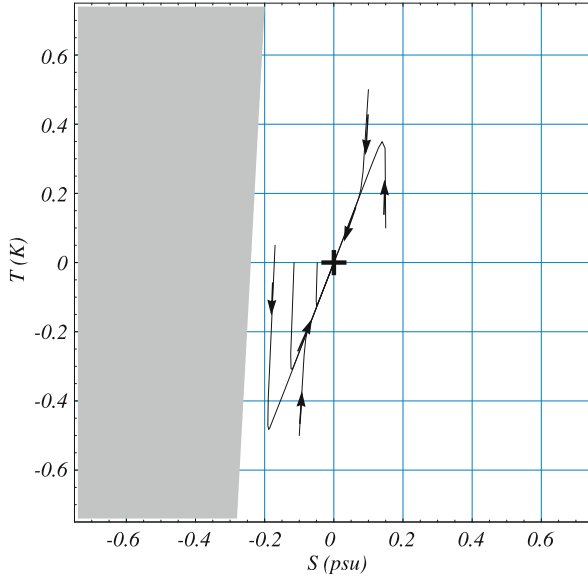
$$x(t) = \frac{1}{\gamma(t)} \exp(At)x_0, \quad (32)$$

with a scaling function $\gamma(t, x_0)$. The models of Stommel [82], and many others are of this type, and their dynamics are therefore exactly known.

It is worth knowing that (29), (30) is equivalent to the multi-dimensional Malthus-Verhulst model (also known as the logistic equation), which was originally proposed to describe the evolution of a biological population. Let x denote the number (or density) of individuals of a certain population. This number will change due to growth, death, and competition. In the simplest version, birth and death rates are assumed proportional to n , but accounting for limited resources and competition it is modified by $(1 - x)$:

$$\frac{d}{dt}x(t) = a(1 - x)x. \quad (33)$$

In climate, the logistic equation is important for Lorenz's [52] error growth model: where $x(t)$ is the algebraic forecast error at time t and a is the linear growth rate.



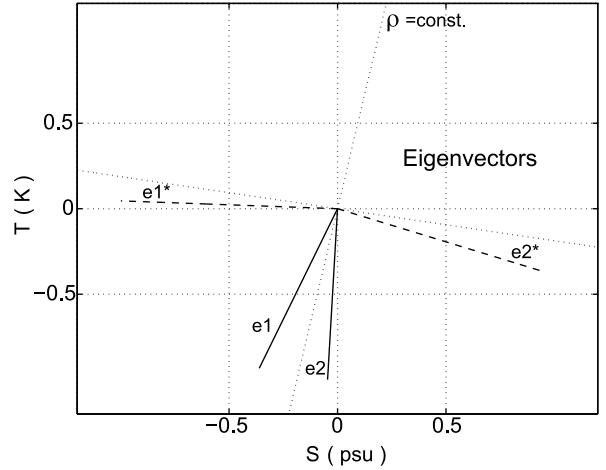
Abrupt Climate Change Modeling, Figure 8

The basin of attraction (white area) and the dynamics in the thermohaline phase space. With initial conditions outside the gray area, the trajectories converge asymptotically to the origin corresponding to the thermally driven solution of the THC. Due to the fast thermal response during the first decade of relaxation, the distance of the trajectories from zero can increase temporarily

It is useful to analyze the dynamics in the phase space spanned by temperature and salinity anomalies and investigate the model sensitivity under anomalous high latitude forcing, induced as an initial perturbation. The lines in Fig. 8 are phase space trajectories after perturbations of different magnitude have been injected into the North Atlantic. We notice that for most trajectories, the distances from zero (0, 0) increase temporarily, where the maximal distance from zero is after a decade. After about 10 years the trajectories in Fig. 8 point into a “mixed temperature/salinity direction”, denoted further as e_1 .

Figure 8 implies that the adjustment of the THC involves two phases: A fast thermal response and a slower response on the e_1 -direction. The vector e_1 is identical with the most unstable mode in the system. Because the scaling function $\gamma(t)$ acts upon both temperature and salinity (32), the evolution of the non-linear model can be well characterized by the eigenvectors of the matrix A , which is discussed in the following.

In our system, the operator A of the box model is found to be non-normal, and the eigenvectors are not orthogonal. One eigenvalue (e_2) is closely related to temperature anomalies, whereas the other (e_1) is a “mixed temperature/salinity eigenvector” (Fig. 9). The eigenvectors of the



Abrupt Climate Change Modeling, Figure 9

Eigenvectors e_1, e_2 , and adjoint eigenvectors e_1^*, e_2^* of the tangent linear operator A^\dagger . The dotted lines show the line of constant density and the perpendicular

adjoint matrix A^\dagger are denoted by e_1^* and e_2^* , respectively. For the non-normal matrix A , the eigenvectors of A and A^\dagger do not coincide, but fulfill the “biorthogonality condition”:

$$e_1^* \perp e_2 \quad \text{and} \quad e_2^* \perp e_1. \quad (34)$$

Both eigenvalues $\lambda_{1,2}$ are real and negative. Because of $\lambda_2 < \lambda_1$, the first term dominates for long time scales and the second for short time scales. Using the biorthogonality condition, we get furthermore the coefficients

$$c_i = \frac{\langle e_i^* | x_0 \rangle}{\langle e_i^* | e_i \rangle} \quad \text{for } i = 1, 2. \quad (35)$$

A perturbation is called “optimal”, if the initial error vector has minimal projection onto the subspace with fastest decaying perturbations, or equivalently if the coefficient c_1 is maximal. This is according to (35) equivalent to x_0 pointing into the direction of e_1^* . This unit vector e_1^* is called the “biorthogonal” [66] to the most unstable eigenvector e_1 which we want to excite. In order to make a geometrical picture for the mathematical considerations, assume that the tail of the vector x_0 is placed on the e_1 -line and its tip on the e_2 -line. This vector is stretched maximally because the tail decays to zero quickly, whereas the tip is hardly unchanged due to the larger eigenvalue λ_1 . The most unstable mode e_1 and its biorthogonal e_1^* differ greatly from each other, and the perturbation that optimally excites the mode bears little resemblance to the mode itself.

It is remarkable that the optimal initial perturbation vector e_1^* does not coincide with a perturbation in sea surface density at high latitudes, which would reside on the dotted line perpendicular to $\rho = \text{const}$ in Fig. 9. Even when using a space spanned by $(\alpha T, \beta S)$ instead of (T, S) , to take into account the different values for the thermal and haline expansion coefficients, vector e_1^* is much more dominated by the scaled salinity anomalies than the temperature anomalies of the high latitudinal box.

Numerical simulations by Manabe and Stouffer [57] showed, for the North Atlantic, that between two and four times the preindustrial CO_2 concentration, a threshold value is passed and the thermohaline circulation ceases completely. One other example of early Holocene rapid climate change is the “8200-yr BP” cooling event recorded in the North Atlantic region possibly induced by freshwater. One possible explanation for this dramatic regional cooling is a shutdown in the formation of deep water in the northern North Atlantic due to freshwater input caused by catastrophic drainage of Laurentide lakes [4,46]. The theoretic considerations and these numerical experiments suggest that formation of deep water in the North Atlantic is highly sensitive to the freshwater forcing.

Resonance in an Atmospheric Circulation Model

An atmospheric general circulation model PUMA [26] is applied to the problem. The model is based on the multi-level spectral model described by Hoskins and Simmons [35]. For our experiments we chose five vertical levels and a T21 horizontal resolution. PUMA belongs to the class of models of intermediate complexity [12]; it has been used to understand principle feedbacks [56], and dynamics on long time scales [76]. For simplicity, the equations are scaled here such that they are dimensionless. The model is linearized about a zonally symmetric mean state providing for a realistic storm track at mid-latitudes [27]. In a simplified version of the model and calculating the linear model A with $n = 214$, one can derive the pseudospectrum. Figure 10 indicates resonances besides the poles (the eigenvalues) indicated by crosses. The $\text{Im}(z)$ -axis shows the frequencies, the $\text{Re}(z)$ -axis the damping/amplification of the modes. Important modes for the climate system are those with $-0.5 < \text{Im}(z) < 0.5$ representing planetary Rossby waves. The basic feature is that transient growth of initially small perturbations can occur even if all the eigenmodes decay exponentially. Mathematically, an arbitrary matrix A can be decomposed as a sum

$$A = D + N \quad (36)$$

where A is diagonalizable, and N is nilpotent (there ex-

ists an integer $q \in \mathbb{N}$ with $N^q = 0$), and D commutes with N (i. e. $DN = NA$). This fact follows from the Jordan–Chevalley decomposition theorem. This means we can compute the exponential of $(A t)$ by reducing to the cases:

$$\exp(At) = \exp((D + N)t) = \exp(Dt) \exp(Nt) \quad (37)$$

where the exponential of Nt can be computed directly from the series expansion, as the series terminates after a finite number of terms. Basically, the number $q \in \mathbb{N}$ is related to the transient growth of the system ($q = 1$ means no transient growth).

The resonant structures are due to the mode interaction: It is not possible to change one variable without the others, because they are not orthogonal. Interestingly, one can also compute the A^\dagger model, showing the optimal perturbation of a mode e_i through its biorthogonal vector (35).

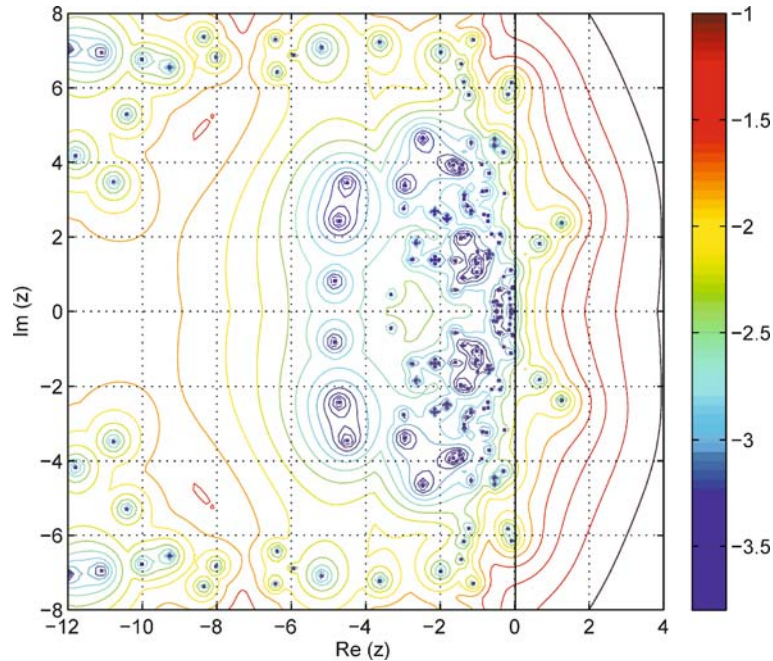
The analysis indicates that non-normality of the system is a fundamental feature of the atmospheric dynamics. This has consequences for the error growth dynamics, and instability of the system, e. g. [48,66]. Similar features are obtained in shear flow systems [71,87] and other hydrodynamic applications. This transient growth mechanism is important for both initial value and forced problems of the climate system.

Future Directions

Until now, details of abrupt climate change are not well known to accurately predict it. With better information, the society could take more confident action to reduce the potential impact of abrupt changes on agriculture, water resources, and the built environment, among other impacts. A better understanding of sea-ice and glacier stability, land-surface processes, and atmospheric and oceanic circulation patterns is needed. Moreover, to effectively use any additional knowledge of these and other physical processes behind abrupt climate change, more sophisticated ways of assessing their interactions must be developed, including:

Better models. At present, the models used to assess climate and its impacts cannot simulate the size, speed, and extent of past abrupt changes, let alone predict future abrupt changes. Efforts are needed to improve how the mechanisms driving abrupt climate change are represented in these models and to more rigorously test models against the climate record.

More theory. There are concepts to find the underlying dynamical system, to derive a theory from a high-order to low-order description similar to what is done in



Abrupt Climate Change Modeling, Figure 10

Contours of $\log_{10}(1/\epsilon)$. The figure displays resonant structures of the linearized atmospheric circulation model. The modes extend to the right half plane and are connected through resonant structures, indicating the transient growth mechanism inherent in atmospheric dynamics

statistical physics (Mori–Zwanzig approach [63,94], Master equation), or in stochastic differential equations. A systematic reduction of the complex system into fewer degrees of freedom shall bring a deeper level of understanding about the underlying physics. A systematic approach was suggested by Saltzman [77]. Spectral and pseudo-spectral concepts have not been used too much in climate theory. There is a variety of phenomenological stochastic models in which non-linearity and fluctuations coexist, and in which this coexistence leads to interesting phenomena that would not arise without the complex interplay.

Paleoclimatic data. More climate information from the distant past would go a long way toward strengthening our understanding of abrupt climate changes and models of past climate. In particular, an enhanced effort is needed to expand the geographic coverage, temporal resolution, and variety of paleoclimatic data. Although the present climate has no direct analogon to the past [54], the dynamical interpretation of data will improve the understanding of thresholds and non-linearities in the Earth system.

Appropriate statistical tools. Because most statistical calculations at present are based on the assumption that climate is not changing but is stationary, they have limited value for non-stationary (changing) climates and for climate-related variables that are often highly skewed

by rapid changes over time such as for abrupt-change regimes. Available statistical tools themselves need to be adapted or replaced with new approaches altogether to better reflect the properties of abrupt climate change.

Synthesis. Physical, ecological, and human systems are complex, non-linear, dynamic and imperfectly understood. Present climate change is producing conditions outside the range of recent historical experience and observation, and it is unclear how the systems will interact with and react to further climate changes. Hence, it is crucial to be able to better understand and recognize abrupt climate changes quickly. This capability will involve improved monitoring of parameters that describe climatic, ecological, and economic systems. Some of the desired data are not uniquely associated with abrupt climate change and, indeed, have broad applications. Other data take on particular importance because they concern properties or regions implicated in postulated mechanisms of abrupt climate change. Research to increase our understanding of abrupt climate change should be designed specifically within the context of the various mechanisms thought to be involved. Focus is required to provide data for process studies from key regions where triggers of abrupt climate change are likely to occur, and to obtain reliable time series of climate indicators that play crucial roles in the postu-

lated mechanisms. Observations could enable early warning of the onset of abrupt climate change. New observational techniques and data-model comparisons will also be required.

Bibliography

Primary Literature

- Alley RB, Anandakrishnan S, Jung P (2001) Stochastic resonance in the North Atlantic. *Paleoceanogr* 16:190–198
- Alley RB, Marotzke J, Nordhaus W, Overpeck J, Peteet D, Pielke R Jr, Pierrehumbert R, Rhines P, Stocker T, Talley L, Wallace JM (2002) Abrupt Climate Change: Inevitable Surprises. US National Academy of Sciences, National Research Council Committee on Abrupt Climate Change, National Academy Press, Washington
- Alverson K, Oldfield F (2000) Abrupt Climate Change. In: Joint Newsletter of the Past Global Changes Project (PAGES) and the Climate Variability and Predictability Project (CLIVAR), vol 8, no 1. Bern, pp 7–10
- Barber DC, Dyke A, Hillaire-Marcel C, Jennings AE, Andrews JT, Kerwin MW, Bilodeau G, McNeely R, Southon J, Morehead MD, Gagnon JM (1999) Forcing of the cold event of 8,200 years ago by catastrophic drainage of Laurentide lakes. *Nature* 400:344–348
- Bartoli G, Sarnthein M, Weinelt M, Erlenkeuser H, Garbe-Schönberg D, Lea DW (2005) Final closure of Panama and the onset of northern hemisphere glaciation. *Earth Planet Sci Lett* 237:33–44
- Bender M, Malaize B, Orchardo J, Sowers T, Jouzel J (1999) Mechanisms of Global Climate Change. Clark P et al (eds) AGU 112:149–164
- Benzi R, Parisi G, Sutera A, Vulpiani A (1982) Stochastic resonance in climatic change. *Tellus* 34:10
- Berger A, Loutre MF (1991) Insolation values for the climate of the last 10 million years. *Quat Sci Rev* 10:297–317
- Berger WH, Labeyrie LD (1987) Abrupt Climatic Change, Evidence and Implications. NATO ASI Series, Series C, Mathematical and Physical Sciences, vol 216. D Reidel, Dordrecht, pp 425
- Broecker WS et al (1985) Does the Ocean-atmosphere System Have More than One Stable Mode of Operation? *Nature* 315:21–26
- Bryan F (1986) High Latitude Salinity Effects and Inter-hemispheric Thermohaline Circulations. *Nature* 323:301–304
- Claussen M, Mysak LA, Weaver AJ, Crucifix M, Fichet T, Loutre M-F, Weber SL, Alcamo J, Alexeev VA, Berger A, Calov R, Ganopolski A, Goosse H, Lohmann G, Lunkeit F, Mokhov II, Petoukhov V, Stone P, Wang Z (2002) Earth System Models of Intermediate Complexity: Closing the Gap in the Spectrum of Climate System Models. *Clim Dyn* 18:579–586
- CLIMAP project members (1976) The surface of the ice age Earth. *Science* 191:1131–1137
- Coxall HK, Wilson PA, Pälike H, Lear CH, Backman J (2005) Rapid stepwise onset of Antarctic glaciation and deeper calcite compensation in the Pacific Ocean. *Nature* 433:53–57. doi:10.1038/nature03135
- Crowley TJ (1992) North Atlantic deep water cools the southern hemisphere. *Paleoceanogr* 7:489–497
- Daubechies I (1992) Ten Lectures on Wavelets. Society for Industrial and Applied Mathematics (SIAM). CBMS-NSF Regional Conference Series in Applied Mathematics, vol 61, Philadelphia
- DeConto RM, Pollard D (2003) Rapid Cenozoic glaciation of Antarctica induced by declining atmospheric CO₂. *Nature* 421:245–249. doi:10.1038/nature01290
- DeMenocal et al (2000) Abrupt onset and termination of the African Humid Period: Rapid climate response to gradual insolation forcing. *Quat Sci Rev* 19:347–361
- Diaz HF, Quayle RG (1980) The climate of the United States since 1895: spatial and temporal changes. *Mon Wea Rev* 108:149–226
- Dijkstra HA, Te Raa L, Weijer W (2004) A systematic approach to determine thresholds of the ocean's thermohaline circulation. *Tellus* 56A(4):362
- Dima M, Lohmann G (2002) Fundamental and derived modes of climate variability. Application to biennial and interannual timescale. *Tellus* 56A:229–249
- Dima M, Lohmann G (2007) A hemispheric mechanism for the Atlantic Multidecadal Oscillation. *J Clim* 20:2706–2719
- EPICA Community Members (2006) One-to-one coupling of glacial climate variability in Greenland and Antarctica. *Nature* 444:195–198. doi:10.1038/nature05301
- Fairbanks RG (1989) A 17,000-year glacio-eustatic sea level record: influence of glacial melting rates on the Younger Dryas event and deep-ocean circulation. *Nature* 342:637–642
- Flohn H (1986) Singular events and catastrophes now and in climatic history. *Naturwissenschaften* 73:136–149
- Fraedrich K, Kirk E, Lunkeit F (1998) Portable University Model of the Atmosphere. DKRZ Report 16, Hamburg
- Frisius T, Lunkeit F, Fraedrich K, James IN (1998) Storm-track organization and variability in a simplified atmospheric global circulation model. *Q J R Meteorol Soc* 124:1019–1043
- Fu C, Diaz HF, Dong D, Fletcher JO (1999) Changes in atmospheric circulation over northern hemisphere oceans associated with the rapid warming of the 1920s. *Int J Climatol* 19(6):581–606
- Ganopolski A, Rahmstorf S (2001) Rapid changes of glacial climate simulated in a coupled climate model. *Nature* 409:153–158
- Ganopolski A, Rahmstorf S (2002) Abrupt glacial climate changes due to stochastic resonance. *Phys Rev Lett* 88(3):038501
- Ganopolski A, Kubatzki C, Claussen M, Brovkin V, Petoukhov V (1998) The influence of vegetation-atmosphere-ocean interaction on climate during the mid-Holocene. *Science* 280:1916
- Hasselmann K (1976) Stochastic climate models, Part 1, Theory. *Tellus* 28:289–485
- Hays JD, Imbrie J, Shackleton NJ (1976) Variations in the Earth's Orbit: Pacemaker of the Ice Ages. *Science* 194:1121–1132
- Henderson GM, Slowey NC (2000) Evidence from U-Th dating against Northern Hemisphere forcing of the penultimate deglaciation. *Nature* 404:61–66
- Hoskins BJ, Simmons AJ (1975) A multi-layer spectral model and the semi-implicit method. *Q J R Meteorol Soc* 101:1231–1250
- Huybers P, Wunsch C (2005) Obliquity pacing of the late Pleistocene glacial terminations. *Nature* 434:491–494. doi:10.1038/nature03401
- Imbrie J, Imbrie JZ (1980) Modeling the climatic response to orbital variations. *Science* 207:943–953

38. IPCC (2007) Summary for Policymakers. In: Climate change 2007: The Physical Science Basis. Contribution of Working Group I to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change. Cambridge University Press, Cambridge and New York
39. Iwashima T, Yamamoto R (1986) Time-space spectral model of low order barotropic system with periodic forcing. *J Meteorol Soc Jpn* 64:183–196
40. Kennett JP, Houtz RE, Andrews PB, Edwards AE, Gostin VA, Hajos M, Hampton M, Jenkins DG, Margolis SV, Ovenshine AT, Perch-Nielsen K (1974) Development of the circum-Antarctic current. *Science* 186:144–147
41. Knorr G, Lohmann G (2003) Southern Ocean Origin for the resumption of Atlantic thermohaline circulation during deglaciation. *Nature* 424:532–536
42. Knorr G, Lohmann G (2007) Rapid transitions in the Atlantic thermohaline circulation triggered by global warming and meltwater during the last deglaciation. *Geochem Geophys Geosyst* 8(12), Q12006:1–22. doi:10.1029/2007GC001604
43. Kwasniok F, Lohmann G (2008) Underlying Dynamics of Glacial Millennial-Scale Climate Transitions Derived from Ice-Core Data. *Phys Rev E* (accepted)
44. Lawver LA, Gahagan LM (2003) Evolution of Cenozoic seaways in the circum-Antarctic region. *Palaeogeography, Palaeoclimatology, Palaeoecology* 198:11–37. doi:10.1016/S0031-0182(03)00392-4
45. Lisiecki LE, Raymo ME (2005) A Pliocene-Pleistocene stack of 57 globally distributed benthic O-18 records. *Paleoceanography* 20:PA1003. doi:10.1029/2004PA001071
46. Lohmann G (2003) Atmospheric and oceanic freshwater transport during weak Atlantic overturning circulation. *Tellus* 55A: 438–449
47. Lohmann G, Gerdes R (1998) Sea ice effects on the Sensitivity of the Thermohaline Circulation in simplified atmosphere-ocean-sea ice models. *J Climate* 11:2789–2803
48. Lohmann G, Schneider J (1999) Dynamics and predictability of Stommel's box model: A phase space perspective with implications for decadal climate variability. *Tellus* 51A:326–336
49. Lohmann G, Schulz M (2000) Reconciling Boelling warmth with peak deglacial meltwater discharge. *Paleoceanography* 15:537–540
50. Lorenz EN (1963) Deterministic nonperiodic flow. *J Atmos Sci* 20:130–141
51. Lorenz EN (1976) Nondeterministic theories of climatic change. *Quat Res* 6:495–506
52. Lorenz EN (1982) Atmospheric predictability experiments with a large numerical model. *Tellus* 34:505–513
53. Lorenz EN (1990) Can chaos and intransitivity lead to interannual variability? *Tellus* 42A:378–389
54. Lorenz S, Lohmann G (2004) Acceleration technique for Milankovitch type forcing in a coupled atmosphere-ocean circulation model: method and application for the Holocene. *Climate Dyn* 23(7–8):727–743. doi:10.1007/s00382-004-0469-y
55. Lund R, Reeves J (2002) Detection of undocumented change-points: A revision of the two-phase regression model. *J Climate* 15:2547–2554
56. Lunkeit F, Bauer SE, Fraedrich K (1998) Storm tracks in a warmer climate: Sensitivity studies with a simplified global circulation model. *Clim Dyn* 14:813–826
57. Manabe S, Stouffer RJ (1993) Century-scale effects of increased atmospheric CO₂ on the ocean-atmosphere system. *Nature* 364:215–218
58. Maraun D, Kurths J (2004) Cross wavelet analysis. Significance testing and pitfalls. *Nonlin Proc Geoph* 11:505–514
59. Maraun D, Kurths J (2005) Epochs of phase coherence between El Niño/Southern Oscillation and Indian monsoon. *Geophys Res Lett* 32:L15709. doi:10.1029/2005GL023225
60. Maslin MA, Li XS, Loutre MF, Berger A (1998) The contribution of orbital forcing to the progressive intensification of Northern Hemisphere Glaciation. *Quat Sci Rev* 17:411–426
61. Maslin MA, Ridgwell A (2005) Mid-Pleistocene Revolution and the eccentricity myth. Special Publication of the Geological Society of London 247:19–34
62. Milankovitch M (1941) *Kanon der Erdbestrahlung*. Royal Serb Acad Spec Publ, Belgrad, 132, Sect. Math Nat Sci 33:484
63. Mori H (1965) A Continued-Fraction Representation of the Time-Correlation Functions *Prog Theor Phys* 33:423–455. doi:10.1143/PTP.34.399
64. North Greenland Ice Core Project members (2004) High-resolution record of Northern Hemisphere climate extending into the last interglacial period. *Nature* 431:147–151
65. Paillard D (1998) The timing of Pleistocene glaciations from a simple multiple-state climate model. *Nature* 391:378–381
66. Palmer TN (1996) Predictability of the atmosphere and oceans: From days to decades. In: Anderson DTA, Willebrand J (eds) Large-scale transport processes in oceans and atmosphere. NATO ASI Series 44. Springer, Berlin, pp 83–155
67. Parker DE, Jones PD, Folland CK, Bevan A (1994) Interdecadal changes of surface temperature since the late nineteenth century. *J Geophys Res* 99:14,373–14,399
68. Peixoto JP, Oort AH (1992) *Physics of Climate*. American Institute of Physics, New York, p 520
69. Petit JR, Jouzel J, Raynaud D, Barkov NI, Barnola JM, Basile I, Bender M, Chappellaz J, Davis M, Delaygue G, Delmotte M, Kotlyakov VM, Legrand M, Lipenkov VY, Lorius C, Pepin L, Ritz C, Saltzman E, Stievenard M (1999) Climate and atmospheric history of the past 420,000 years from the Vostok ice core, Antarctica. *Nature* 399:429–436
70. Raymo M, Ganley K, Carter S, Oppo DW, McManus J (1998) Millennial-scale climate instability during the early Pleistocene epoch. *Nature* 392:699–701
71. Reddy SC, Schmidt P, Henningson D (1993) Pseudospectra of the Orr-Sommerfeld operator. *SIAM J Appl Math* 53:15–47
72. Rial JA (1999) Pacemaking the Ice Ages by Frequency Modulation of Earth's Orbital Eccentricity. *Science* 285:564–568
73. Rial JA (2004) Abrupt Climate Change: Chaos and Order at Orbital and Millennial Scales. *Glob Plan Change* 41:95–109
74. Ridgwell AJ, Watson AJ, Raymo ME (1999) Is the spectral signature of the 100 Kyr glacial cycle consistent with a Milankovitch origin? *Paleoceanography* 14:437–440
75. Rogers JC (1985) Atmospheric circulation changes associated with the warming over the northern North Atlantic in the 1920s. *J Climate Appl Meteorol* 24:1303–1310
76. Romanova V, Lohmann G, Grosfeld K, Butzin M (2006) The relative role of oceanic heat transport and orography on glacial climate. *Quat Sci Rev* 25:832–845. doi:10.1016/j.quascirev.2005.07.007
77. Saltzman (2002) Dynamical Paleoclimatology. Generalized Theory of Global Climate Change. In: International Geophysics Series, vol 80. Harcourt-Academic Press (Elsevier Science), San Diego, p 354

78. Schulz M, Paul A, Timmermann A (2004) Glacial-Interglacial Contrast in Climate Variability at Centennial-to-Millennial Timescales: Observations and Conceptual Model. *Quat Sci Rev* 23:2219
79. Seidel DJ, Lanzante JR (2004) An assessment of three alternatives to linear trends for characterizing global atmospheric temperature changes. *J Geophys Res* 109:D14108. doi:10.1029/2003JD004414
80. Stocker TF (1998) The seesaw effect. *Science* 282:61–62
81. Stocker TF, Johnsen SJ (2003) A minimum thermodynamic model for the bipolar seesaw. *Paleoceanography* 18(4):1087
82. Stommel H (1961) Thermohaline Convection with Two Stable Regimes of Flow. *Tellus* 13:224–230
83. Tiedemann R, Sarnthein M, Shackleton NJ (1994) Astronomic time scale for the Pliocene Atlantic $\delta^{18}\text{O}$ and dust flux records of Ocean Drilling Program site 659. *Paleoceanography* 9:19–638
84. Timmermann A, Lohmann G (2000) Noise-Induced Transitions in a simplified model of the thermohaline circulation. *J Phys Oceanogr* 30(8):1891–1900
85. Timmermann A, Oberhuber J, Bracher A, Esch M, Latif M, Roeckner E (1999) Increased El Niño frequency in a climate model forced by future greenhouse warming. *Nature* 398:694–696
86. Torrence C, Compo G (1998) A practical guide to wavelet analysis. *Bull Amer Meteor Soc* 79:61–78
87. Trefethen LN, Trefethen AE, Reddy SC, Driscoll TA (1993) Hydrodynamic stability without eigenvalues. *Science* 261:578–584
88. Trenberth KE (1990) Recent observed interdecadal climate changes in the Northern Hemisphere. *Bull Am Meteorol Soc* 71:988–993
89. Uhlenbeck GE, Ornstein LS (1930) On the theory of Brownian Motion. *Phys Rev* 36:823–841
90. Wunsch C (1999) The interpretation of short climate records, with comments on the North Atlantic and Southern Oscillation. *Bull Amer Meteor Soc* 80:245–255
91. Wunsch C (2004) Quantitative estimate of the Milankovitch-forced contribution to observed Quaternary climate change. *Quat Sci Rev* 23(9–10):1001–1012
92. Yamamoto R, Iwashima T, Sanga NK (1985) Climatic jump: a hypothesis in climate diagnosis. *J Meteorol Soc Jpn* 63:1157–1160
93. Zachos J, Pagani M, Sloan L, Thomas E, Billups K (2001) Trends, Rhythms, and Aberrations in Global Climate 65 Ma to Present. *Science* 292(5517):686–693
94. Zwanzig R (1980) Thermodynamic modeling of systems far from equilibrium. In: Garrido L (ed) *Lecture Notes in Physics* 132, in *Systems Far From Equilibrium*. Springer, Berlin

Books and Reviews

- Dijkstra HA (2005) *Nonlinear Physical Oceanography*, 2nd revised and extended edition. Springer, New York, pp 537
- Hansen J, Sato M, Kharecha P (2007) Climate change and trace gases. *Phil Trans R Soc A* 365:1925–1954. doi:10.1098/rsta.2007.2052
- Lockwood JG (2001) Abrupt and sudden climate transitions and fluctuations: a review. *Int J Climat* 21:1153–1179
- Rial JA, Pielke RA Sr, Beniston M, Claussen M, Canadell J, Cox P, Held H, deNoblet-Ducudre, Prinn R, Reynolds J, Salas JD (2004)

Nonlinearities, Feedbacks and Critical Thresholds Within the Earth's Climate System. *Clim Chang* 65:11–38

Ruddiman WF (2001) *Earth's Climate. Past and Future*. WH Freeman, New York, p 465

Stocker TF (1999) Abrupt climate changes from the past to the future—a review. *Int J Earth Sci* 88:365–374

Acceleration Mechanisms

DON B. MELROSE

School of Physics, University of Sydney, Sydney, Australia

Article Outline

Glossary

Definition of the Subject

Introduction

Stochastic Acceleration

Diffusive Shock Acceleration

DSA at Multiple Shocks

Applications of DSA

Resonant Scattering

Acceleration by Parallel Electric Fields

Other Acceleration Mechanisms

Future Directions

Appendix A: Quasilinear Equations

Bibliography

Glossary

Acceleration mechanism Any process, or sequence of processes, that results in a small fraction of the particles in a plasma each gaining an energy greatly in excess of the thermal energy.

Heliosphere The region around the Sun populated by plasma escaping from the Sun in the solar wind, and extending to well beyond the orbits of the planets.

Magnetosphere The region around a planet or star populated by plasma on magnetic field lines tied to the planet or star.

Resonance The gyroresonance condition (6) corresponds to the wave frequency being an integral, s , multiple of the particle gyrofrequency in the rest frame of the particle gyrocenter; $s = 0$ is also called the Cerenkov condition.

Shock A fast magnetoacoustic shock is a discontinuity in plasma flow that propagates into the upstream plasma faster than the Alfvén speed, compressing both the plasma and the magnetic field.

Solar corona The hot ($> 10^6$ K) outer atmosphere of the Sun.

Solar flare An explosion in the solar corona that leads to copious acceleration of fast particles.

Definition of the Subject

A typical astrophysical or space plasma consists of thermal plasma components, a magnetic field, various non-thermal distributions of fast particles and turbulence involving various kinds of waves. An acceleration mechanism is the process, or sequence of processes, by which the fast particles gain their energy. There is no single, universal acceleration mechanism, and it is now recognized that the data require a rich variety of different acceleration mechanisms operating under different conditions in different space and astrophysical plasmas. Three generic mechanisms are emphasized in this review: stochastic acceleration, diffusive shock acceleration and acceleration by parallel electric fields.

Introduction

The major observational discoveries in this field were almost all unexpected and initially quite baffling. The first notable discovery was that of cosmic rays (CRs), by Victor Hess in 1912. Hess measured the rate of ion production inside a hermetically sealed container in a balloon flight to an altitude of 5300 m, and concluded that “The results of my observation are best explained by the assumption that a radiation of very great penetrating power enters our atmosphere from above”. By the late 1920s CRs were identified as charged particles, and not photons, and due to their isotropy it was inferred that they come from the Galaxy. Perhaps the earliest idea on the origin of Galactic CRs (GCRs) has stood the test of time: in 1934 Baade and Zwicky [8] proposed that GCRs are accelerated in supernova explosions. Supernovae are the only adequate source of the energy needed to replace GCRs, whose lifetime in the Galaxy is about 10^7 yr [34]. The observed momentum distribution of GCRs is a power law, $f(p) \propto p^{-b}$, with more recent observations implying $b = 4.7$ below a knee in the distribution at $p \approx 10^{15}$ eV/c with the spectrum steepening at higher momenta; Axford [6] referred to these as GCR1 and GCR2, respectively.

The next major discoveries followed the development of radio astronomy in the 1940s. This led to the identification of a wide class of sources radiating by a nonthermal mechanism, identified in the late 1940s as synchrotron radiation. The presence of highly relativistic electrons in synchrotron sources raises the question of how they are accelerated. Subsequent development of X-ray and gamma-ray

astronomy placed severe constraints on the acceleration of the relativistic electrons in synchrotron sources, due to the short life times of electrons radiating at high frequencies. Solar radio astronomy led to the identification of several different types of radio bursts associated with solar flares, including type III bursts due to ≈ 10 keV electrons escaping from the Sun, type II bursts associated with shocks that result from the flare and type I and type IV emission that involve radio continua. All these phenomena require specific acceleration mechanisms to explain the presence of the radio-emitting electrons.

Space physics began in the late 1950s with the discovery of the Earth’s Van Allen radiation belts, raising the question of how these energetic electrons and ions trapped in the Earth’s magnetosphere are accelerated. Further space exploration led to the identification of energetic particle distributions in essentially all space plasmas: other planetary magnetospheres, the solar wind, and more specifically associated with bow shocks and interplanetary shocks. Observations from spacecraft provide in situ data on particles and fields, in the Earth’s magnetosphere, in other planetary magnetospheres, and throughout the heliosphere. In the case of the magnetosphere, Earth-based data on precipitation of particles, notably in the auroral zones, and on radio waves originating from the magnetosphere complement the spacecraft data on particles and waves in situ.

Current-day ideas on acceleration mechanisms are dominated by three generic mechanisms, plus a rich variety of specific mechanisms of various kinds. Two of the generic mechanisms had forerunners in two papers by Fermi [30,31]. In the first of these [30], Fermi suggested that the acceleration of GCRs is due to reflections off moving interstellar clouds. CRs gain energy in head-on collisions with clouds, and lose energy in overtaking collisions; a net average gain results from head-on collisions being more probable than overtaking collisions. This model is recognized as the first example of the generic mechanism of stochastic acceleration. In the second paper [31], Fermi proposed acceleration between ‘closing jaws’ in which all reflections are head-on and hence energy-increasing. This idea is an essential ingredient in the astrophysically most important of the generic mechanisms: diffusive shock acceleration (DSA). The third generic mechanism is acceleration by a parallel electric field, E_{\parallel} ; in particular, there is compelling evidence for auroral electrons resulting from such acceleration. Both stochastic acceleration and DSA require that the particles be scattered efficiently in order to be accelerated, and an important step in the development of the theory is that very efficient pitch-angle scattering of fast particles can result from gyroresonant inter-

actions with low-frequency waves. The following remarks relate to the three generic mechanisms and to resonance acceleration.

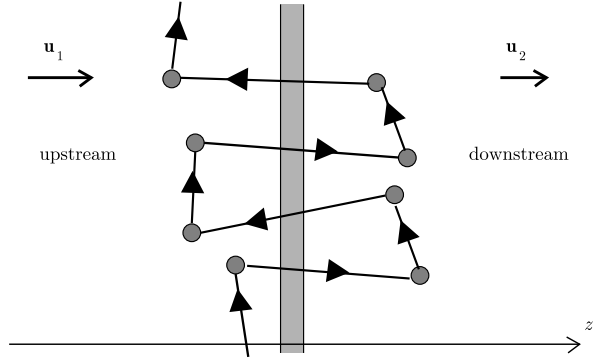
Stochastic Acceleration

The original version of Fermi mechanism [30], involving reflection off magnetized interstellar clouds, is too slow to account for GCRs, basically because such collisions are so infrequent. The acceleration mechanism is more efficient for more frequent, smaller energy changes than for less frequent, larger energy changes. This led to the recognition that a more efficient Fermi-type acceleration can result from MHD turbulence [23,50,82,83,95]. One form of the idea is that the compression and rarefaction of B , associated with the MHD turbulence, conserves the magnetic moment, p_{\perp}^2/B , and when combined with a scattering process that maintains the isotropy of the particles, this implies a ‘betatron’ acceleration, or ‘magnetic pumping’ [15,88]. A closer analogy with Fermi acceleration is due to the small fraction of the particles that reflect off the compressions in B , which propagate at the Alfvén speed, v_A . Besides the betatron-type acceleration and the reflection off moving compressions, there is also a transit acceleration [90] due to particles diffusing through the compressions and rarefactions. All these energy changes contribute to stochastic acceleration by MHD turbulence [57].

The mathematical description of Fermi-type acceleration is in terms of isotropic diffusion in momentum space. In the early literature stochastic acceleration was treated using a Fokker–Planck approach that includes a term that describes a systematic acceleration and a term that describes a diffusion in energy. It was shown by Tverskoi [98] that for Fermi’s original mechanism this equation reduces to an isotropic diffusion in momentum space. The same equation is derived for acceleration by isotropic turbulence [40,57,93]. However, MHD turbulence is never isotropic: it consists of a spectrum of MHD waves, in both the (non-compressive) Alfvén mode and the (compressive) fast magnetoacoustic mode. It is now assumed that the isotropy is due to efficient resonant scattering, with no implication that the turbulence itself is isotropic.

Diffusive Shock Acceleration

Fermi [31] suggested that the acceleration can be efficient if all reflections are head-on. A simple example is when two magnetized clouds approach each other, and a particle bounces back and forth reflecting off the approaching clouds. A related example is for a shock propagating into a closed magnetic loop, where a particle trapped in the loop bounces back and forth being reflected head-on by



Acceleration Mechanisms, Figure 1

Diffusive shock acceleration is illustrated: the shaded vertical region is the shock, the circular blobs denote idealized scattering centers, and the solid line with arrows denotes the path of an idealized fast particle

the approaching shock [69,106]. However, such examples are constrained by the fact that acceleration ceases when the approaching structures meet. In 1977–8 several different authors [7,10,17,56] recognized that a single shock can accelerate particles efficiently, provided that the particles are scattered on either side of the shock, as illustrated in Fig. 1. This mechanism is called diffusive shock acceleration (DSA).

To understand why DSA occurs, first note that the fluid velocity changes discontinuously across the shock. Let the two sides of the shock be labeled 1 and 2. Consider a particle on side 1 about to cross the shock and enter side 2. The scattering centers on side 2 are convected along with the plasma flow (velocities u_1, u_2), and the particle sees them approaching it head on at $|u_1 - u_2|$. Once the particle crosses the shock and is scattered it gains energy due to the oncoming motion of the scatterers. After being scattered a number of times on side 2, the particle can diffuse back to the shock and cross back onto side 1. On doing so, it sees the scattering centers on side 1 approaching it head on at $|u_1 - u_2|$. Again it gains energy on being scattered. DSA requires efficient scattering, and this can be achieved only by resonant scattering.

Resonant Scattering

Both stochastic acceleration and DSA require that the fast particles be scattered such that their distribution remains almost isotropic. The recognition that fast particles are scattered very efficiently developed in two other contexts before its significance to acceleration mechanism became fully appreciated. These developments essentially defined a new field of “plasma astrophysics” that emerged in the mid 1960s.

Efficient scattering is needed to explain precipitation of particles trapped in the magnetosphere. Under quiescent conditions, ions and electrons precipitate steadily (albeit in localized bursts) from the radiation belts, implying that there is a continuous resupply of fast particles. The precipitation is interpreted in terms of pitch-angle scattering of trapped particles into the loss cone. Pitch-angle scattering of ions [24,105] involves ion cyclotron waves, and the scattering of electrons involves whistlers [27,52]. Efficient scattering is also needed to explain the data on GCRs. Both their estimated residence time, $\sim 10^7$ yr, in the Galactic disc and their small observed anisotropy imply that GCRs diffuse slowly through the Galaxy. The net flow speed is of order the Alfvén speed, $v_A \sim 10^{-4}c$. The efficient scattering is attributed to resonant scattering by Alfvén (and magnetoacoustic waves) in the interstellar medium.

Resonant scattering, also called pitch-angle scattering, occurs when the gyroresonance condition is satisfied for a fast particle interacting with a wave whose phase speed is much less than the speed of the particle. To the particle, the wave appears as a spatially periodic oscillation in the magnetic field. The resonance condition corresponds to the wavelength of the magnetic fluctuation being equal to the distance that the particle propagates along the field line in a gyroperiod. As a consequence, the Lorentz force, $q\mathbf{v} \times \delta\mathbf{B}$, due to the magnetic fluctuation, $\delta\mathbf{B}$, in the wave is in the same direction at the same phase each gyroperiod. This causes a systematic change in the momentum of the particle, without any change in energy, corresponding to a change in the pitch angle, $\alpha = \arctan(p_{\parallel}/p_{\perp})$. The sense of the change remains the same until phase coherence is lost. Such interactions cause random changes in α , as a result of which the particles diffuse in α . The resonance condition for a given wave (given ω , k_{\parallel}) defines a resonance curve in velocity ($v_{\parallel}-v_{\perp}$) space or in momentum ($p_{\parallel}-p_{\perp}$) space; this curve is a conic section in general and it reduces to a circle in the limit of slow phase speed. Pitch-angle diffusion corresponds to particles diffusing around the circumference of this circle.

The resonant interaction results in the waves either growing or damping, depending on whether energy is transferred from the particles to the waves, or vice versa. For any given distribution of particles, all the particles that can resonate with a given wave lie on the resonance curve. If the gradient in the distribution function along this curve is an increasing function of ε , more particles lose energy to the wave than gain energy from the wave, and the wave grows. For any anisotropic distribution, there is a choice of the relevant signs (harmonic number $s = \pm 1$, $k_{\parallel}/|k_{\parallel}|$ and $v_{\parallel}/|v_{\parallel}|$) that implies growth of those waves that carry away the excess momentum. The back reaction on the distribu-

tion of particles, called quasilinear relaxation, reduces the anisotropy that causes the wave growth [74]. This implies that the resonant waves needed to scatter the particles can be generated by the anisotropic particles themselves.

In DSA the resonant waves can be generated by a more subtle anisotropy. Upstream of the shock the density of the fast particles decreases with distance from the shock, and the spatial gradient implies a form of anisotropy that can cause the resonant waves to grow. Analogous to the scattering of streaming CRs, the growth of the resonant waves decreases rapidly with increasing particle energy, and some other source of resonant waves is required for higher energy particles. Scattering downstream of the shock is less problematic, with several possible sources of resonant waves, including waves generated in the upstream region and swept back across the shock.

Acceleration by Parallel Electric Fields

There is overwhelming evidence that acceleration by E_{\parallel} occurs in the Earth's auroral zones. Acceleration by E_{\parallel} plays a central role in populating pulsar magnetospheres with electron-positron pairs, and is a possible mechanism for the "bulk energization" mechanism for energetic electrons in solar flares.

Acceleration by E_{\parallel} is necessarily associated with a parallel current, J_{\parallel} : the relevant source term from Maxwell's equations for the energy transfer is $J_{\parallel}E_{\parallel}$. Historically, the existence of J_{\parallel} in the auroral zones is perhaps the oldest idea relevant to the theory of acceleration, having been predicted by Kristian Birkeland in 1903. Understanding acceleration by E_{\parallel} remains problematical. One conceptual difficulty is that astrophysical plasmas are typically described using ideal magnetohydrodynamics (MHD), in which one has $E_{\parallel} = 0$ by hypothesis: theories involving E_{\parallel} are necessarily outside the scope of MHD. A related difficulty is that E_{\parallel} accelerates particles of oppositely signed charge in opposite directions, setting up a current flow that is opposed by inductive effects. Both inductive effects and the charge separation resulting from the current flow tend to suppress the E_{\parallel} and hence the acceleration.

A generic model has emerged for the acceleration by E_{\parallel} of auroral electrons [103]. The model requires a mechanical driver, which sets up an EMF, and a closed circuit around which the current flows. The EMF is made available by magnetic reconnection, which acts like a switch setting up the current flow around the circuit. The energy released is transported away through a Poynting flux, propagating at v_A . This energy is transferred, at least in part, to energetic particles, in an acceleration region, which acts as a load in the circuit. The available po-

tential localizes across the acceleration region, producing the E_{\parallel} .

Suggested astrophysical applications of circuit models tend to be viewed with suspicion, especially because the physics involved is specifically excluded in the assumptions of ideal MHD. Some non-ideal-MHD effect is needed to explain why $E_{\parallel} \neq 0$ develops. One such mechanism is resistivity, which is included in non-ideal or resistive MHD, but the E_{\parallel} allowed by classical resistivity is too weak to play any significant role in the fully ionized plasmas of interest. Currently favored possibilities include double layers and inertial Alfvén waves.

Despite its apparent simplicity, acceleration by E_{\parallel} is the least understood of the three generic mechanisms.

Stochastic Acceleration

Second-order Fermi acceleration [30] is now identified as an archetypical form of stochastic acceleration, and the term Fermi acceleration is sometimes used in a generic sense to describe stochastic acceleration.

Diffusion in Momentum Space

The isotropic diffusive equation in momentum space that describes stochastic acceleration is [57,98,99]

$$\frac{\partial \langle f \rangle(p)}{\partial t} = \frac{1}{p^2} \frac{\partial}{\partial p} \left[p^2 D_{pp}(p) \frac{\partial}{\partial p} \right] \langle f \rangle(p), \quad (1)$$

where $\langle f \rangle(p)$ is the particle distribution function, $f(p)$, averaged over pitch angle. The diffusion coefficient is

$$D_{pp}(p) = \zeta_A \frac{c p^2}{4v} \left(1 - \frac{v_A^2}{v^2} \right)^2, \quad (2)$$

where ζ_A is the acceleration rate. The final factor in (2) does not appear in simpler treatments of Fermi-type acceleration; this factor may be ignored only in the limit of particle speeds much greater than the Alfvén speed, $v \gg v_A$. The meaning of ζ_A is most easily understood by estimating the mean rate of acceleration [72,97]

$$\left\langle \frac{d\varepsilon}{dt} \right\rangle = \frac{1}{p^2} \frac{\partial}{\partial p} [v p^2 D_{pp}(p)], \quad (3)$$

where $\varepsilon = (m^2 c^4 + p_{\parallel}^2 c^2 + p_{\perp}^2 c^2)^{1/2}$ is the energy. In the limit $v \gg v_A$ (3) gives

$$\left\langle \frac{d\varepsilon}{dt} \right\rangle \approx \zeta_A p c, \quad (4)$$

which reduces for highly relativistic particles to the familiar form $\langle d\varepsilon/dt \rangle \approx \zeta_A \varepsilon$. The acceleration rate may be estimated as

$$\zeta_A = \frac{\pi}{4} \bar{\omega} \left(\frac{\delta B}{B} \right)^2, \quad (5)$$

where δB is the magnetic amplitude in the waves and $\bar{\omega}$ is their mean frequency. A remarkable feature of (5) is that, although effective scattering is an essential ingredient in the theory, the acceleration rate is independent of the details of the scattering. The parameters that appear in (5) refer only to the MHD turbulence that is causing the acceleration.

Treatment in Terms of Resonant Interactions

Further insight into Fermi-type acceleration was given by Achterberg [1] who showed that it is equivalent to resonant damping of magnetoacoustic waves in the presence of efficient pitch-angle scattering. The general gyroresonance condition is

$$\omega - s\Omega - k_{\parallel} v_{\parallel} = 0, \quad (6)$$

where ω is the wave frequency, k_{\parallel} is the component of its wave vector parallel to \mathbf{B} , $s = 0, \pm 1, \dots$ is the harmonic number, $\Omega = |q|B/\gamma m$ is the relativistic gyrofrequency of the particle with charge q , mass m , Lorentz factor γ and velocity v_{\parallel} parallel to \mathbf{B} . The effect of gyroresonant interactions on the particles and the waves is described by the quasilinear equations, which are written down in the Appendix. The particular resonant interaction of interest here is at the Cerenkov resonance, $s = 0$. Resonance at $s = 0$ is possible for waves that have a component of their electric field along \mathbf{B} , and this is the case for magnetoacoustic waves, but not for Alfvén waves (in ideal MHD). The resonant interaction alone would cause the particles to diffuse in p_{\parallel} , with p_{\perp} remaining constant. This corresponds to the special case

$$\frac{df(\mathbf{p})}{dt} = \frac{\partial}{\partial p_{\parallel}} \left[D_{\parallel\parallel}(\mathbf{p}) \frac{\partial f(\mathbf{p})}{\partial p_{\parallel}} \right] \quad (7)$$

of the general diffusion equation (33). In the absence of pitch-angle scattering, the distribution of fast particles becomes increasingly anisotropic, favoring p_{\parallel} over p_{\perp} , leading to a suppression of the acceleration. In the presence of efficient pitch-angle scattering, one averages (7) over pitch-angle, leading to the isotropic diffusion equation (3), with $D_{pp}(p) = \frac{1}{2} \int d\mu \mu^2 D_{\parallel\parallel}(\mathbf{p})$, where $\mu = p_{\parallel}/p$ is the cosine of the pitch angle. The resulting expression for $D_{pp}(p)$, given by (2) with (5), confirms that this resonant interaction is equivalent to Fermi acceleration.

Diffusive Shock Acceleration

Diffusive shock acceleration (DSA) is now recognized as the most important acceleration mechanism in astrophysical plasmas: its identification was a major achievement in

the field. Prior to the identification of DSA, it was difficult to explain how acceleration can be as efficient as it appears to be, and why power-law energy spectra are so common. DSA is so efficient that one needs to look for self-regulation processes that limit its efficiency, and DSA implies power-law spectra of the form observed without any additional assumptions.

The treatment of DSA given below is a nonrelativistic, single-particle theory. The assumption that collective effects of the fast particles can be neglected is not necessarily valid: DSA is so efficient that the fast particles can become an important component of the upstream plasma. Once the pressure associated with the fast particles becomes comparable with the thermal pressure, the structure of the shock is modified by this pressure. This can result in the stresses being transferred from the downstream to the upstream plasma primarily through the fast particles, with no discontinuity in the density of the thermal gas [67]. Such nonlinear effects provide a constraint on DSA.

Diffusive Treatment of DSA

Consider a distribution of particles $f(p, z)$ that is averaged over pitch angle and is a function of distance z from a shock in a frame in which the shock is at rest. It is assumed that scattering causes the particles to diffuse in the z direction with diffusion coefficient $\kappa(z)$. The particles are also assumed to be streaming with the streaming speed u . The diffusion is described by

$$\begin{aligned} \frac{\partial f(p, z)}{\partial t} + u \frac{\partial f(p, z)}{\partial z} + \dot{p} \frac{\partial f(p, z)}{\partial p} \\ = \frac{\partial}{\partial z} \left(\kappa(z) \frac{\partial f(p, z)}{\partial z} \right) + Q(p, z) - f_{\text{esc}}(p), \end{aligned} \quad (8)$$

where $Q(p, z)$ is a source term, where the sink term $f_{\text{esc}}(p)$ takes account of escape of particles downstream from the shock, and where the term involving $\dot{p} = -(1/3)p\partial u/\partial z$ determines the energy changes. It is assumed that the speed u changes abruptly across the shock:

$$\begin{aligned} u = \begin{cases} u_1 & \text{for } z < 0 \text{ (upstream),} \\ u_2 & \text{for } z > 0 \text{ (downstream),} \end{cases} \quad (9) \\ \frac{\partial u}{\partial z} = (u_1 - u_2)\delta(z). \end{aligned}$$

A stationary solution of (8) exists when both the source and the sink term are neglected, such that the equation reduces to $u \partial f/\partial z = (\partial/\partial z)(\kappa \partial f/\partial z)$; a general solution is

$$f(p, z) = A + B \exp \left[u \int dz \frac{1}{\kappa(z)} \right], \quad (10)$$

with u constant on either side of the shock, and where A, B are constants. In the upstream region, $z < 0$, one has $u/\kappa(z) \neq 0$, and the solution (10) diverges at $z \rightarrow -\infty$ unless one has $B = 0$ for $z < 0$. Writing $f_{\pm}(p) = \lim_{z \rightarrow \pm\infty} f(p, z)$, one has

$$f(p, z) = \begin{cases} f_{-}(p) + [f_{+}(p) - f_{-}(p)] \exp \left[u_1 \int_0^{\infty} dz \frac{1}{\kappa(z)} \right] & \text{for } z < 0, \\ f_{+}(p) & \text{for } z > 0. \end{cases} \quad (11)$$

On matching the discontinuity in the derivative of this solution with the discontinuity due to the acceleration term, one finds

$$f_{+}(p) = b p^{-b} \int_0^p dp' p'^{(b-1)} f_{-}(p'), \quad b = \frac{3u_1}{u_2 - u_1}, \quad (12)$$

which determines the downstream solution in terms of the upstream solution. That is, if the shock propagates into a region where the distribution is $f_{-}(p)$, then after the shock has passed the distribution is $f_{+}(p)$.

For monoenergetic injection, $f_{-}(p) \propto \delta(p - p_0)$ say, (12) implies $f_{+}(p) \propto p^{-b}$. The power law index, b , is determined by the ratio u_1/u_2 , which is determined by the strength of the shock. In terms of the Mach number $M = u_1/v_{A1}$, where v_{A1} is the Alfvén speed upstream of the shock, and the ratio Γ of specific heats, one finds

$$b = \frac{3r}{r-1} = \frac{3(\Gamma+1)M^2}{2(M^2-1)}, \quad r = \frac{(\Gamma+1)M^2}{2+(\Gamma-1)M^2}, \quad (13)$$

where r is the compression factor across the shock. For $\Gamma = 5/3$ and $M^2 \gg 1$, the maximum strength of the shock is $r = 4$ and the minimum value of the spectral index is $b = 4$.

Alternative Treatment of DSA

An alternative treatment of DSA is as follows. Consider a cycle in which a particle with momentum p crosses the shock from downstream to upstream and back again. Let $\Delta(p)$ be the change in p in a cycle, and $P(p)$ be the probability per cycle of the particle escaping far downstream. Simple theory implies

$$\Delta(p) = \frac{4(r-1)}{3r} \frac{u_1}{v} p, \quad P(p) = \frac{4u_1}{vr}. \quad (14)$$

After one cycle, $p \rightarrow p' = p + \Delta(p)$ implies that the number of particles per unit momentum changes according to $4\pi p^2 f(p) dp \rightarrow 4\pi p'^2 f(p') dp' = [1 -$

$P(p)]4\pi p^2 f(p)dp$. After integration, one imposes the requirement that the downstream flux of particles balances the upstream flux, and a result equivalent to (12) is derived.

This model allows one to estimate the acceleration time, t_a , in terms of the cycle time t_c . One has $t_c = (\lambda_1/u_1 + \lambda_2/u_2) = (\lambda_1 + r\lambda_2)/u_1$, where $\lambda_{1,2}$ are the scattering mean free paths in the upstream and downstream regions, respectively. For isotropic scattering, the scattering mean free path, λ , is related to the spatial diffusion coefficient, κ , by $\kappa = \lambda v/3$, where v is the speed of the particle. The mean free path may be written as $\lambda = \eta r_g$, where r_g is the gyroradius, and it is usually assumed that η is a constant of order unity, called Bohm diffusion. The rate of momentum gain is given by

$$\frac{dp}{dt} = \frac{p}{t_a} - \left(\frac{dp}{dt}\right)_{\text{loss}}, \quad t_a = \frac{u_1^2}{c\tilde{\lambda}}, \quad \tilde{\lambda} = \frac{3r(\lambda_1 + r\lambda_2)}{4(r-1)}, \quad (15)$$

where t_a is an acceleration time, $\tilde{\lambda}$ is a mean scattering free path, and where a loss term is included.

Injection and Preacceleration

DSA requires efficient scattering, and this is attributed to resonant scattering, Sect. “Resonant Scattering”. For ions, scattering by Alfvén and magnetoacoustic waves requires $v \gg v_A$, and for nonrelativistic electrons, scattering by whistler waves requires $v \gg 43v_A$. These conditions are typically not satisfied for thermal particles, and one needs to postulate an injection or preacceleration mechanism to create a seed population of ions or electrons above the respective thresholds for DSA to operate on them.

One possible type of injection mechanism involves collisions populating a depleted thermal tail. The thresholds ($v \approx v_A$ for ions or $v \gg 43v_A$ for electrons) are somewhere in the tail of the thermal distribution of particles, and the acceleration removes these particles, leaving a depleted thermal tail. Collisions between thermal particles cause a net flux of particles in momentum space into the high-energy tail, tending to restore a thermal distribution. The rate at which nonthermal particles are created by this process depends on the collision frequency and on the details of the acceleration mechanism [38]. However, this process encounters two serious difficulties. First, it is too slow to account for the required injection. Second, for ions it is extremely sensitive to the charge and mass, implying that the relative isotopic abundances of accelerated ions should differ by very large factors from the relative isotopic abundances of the background plasma, and this is simply not the case in general. Other preacceleration mechanisms

involve either electric fields associated with the shock itself, or waves associated with the shock. These are discussed briefly below in connection with shock drift acceleration.

DSA at Relativistic Shocks

In the extension of DSA to highly relativistic shocks, the assumption of near isotropy of the pitch angle distribution is not valid, and the simple analytic theory does not generalize in a straightforward way. Numerical calculations show that the resulting spectrum is a power law, with $b = 4.2\text{--}4.3$ [4,41].

DSA at Multiple Shocks

In many of the applications of DSA, it is unlikely that a single shock is responsible for the acceleration [2]. Extension of the theory to multiple shocks shows that DSA is related to Fermi-type acceleration: there are energy gains associated with DSA at each shock, and energy losses due to decompression between shocks. It is straightforward to include synchrotron losses in such a model, and the competition between acceleration and synchrotron losses can possibly explain flat or slightly inverted synchrotron spectra in some sources.

Multiple Shock Model

Consider the following model for DSA at multiple shocks.

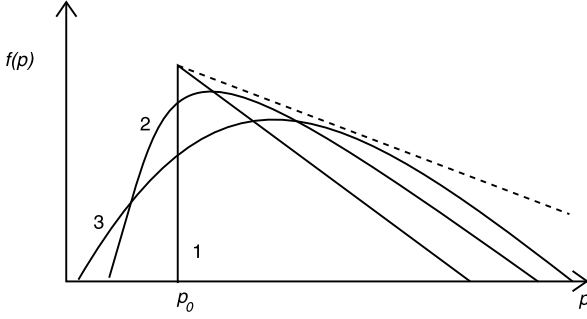
- All shocks are identical with compression ratio, $r = b/(b-3)$, and all have the same injection spectrum, $\phi_0(p)$, assumed to be monoenergetic, $\phi_0(p) \propto \delta(p - p_0)$.
- The distribution downstream of the first shock, $f_1(p)$, results from acceleration of the injection spectrum, $\phi_0(p)$, at the shock, cf. (12):

$$f_1(p) = bp^{-b} \int_0^p dq q^{b-1} \phi_0(q), \quad b = \frac{3r}{r-1}. \quad (16)$$

- Between each shock a decompression occurs. Assuming that scattering keeps the particles isotropic, the distribution function after decompression is $f'_1(p) = f_1(p/R)$, with $R = r^{-1/3}$. Hence the injection spectrum into the second shock is

$$f'_1(p) = b(p/R)^{-b} \int_0^{p/R} dq q^{b-1} \phi_0(q). \quad (17)$$

- The injection spectrum at any subsequent shock consists of the sum of $\phi_0(p)$ and the decompressed spectrum resulting from acceleration at the previous shock.



Acceleration Mechanisms, Figure 2

The distribution function $f'_i(p, p_0)$ for particles accelerated after $i = 1-3$ shock. The dashed line indicates the asymptotic solution, $f_\infty(p)$ given by (18)

The model implies that downstream of the n th shock (after decompression) the distribution is

$$f'_n(p) = \sum_{i=1}^n f'_i(p, p_0) \quad (18)$$

$$f'_i(p, p_0) = \frac{K b^i}{p_0} \left(\frac{p}{R^i p_0} \right)^{-b} \frac{(\ln p/R^i p_0)^{i-1}}{(i-1)!}.$$

After an arbitrarily large number of shocks, the spectrum approaches $f_\infty(p) \propto p^{-3}$ at $p > p_0$, as illustrated in Fig. 2. This spectrum is harder, by one power of p , than the spectrum $\propto p^{-4}$ from a single strong shock. Moreover, the spectrum $f_\infty(p) \propto p^{-3}$ is approached irrespective of the strength of the shocks, although the stronger the shocks the faster it is approached. An interpretation is that the combination of DSA and decompression leads to a Fermi-like acceleration mechanism: the asymptotic solution for Fermi-type acceleration for constant injection at $p = p_0$ is a power law with $b = 3$ for $p > p_0$. Such a distribution corresponds to a flat synchrotron spectrum: the intensity of synchrotron radiation is a power law in frequency, ν , with $I(\nu) \propto \nu^{-\alpha}$, $\alpha = (b-3)/2$.

Inclusion of Synchrotron Losses

Synchrotron losses cause the momentum of the radiating electron to decrease at the rate $dp/dt = -C_1 B^2 p^2$, with C_1 a constant. Synchrotron losses limit DSA: there is a momentum, $p = p_c$, above which the synchrotron loss rate exceeds the rate of acceleration, and acceleration to $p > p_c$ is not possible. The average acceleration rate over one cycle due to DSA is $dp/dt = \Delta(p)/t_c = C_2 p$, and with the inclusion of synchrotron losses this is replaced by $dp/dt = C_2 p - C_1 B^2 p^2 = C_2 p(1 - p/p_c)$ with

$p_c = C_2/C_1 B^2$. It is straightforward to repeat the calculation of DSA at multiple shocks including the synchrotron losses [75]. In Fig. 3 the logarithm of the distribution function is plotted as a function of $\log(p/p_0)$. The synchrotron cutoff momentum is chosen to be $p_c = 10^3 p_0$, and all the shocks have $r = 3.8$. The distribution below the synchrotron cutoff, $p < p_c$, due to the cumulative effect of injection at every shock, becomes harder than $b = 3$, such that the slope has a peak (with $b \approx 2$) just below p_c . Such a distribution corresponds to a weakly inverted spectrum with a peak just below a sharp cutoff due to synchrotron losses.

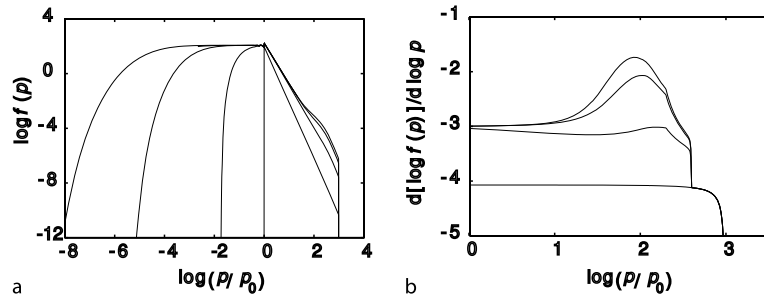
Applications of DSA

There is an extensive literature on applications of DSA, and only a few remarks on some of the applications are made here. The applications are separated into astrophysical, solar and space-physics, and the emphasis in each of these is qualitatively different. In astrophysical applications the emphasis is on the acceleration of highly relativistic particles, particularly GCRs and synchrotron-emitting electrons. In solar applications one needs to explain detailed data on the properties of the solar energetic particles (SEPs) observed in space, and on the radio, X-ray and gamma-ray signatures of them in the solar atmosphere. In space-physics, the objective is to model the detailed distributions of particles and fields measured in situ. The comments here indicate these different emphases.

Acceleration in Young Supernova Remnants

Recent observational and theoretical developments have been combined in the currently favored suggestion that GCRs are accelerated at the shocks where young supernova remnants (SNRs) interact with the interstellar medium. The observational development includes X-ray, gamma ray and TeV photon data from young supernovae, including the remnants of historical supernovae $\lesssim 10^3$ yr old, that show edge-brightening, which is interpreted in terms of synchrotron emission from relativistic electrons accelerated at an outer shock [43,101,104]. The suggestion is that GCRs are also accelerated at these shocks. The theoretical development is the suggestion [11,12] that the CRs can cause the magnetic field to be amplified by a very large factor in association with these shocks.

Since the original suggestion by Baade and Zwicky, it has been widely believed that the acceleration of GCRs is due to shocks generated by supernovae, and DSA provides an effective mechanism by which this occurs. Hillas [43] gave four arguments in favor of the location being the shocks in young SNRs:



Acceleration Mechanisms, Figure 3

The cumulative effect of DSA plus synchrotron losses after many shocks with injection at each shock; **a** the distribution, **b** the slope of the distribution for $p > p_0$: $r = 3.8$, $p_c/p_0 = 10^3$, $N = 1, 10, 30, 50$

- (a) DSA produces a power-law spectrum of about the right slope.
- (b) The model is consistent with the long-standing arguments that the only adequate energy source for GCRs is supernovae.
- (c) The composition of GCR1 is consistent with ordinary interstellar matter being injected into DSA, and it is at least plausible that such composition-independent injection occurs at such shocks.
- (d) When coupled with self-generation of magnetic field, DSA can explain the energy at which the knee between GCR1 and GCR2 occurs [44], cf. however [81]. Although the details are not complete, these arguments provide a plausible overview as to how GCR1 are accelerated.

The amplification of the magnetic field is attributed to the nonlinear development of a nonresonant plasma instability driven by the CRs [11,12,86]. A strong, self-generated magnetic field allows DSA to accelerate particles to higher energy than otherwise, due to the maximum energy being proportional to B .

Acceleration in Astrophysical Jets

Astrophysical jets are associated with accretion discs, around proto-stars and various classes of compact objects, including the supermassive black holes in active galactic nuclei (AGN). Radio jets associated with AGN can extend to enormous distances from the AGN; these jets require that the synchrotron-emitting electrons be accelerated in situ [16]. The appearance (characterized by various knots) of the jet in M87 [70] in radio and optical is remarkably similar, whereas the much shorter synchrotron loss time of the optically-emitting electrons leads one to expect the optical emission to be much more sharply localized about

acceleration sites. A possible model for the acceleration involves DSA at multiple shocks, which develop naturally in the flow, provided the acceleration is sufficiently fast to overcome the losses of the optically-emitting electrons. The required scattering of the optically-emitting electrons can plausibly be attributed to a Kolmogorov spectrum of turbulence [76].

A subclass of radio sources associated with AGN have flat or inverted spectra, corresponding to power laws $I(\nu) \propto \nu^\alpha$ with $\alpha \approx 0$ or $\alpha > 0$ over at least a decade in frequency. Synchrotron self-absorption was proposed for such spectra [51]. A specific model can account for flat spectra [68], but it requires such special conditions that it was referred to as a “cosmic conspiracy” [22]. A possible alternative explanation, for spectra with $\alpha < 1/3$, is in terms of an acceleration mechanism that produces a power-law electron spectrum with $b = 3 - 2\alpha$. DSA at a single nonrelativistic shock produces a spectrum with $b > 4$ corresponding to $\alpha < -0.5$. A relativistic shock can produce a somewhat harder spectrum but cannot account for flat spectra. DSA at multiple shocks tends towards $b = 3$, $\alpha = 0$, and provides a possible alternative explanation for flat spectra. The pile-up effect [87] occurs naturally when synchrotron losses are included in DSA at multiple shocks [75], and can account for inverted spectra with $\alpha \leq 1/3$.

Solar Energetic Particles

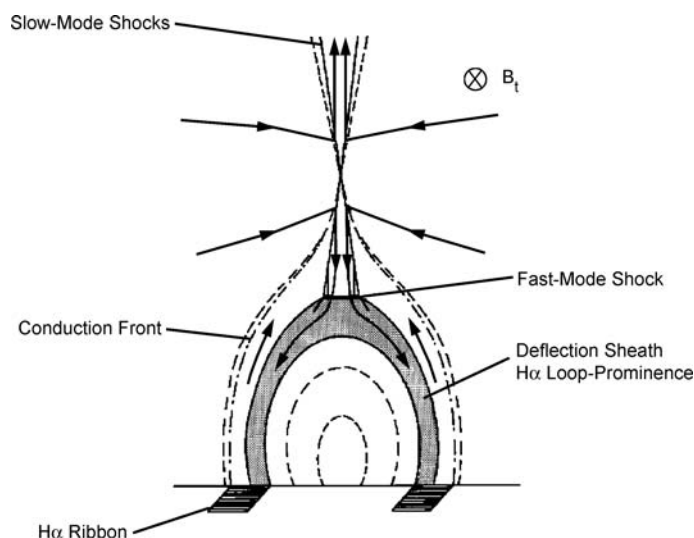
It has long been recognized that acceleration of fast particles occurs in the solar corona in connection with solar flares. There is a power-law distribution of flares in energy or in area: the energy released in a flare is approximately proportional to the area that brightens in $H\alpha$. All flares produce fast nonrelativistic (10–20 keV) electrons, observed through their radio (type III bursts) or

X-ray (bright points) emission. Large flares also produce energetic ions and relativistic electrons. In the early literature [108], it was assumed that the acceleration occurs in two phases: the first (impulsive) phase involving nonrelativistic electrons (and perhaps ions), attributed to some form of “bulk energization”, and the second phase involves a slower acceleration of solar CRs, attributed to the shocks [59], such as those that produce type II radio bursts. However, this simple picture is not consistent with other data: gamma-ray data show that the relativistic electrons and energetic ions appear immediately after the onset of the flare, without the predicted delay. Spacecraft data on solar energetic particles (SEPs) reveal further phenomena and correlations that are not consistent with the two-phase model. Various different acceleration mechanisms have been proposed for SEPs [77].

A new paradigm has emerged for the interpretation of SEPs observed in space: events are separated into flare-associated SEPs and CME-associated SEPs [62]. In a CME (coronal mass ejection) a previously magnetically bound mass of corona plasma becomes detached, accelerates away from the Sun, and drives a shock ahead of it. The detachment of a CME involves some form of magnetic reconnection in the corona. Earlier ideas on a tight correlation between flares and CMEs have not been confirmed by more recent data. Although there is some correlation, it is not one-to-one, and flares and CMEs are now regarded as distinct phenomena. The acceleration of CME-associated SEPs is plausibly due to DSA, but how the flare-associated SEPs are accelerated remains poorly understood.

There are anomalies in the ionic composition in flare-associated SEPs, the most notable of which concerns ^3He . The ratio of ^3He to ^4He in the solar photosphere is 5×10^{-4} , but the ratio in flare-associated SEPs is highly variable, and can be greatly enhanced, even exceeding unity in exceptional cases. A plausible explanation for ^3He -rich events is preferential preacceleration feeding into DSA. A specific preacceleration mechanism that selects ^3He involves the generation of ion cyclotron waves between the proton and ^4He cyclotron resonances, with these waves damping by accelerating ^3He [32]. The formulation of a self-consistent model for the acceleration of flare-associated SEPs is complicated by the paucity of direct signatures of processes in the solar corona, and the difficulty in reconciling the data on various emissions from energetic particles precipitating into the photosphere with the SEP data.

Bulk energization of electrons is not understood. A large fraction of the energy released in a flare goes into bulk energization, requiring acceleration of virtually all the electrons in the flaring region of the corona. One model that involves shock acceleration is illustrated in Fig. 4: magnetic reconnection leads to an outflow of plasma from the reconnection site, with the upflow associated with a CME and the downflow running into an underlying closed magnetic loop. In such a model, the electron heating is identified with shock acceleration where the downflow is stopped by the underlying loop. There are also various non-shock-associated mechanisms, involving E_{\parallel} , for bulk energization.



Acceleration Mechanisms, Figure 4

A cartoon [33] illustrating one suggestion for how magnetic reconnection leads to acceleration of particles in a solar flare

Resonant Scattering

The theory of resonant scattering was developed initially in two different contexts: the diffusion of GCRs through the ISM and the formation and stability of the Earth's Van Allen belts.

Streaming of GCRs

A simple model for CRs streaming along the field lines at v_{CR} is

$$\begin{aligned} f(p, \alpha) &= f_0(p) \left(1 + \frac{v_{\text{CR}}}{v} \cos \alpha \right), \\ f_0(p) &= K_{\text{CR}} \left(\frac{p}{p_0} \right)^{-b}, \end{aligned} \quad (19)$$

where the non-streaming part of the distribution function, $f_0(p)$, is measured to be a power law above some minimum p_0 , with K_{CR} a normalization constant.

The anisotropic CRs can generate the resonant waves that scatter them. Growth of Alfvén (A) and magnetoacoustic (M) waves may be described in terms of a negative absorption coefficient. On evaluating the absorption coefficient using the general expression (32) and inserting numerical values for CRs one finds

$$\begin{aligned} \gamma_{A,M}(\omega, \theta) &= -2.7 \times 10^{-7} \left(\frac{n_e}{1 \text{ cm}^{-3}} \right)^{-1/2} \left(\frac{p}{p_0} \right)^{-1.6} \\ &\quad \left(\frac{\cos \theta}{|\cos \theta|} \frac{v_{\text{CR}}}{c} - \frac{b}{3} \frac{v_A}{c} \right), \end{aligned} \quad (20)$$

where $|\cos \theta|$ is approximated by unity. It follows from (20) that the waves grow in the forward streaming direction ($\cos \theta > 0$ for $v_{\text{CR}} > 0$) for $v_{\text{CR}} > (b/3)v_A$. The scattering of the CRs by the waves generated by the streaming particles themselves causes v_{CR} to reduce. The rate of reduction may be evaluated using the quasilinear equation written down in the Appendix. One finds

$$\frac{dv_{\text{CR}}}{dt} = -v_s \left(v_{\text{CR}} - \frac{\zeta - 1}{2} \frac{|\cos \theta|}{\cos \theta} \frac{b}{3} v_A \right), \quad (21)$$

with $\zeta = (\mathcal{F}_A^+ - \mathcal{F}_A^-)/(\mathcal{F}_A^+ + \mathcal{F}_A^-)$, where \mathcal{F}_A^\pm are the fluxes of Alfvén waves in the forward (+) and backward (−) streaming directions, and with the scattering frequency given by

$$v_s = 3 \int_{-1}^{+1} d \cos \alpha \sin^2 \alpha D_{\alpha\alpha}, \quad (22)$$

with $D_{\alpha\alpha}$ determined by (34).

The growth rate (20) is fast enough to account for growth of the waves that resonate with lower-energy GCRs, and the scattering time is fast enough to reduce the streaming to close to $(b/3)v_A$. However, the growth time increases rapidly with p , with an approximate one-to-one relation between the resonant wave number k and the momentum of the resonant particles $kp = |q|B$. For $p \gg 10^{13}$ eV/c the growth time becomes ineffective, and self-generation of the resonant waves cannot account for efficient scattering.

Scattering of Higher Energy CRs

For higher-energy GCRs, effective scattering requires an external source of MHD waves. There is a turbulent spectrum in the ISM, and the observations [5] show it to be a Kolmogorov-like spectrum

$$W(k) \propto k^{-5/3}, \quad (23)$$

where $W(k)$ is the energy in the waves per unit wavenumber k . Resonant scattering by a Kolmogorov spectrum implies a scattering frequency that decreases only slowly, $v_s \propto p^{-1/3}$, with increasing p . This scattering must dominate scattering by self-generated waves at higher p . It is likely that the distribution of turbulence is far from homogeneous, and that the most effective scattering occurs in localized regions of enhanced turbulence. How effective scattering of higher energy particles occurs is a poorly understood aspect of acceleration mechanisms.

Nonresonant Versus Resonant Growth

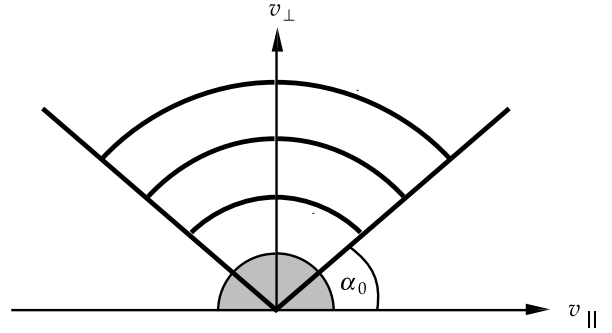
Plasma instabilities can occur in two forms: resonant and nonresonant. A nonresonant instability involves an intrinsically growing wave, described by a complex solution of a real dispersion equation. A recent suggestion that has stimulated considerable interests involves an instability [11,12] that is a nonresonant counterpart of (20) [73]. The potential importance of this instability is twofold. First, it can allow the lower-energy particles to generate a wide spectrum of waves including those needed to scatter higher-energy particles. Second, its nonlinear development can lead to amplification of the magnetic field, which ultimately causes the instability to saturate [11,12,86]. A similar idea has been proposed in connection with generation of magnetic fields through a Weibel instability associated with the relativistic shocks that generate gamma-ray bursts [107]. Such nonresonant growth is an important ingredient in the currently favored acceleration mechanism for GCR1 [43].

Formation of the Van Allen Belts

A simple model for the formation of the radiation belts is based on adiabatic invariants, which are associated with quasi-periodic motions. The first adiabatic invariant is associated with the gyration about a field line, and it implies that $M \propto p_{\perp}^2/B$ is an invariant. After averaging over the gyrations, a fast particle bounces back and forth along a field line between two reflection points. The adiabatic invariant associated with this bounce motion, denoted J , is the integral of p_{\parallel} along the field lines between the reflection points. After averaging over bounce motion, the remaining motion is a periodic drift around the Earth. There is a third adiabatic invariant, Φ , which corresponds to the integral for the vector potential around the Earth. For a dipole-like magnetic field it is convenient to label a field line by the radial distance, L , to its midpoint. Then one has $M \propto p_{\perp}^2 L^3$, $J \propto p_{\parallel} L$, $\Phi \propto 1/L$. Violation of the third adiabatic invariant, due to perturbations in the Earth magnetic field with a period equal to that of the drift motion, causes the particles to diffuse in L at constant M and J . The steady state density that would result from such diffusion is proportional to $p_{\perp}^2 p_{\parallel} \propto 1/L^4$. Particles that enter the magnetosphere from the solar wind tend to diffuse inwards to build up this density. As the particles diffuse to smaller L , their momentum increases according to $p \propto L^{-3/4}$. The inward diffusion ceases at the point where pitch-angle scattering (which violates conservation of M) becomes effective and causes the particles to be scattered into the loss cone and to precipitate into the Earth's neutral atmosphere. Due to their loss-cone anisotropy, illustrated schematically in Fig. 5, the anisotropic particles can generate the resonant waves needed to scatter them, with the growth rate increasing rapidly with decreasing L . In this way, resonant scattering provides the sink (at smaller L) that allows the Van Allen belts to remain in a steady state when the magnetosphere is quiescent. The absorption coefficient for the waves tends to be maintained near zero, referred to as marginal stability, such that sporadic localized bursts of wave growth lead to sporadic localized bursts of particle precipitation. Such bursts of wave growth can be triggered by artificially generated VLF signals [42].

Acceleration by Parallel Electric Fields

Acceleration by an electric field parallel to the magnetic field, E_{\parallel} , is the simplest conceivable acceleration mechanism, but also the least understood. The argument for acceleration by E_{\parallel} is compelling in connection with aurorae, is very strong in connection with pulsars, and is strong for bulk energization in solar flares.



Acceleration Mechanisms, Figure 5

An idealized loss-cone distribution is illustrated; α_0 is the loss cone angle, the shaded region is filled by isotropic thermal particles, and the circular arcs denote contours of constant f

Available Potential

It is sometimes helpful to note that all relevant acceleration mechanisms can be attributed to an inductive electric field, and to identify the specific inductive field. In particular, the maximum available potential, or EMF (electromotive force), determines the maximum energy to which particles can be accelerated. There are two general useful semi-quantitative estimates of the available potential: one is the EMF associated with changing magnetic flux through a circuit; the other involves the power radiated by a rotating magnet in vacuo.

EMF Due to a Relative Motions The EMF in a simple circuit, due to changing magnetic flux through the circuit, is $\Phi = -d(BA)/dt$, where A is the area of the circuit. Although Φ can be due to a changing magnetic field, $\Phi = AdB/dt$, in most cases of interest it is due to relative motions, $\Phi = BdA/dt = BLv$, where L and v need to be identified in a specific model. A $\Phi = BLv$ appears along a magnetic flux tube line that connects two MHD regions in relative motion to each other, with v the relative velocity perpendicular to the flux tube, with a magnetic field B in one of the regions, with L the perpendicular dimension of the flux tube orthogonal to the relative velocity in this region. This simple argument needs to be complemented with an identification of how the current closes, and how this potential is distributed around the closed circuit. This model provides a plausible estimate for the energy of auroral electrons and for electrons accelerated in Jupiter's magnetosphere. In the case of auroral particles, the velocity, v , is between the Earth and the solar wind, and the acceleration occurs along the auroral field lines that connect these two regions. Parameters B and L across the magnetotail and v for the solar wind, give Φ of several kV,

consistent with the observed auroral electrons of a several keV. In an early model for the Io–Jupiter interaction, the moon Io was assumed to drag the Jovian magnetic field lines threading it through the corotating magnetosphere, leading to the estimate $\Phi = B_{10}L_{10}v_{10}$, with B_{10} is the Jovian magnetic field at the orbit of Io, L_{10} is the diameter of Io and v_{10} is the relative velocity of Io to the corotating magnetic field lines [37]. This gives Φ of a few MV, consistent with a copious supply of few-MeV electrons associated with Io's motion through the Jovian magnetosphere.

Another application of this estimate of Φ provides a limit on the maximum energy to which particles can be acceleration due to DSA. The relative flow on opposite sides of the shock implies an electric field in the plane of the shock front: one may identify this electric field as Bv , with v the relative flow speed perpendicular to \mathbf{B} . Then $\Phi = BvL$ is the potential available across the lateral extent, L , of the shock. The maximum energy to which a particle can be accelerated through DSA is limited to $< eBvL$ per unit charge.

A point of historical interest is that the earliest recognized suggestion for acceleration of CRs involved an inductive field due to dB/dt : in 1933 Swann [94] proposed that the acceleration occurs in starspots due to the magnetic field increasing from zero to $B = 0.2$ T in a time $\tau = 10^6$ s in a region of size $L = 3 \times 10^8$ m, giving $\Phi = BL^2/\tau = 2 \times 10^{10}$ V. Although this model is not considered realistic, it is of interest that a similar value for Φ results from $\Phi = BLv$ with the same B , L and with L/τ replaced by $v = 3 \text{ km s}^{-1}$; these numbers are close to what one would estimate for the potential difference between field lines on opposite sides of a rotating sunspot. Such large potential drops are available in the solar atmosphere, but the implications of this are poorly understood.

EMF Due to a Rotating Compact Object In a simple model for the electrodynamics of a rotating magnetized compact object, such as a neutron star or a black hole, the slowing down is estimated assuming that the rotational energy loss is due to magnetic dipole radiation. The power radiated may be written as $I\Phi$, where the effective displacement current, I , and the available potential, Φ , are related by $\Phi = Z_0 I$, with $Z_0 = \mu_0 c = 377 \Omega$ the impedance of the vacuum. Equating Φ^2/Z_0 to the rate of rotational energy loss provides as estimate of Φ . Although the slowing down for pulsars is not actually due to magnetic dipole radiation, estimates based on this model underpin the basic interpretation of pulsars, including estimates of the age, the surface magnetic field and the available potential. The Φ may be interpreted as between the stellar surface and the pulsar wind region outside the light cylinder, lead-

ing to an E_{\parallel} along the polar-cap field lines that connect these regions.

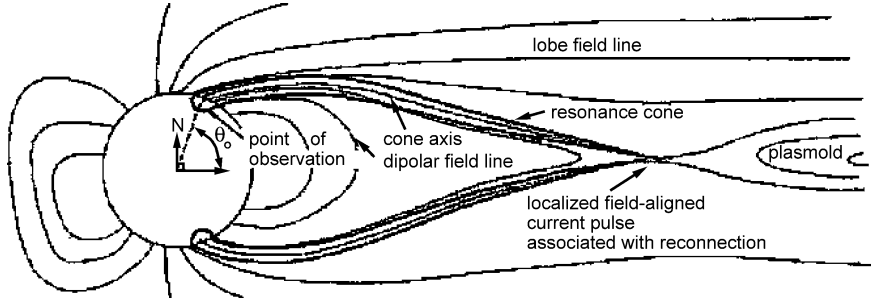
Acceleration of Auroral Electrons

Magnetic reconnection in the Earth's magnetotail, in a so-called magnetospheric substorm, results in energetic electrons precipitating into the upper atmosphere, producing the visible aurora. A build up of the magnetic stresses prior to a substorm results from reconnection, at the sunward edge of the magnetosphere, between terrestrial field lines and with interplanetary field lines, with the solar wind transferring magnetic flux from the front of the magnetosphere to the magnetotail. There is a build up of the cross-tail current associated with the build up of the magnetic flux. A generic model for a substorm [103] involves relaxation of these stresses through reconnection, which creates closed field lines, and allows transfer of magnetic flux to the front of the magnetosphere as these reconnected field rotate. The reconnection also causes the cross-tail current to be partly redirected around a circuit along the reconnecting field lines, so that the current closes by flowing along field lines to the ionosphere, across the field lines due to the Pedersen conductivity there, and returning to the magnetotail along a nearby field lines. The magnetic energy released is goes partly into an Alfvénic flux towards the Earth, and this energy flux is partially converted into energetic electrons in an acceleration region. The acceleration region is far removed from the reconnection site, and is at a height $> 10^3$ km, well above the location of the visible aurora, as indicated schematically Fig. 6. The acceleration is due to an E_{\parallel} in the acceleration region, but the mechanism that sets up the E_{\parallel} is still not clear. Three ideas are discussed briefly here.

A parallel potential difference develops between the ionosphere, Φ_{ion} , and the magnetosphere, Φ_{mag} , along a current-carrying field line. The current implies a flow of electrons relative to ions. There is a limited supply of magnetospheric electrons to carry the current, due to most electrons mirroring before they reach the ionosphere. The potential difference develops to allow the current to flow. Let R_m be the mirror ratio, along a current-carrying field line. For $1 \ll e\Phi_{\parallel}/T_e \ll R_m$, the so-called Knight relation [54,63,102] gives

$$J_{\parallel} \approx K(\Phi_{\text{ion}} - \Phi_{\text{mag}}), \quad K = \frac{e^2 n_e}{(2\pi)^{1/2} m_e V_e}, \quad (24)$$

where $T_e = m_e V_e^2$ is the electron temperature. The Knight relation (24) is useful for a number of purposes, but it is not particularly useful in considering how the potential localizes to create the E_{\parallel} in the acceleration region.



Acceleration Mechanisms, Figure 6

A cartoon [13] showing the regions in the Earth's magnetosphere associated with the acceleration of auroral electrons. The generation region is where magnetic reconnection occurs, launching Alfvén waves, which accelerate electrons in an acceleration region where the EMF from the generator is localized along field lines

In situ observations of E_{\parallel} in the acceleration region have been interpreted in a variety of ways. Localized structures sweeping across a spacecraft were initially called electrostatic shocks [78] and also referred to as double layers (DLs) [19,28]. The theory of DLs is well established for laboratory plasmas, and models for DLs have been adapted to space and astrophysical plasma applications [18,85]. DLs are classified as weak or strong, depending on whether the potential difference is of order or much greater than T_e/e , respectively. Some models for the formation and structure of DLs involve two other ideas that have been discussed widely in the literature: anomalous resistivity and electron phase space holes [80]. However, the details of how localized DLs accelerate particles and produce the observed fluxes of energetic electrons is still not clear.

Another way in which an E_{\parallel} can arise is through the Alfvén waves that transport the energy released in the magnetotail [92,103,109]. The condition $E_{\parallel} = 0$ is strictly valid for an Alfvén wave only in ideal MHD, and $E_{\parallel} \neq 0$ can arise due to electron inertia, due to finite thermal gyroradii and due to (anomalous) resistivity. Low-frequency Alfvén waves with large k_{\perp} become either inertial Alfvén waves (IAW) or kinetic Alfvén waves, depending on whether inertia or thermal effects dominate. The dispersion relation and the ratio of the parallel to the perpendicular electric fields in an IAW depend on the skin depth, $\lambda_e = c/\omega_p$:

$$\frac{\omega}{|k_{\parallel}|} = \frac{v_A}{(1 + k_{\perp}^2 \lambda_e^2)^{1/2}}, \quad \frac{E_{\parallel}}{E_{\perp}} = \frac{k_{\parallel} k_{\perp} \lambda_e^2}{1 + k_{\perp}^2 \lambda_e^2}. \quad (25)$$

The E_{\parallel} associated with IAWs is one suggested mechanism for auroral acceleration: the skin depth is several kilometers, of the order, 10 km, of observed changes in J_{\parallel} across field lines [103,109]. Acceleration by IAWs is a relatively old idea [36], favored for electrons accelerated by Alfvén

waves generated by Io's motion through the Jovian magnetosphere [35]. An important detail is that this E_{\parallel} , being a wave field, reverses sign, accelerating electrons in opposite directions, every half wave period.

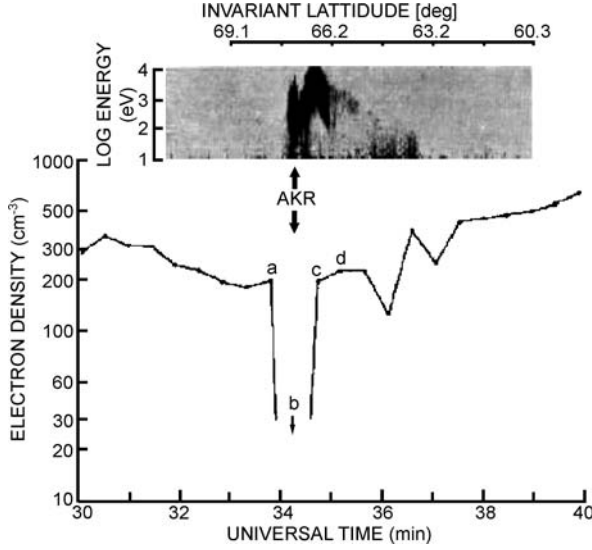
Another relevant idea is that of an Alfvénic resonator. Inhomogeneities along field lines cause v_A to change, and this leads to reflection of Alfvén waves at the top of the ionosphere. Suppose there is also reflection at some higher region. This combination leads to localized trapping of wave energy in the Alfvén resonator between the reflection points [29,64]. A self-consistent acceleration model requires that the upper reflection site be where the E_{\parallel} develops and the electrons are accelerated, perhaps resulting in a localized current instability and associated anomalous resistivity that cause the Alfvén waves to be reflected.

An extreme example of auroral acceleration occurs in inverted-V electrons. The upward current can essentially depopulate a localized auroral flux tube of all thermal electrons, leading to a plasma cavity in which the inverted-V electrons are observed, Fig. 7. The “inverted-V” refers to the dynamic spectrum observed by a spacecraft as it crosses the cavity: lowest energies near the edges and maximum energy near the center. Inverted-V electrons correlate with auroral kilometric radiation [39], which is due to electron cyclotron maser emission.

One can conclude that despite significant progress, the details of the acceleration of auroral electrons still need further investigation [20].

Acceleration in Pulsars Gaps

Pulsars are rapidly rotating neutron stars with superstrong magnetic fields. There are three broad classes: ordinary, recycled (or millisecond), and magnetars; only ordinary pulsars are considered here. These have periods $P \approx 0.1$ –1 s, surface magnetic fields $B \approx 10^7$ – 10^9 T and are slow-



Acceleration Mechanisms, Figure 7

A density depletion in an auroral region of the Earth; inverted-V electrons carry a large upward current in such regions [14]

ing down at a rate $\dot{P} \sim 10^{-15}$. The age, $P/2\dot{P}$, and the surface magnetic field, $B \propto (P\dot{P})^{1/2}$, are estimated assuming the rotating magnetic dipole model in which the rotational energy loss is due to magnetic dipole radiation. Equating the energy loss due to the observed slowing down to Φ^2/Z_0 gives an estimate for the potential available along field lines in the polar-cap region. For the Crab pulsar the rotational energy loss is approximately 10^{31} W, implying $\Phi \approx 6 \times 10^{16}$ V; the available potentials in most other pulsars are somewhat smaller than this. In the most widely favored model, the electric field is assumed to cause a charge-limited flow from the stellar surface, providing the “Goldreich–Julian” charge density required by the corotation electric field. However, the Goldreich–Julian charge density cannot be set up globally in this way, and requires additional source of charge within the magnetosphere. The available potential is assumed to develop across localized regions called “gaps” where acceleration of “primary” particles, due to $E_{\parallel} \neq 0$, results in the emission of gamma rays which trigger a pair cascade that produces the necessary secondary (electron–positron) charges. This additional source of charge is assumed to screen E_{\parallel} outside the gaps. However, there is no fully consistent model and no consensus on the details of the structure and location of the gaps. The present author believes that existing models, based on a stationary distribution of plasma in a corotating frame, are violently unstable to temporal perturbations, resulting in large-amplitude oscillations in E_{\parallel} [61].

Although the details of acceleration by E_{\parallel} are quite different in the pulsar case from the auroral case, the underlying difficulties in understanding such acceleration are similar. A static model for acceleration by E_{\parallel} encounters seemingly insurmountable difficulties, and an oscillating model involves unfamiliar physics.

Acceleration of Nonrelativistic Solar Electrons

A large fraction of the energy released in a solar flare goes into very hot electrons: all the electrons in a relatively large volume are accelerated in bulk to 10–20 keV. Several models for bulk heating involving acceleration by E_{\parallel} have been proposed, but all have unsatisfactory features. One idea is to invoke a runaway process [45]. Consider the equation of motion of an electron in an electric field subject to Coulomb collisions with a collision frequency $\nu_e(\nu) = \nu_e(V_e/\nu)^3$, where V_e is the thermal speed of electrons:

$$\frac{d\mathbf{v}}{dt} = -\frac{e\mathbf{E}}{m} - \nu_e \mathbf{v} \left(\frac{V_e}{v} \right)^3. \quad (26)$$

It follows that electrons with speed

$$\frac{v}{V_e} > \left(\frac{E_D}{E} \right)^{1/2}, \quad E_D = \frac{mV_e \nu_e}{e}, \quad (27)$$

are freely accelerated; E_D is called the Dreicer field. This so-called electron runaway sets up a charge separation that should quickly screen E_{\parallel} and suppress the acceleration. It also causes J_{\parallel} to change, and this is opposed by inductive effects. Runaway on its own is not a plausible acceleration mechanism.

There are inconsistencies in estimates of the available potential, Φ . Assuming that bulk energization is due to acceleration by an E_{\parallel} implies $\Phi \approx 10$ –20 kV, that is of order 10^4 V. Two arguments suggest a much larger potential. On the one hand, there is observational evidence that large flares tend to occur in flux loops carrying large currents, $I \gtrsim 10^{12}$ A. If one identifies I with the observed current, then to produce the power $I\Phi$ of order 10^{22} W released in a large flare requires Φ of order 10^{10} V. On the other hand, as indicated above, the potential available due to photospheric motions is estimated above to be 10^9 – 10^{10} V. This potential cannot be ignored when two magnetic loops come in contact and reconnect: the potential difference between the reconnecting field lines is expected to be of order 10^9 – 10^{10} V.

There are several problems that have yet to be clearly resolved. One problem is how the required $E_{\parallel} \neq 0$ develops. The suggestions as to how this problem is resolved in the

auroral acceleration region need to be explored for the solar application. (An exception is inertial Alfvén waves: the skin depth being far too small to be relevant in the solar corona.) The suggestion that double layers might play an important role in solar flares has a long history [85], as do the suggestions involving anomalous resistivity [26] and an Alfvénic resonator [47,48]. However, none of the existing models is satisfactory. Another problem is how to reconcile the estimates $\Phi \approx 10^4$ V and 10^9 – 10^{10} V. It appears that the available 10^9 – 10^{10} V must localize into $> 10^5$ regions with $E_{\parallel} \neq 0$ and $\Phi \approx 10^4$ V, where the acceleration occurs, such that the accelerated electrons that escape from one region do not pass through any other. A third problem is that, like the auroral and pulsar cases, any model for acceleration by a static E_{\parallel} encounters serious difficulties. Although a large-scale oscillating E_{\parallel} might plausibly overcome these difficulties, there is no model for such oscillations.

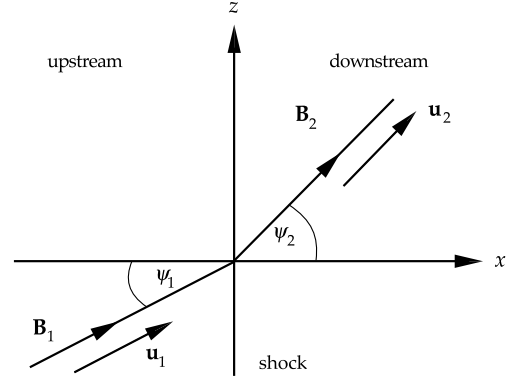
Other Acceleration Mechanisms

The three most important acceleration mechanisms are DSA, stochastic acceleration and acceleration by parallel electric fields. Various other mechanisms are relevant in specific applications. Examples mentioned above are the acceleration of energetic particles in the Earth's Van Allen belts and various preacceleration mechanisms for DSA. Some comments are made here on three other mechanisms: shock drift acceleration, gyroresonant acceleration and acceleration in current sheets.

Shock Drift Acceleration (SDA)

When a particle encounters a shock it is either transmitted across the shock front or reflected from it, and in either case it tends to gain energy [96]. Shock drift acceleration (SDA) is attributed to the grad- B drift when the particle encounters the abrupt change in B : the scalar product of drift velocity and acceleration by the convective electric field, $-q\mathbf{u} \times \mathbf{B}$, is positive, implying energy gain. The energy gained by a particle depends on how far it drifts along the front. A simple model for DSA is based on p_{\perp}^2/B being conserved when the particle encounters the shock. This conservation law is somewhat surprising (it cannot be justified on the usual basis that p_{\perp}^2/B is an adiabatic invariant), and it applies only approximately and only after averaging over gyrophase [25,84].

In treating SDA it is convenient to make a Lorentz transformation to the de Hoffmann–Teller frame. The frame most widely used in discussing a shock is the shock-normal frame, in which the shock front is stationary and the upstream flow velocity is normal to it. In this frame



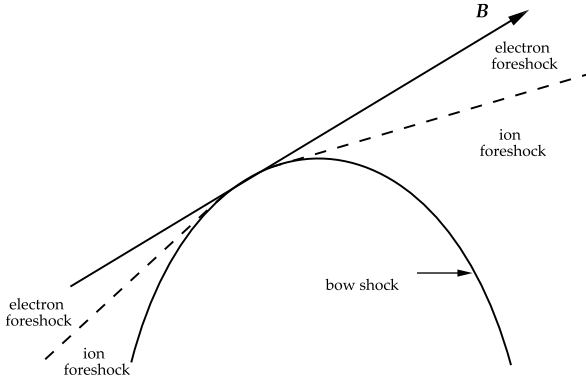
Acceleration Mechanisms, Figure 8

In the de Hoffmann–Teller frame the fluid velocities are parallel to the magnetic fields on either side of the shock

there is a convective electric field, $-\mathbf{u} \times \mathbf{B}$, on both sides of the shock, and the point of intersection of \mathbf{B} with the shock moves in the plane of the shock. In the de Hoffmann–Teller frame this point of intersection is stationary, \mathbf{u} and \mathbf{B} are parallel to each other on both sides of the shock, and there is no electric field, Fig. 8. The de Hoffmann–Teller frame exists only for “subluminal” shocks for which the velocity of intersection is less than c ; in the “superluminal” case, where the velocity of intersection is greater than c , there exists a frame in which \mathbf{B} is perpendicular to the shock normal. In the de Hoffmann–Teller frame the energy of a particle is unchanged when it is transmitted across the shock, or is reflected from the shock. A simple theory for SDA involves the energy being conserved in the de Hoffmann–Teller frame on transmission or reflection, with the implications of this being found by Lorentz transforming to the shock-normal frame.

Let quantities in the de Hoffmann–Teller and shock-normal frames be denoted with and without primes, respectively. Let ψ_1 be the angle between \mathbf{u}_1 and \mathbf{B}_1 . Provided that the relative velocity between the two frames is nonrelativistic, the angle ψ_1 is the same in the two frames, and the relative velocity is $u_0 = u_1 \tan \psi_1$. In the nonrelativistic case, the pitch angles in the two frames are related by $v' \cos \alpha' = v \cos \alpha + u_0$, $v' \sin \alpha' = v \sin \alpha$. Reflection is possible only for $\sin^2 \alpha'_1 \geq B_1/B_2$, and the maximum change in energy occurs for reflected particles at the threshold, $\alpha'_1 = \alpha_c$, $\sin^2 \alpha_c = B_1/B_2$. The maximum ratio of the energy after reflection (ref) to before (inc) is for a particle with $v_1 \cos \alpha_1 = -u_0(1 - \cos \alpha_c)$, that is propagating away from the shock, and is overtaken by the shock. The maximum ratio is

$$\left(\frac{E_{\text{ref}}}{E_{\text{inc}}} \right)_{\text{max}} = \frac{1 + (1 - B_1/B_2)^{1/2}}{1 - (1 - B_1/B_2)^{1/2}}, \quad (28)$$



Acceleration Mechanisms, Figure 9

The foreshock may be separated into an electron foreshock, nearer to the field line that is tangent to the bow shock, and an ion foreshock, nearer to the bow shock; the *dashed line* indicates the ill-defined boundary between these two regions, which actually merge continuously into each other

which is also the case in a relativistic treatment [53]. For a strong shock, $B_2 \rightarrow 4B_1$, the ratio (28) is $(2 + \sqrt{3})/2 - \sqrt{3} = 13.93$. However, the energy ratio decreases rapidly away from the point in the momentum space of the incident particles where this maximum occurs [9].

SDA accounts well for relevant observations of particles associated with shocks in the heliosphere, in particular, the Earth's bow shock and other planetary bow shocks. SDA produces a foreshock region ahead of any (convex) curved shock. The foreshock is the region ahead of the shock and behind the field line that is tangent to the shock, as illustrated in Fig. 9. Fast particles reflected from the shock propagate along the field lines as they are swept towards the shock by the upstream flow. Electrons have higher speeds than ions, so that they propagate further before encountering the shock, producing an electron foreshock region that is much larger than the ion foreshock region. SDA should similarly populate a foreshock region ahead of any curved shock.

Type II solar radio bursts are directly associated with shocks in the solar corona, and extending into the solar wind [79]. Acceleration by SDA is a favored mechanism for the acceleration of electrons that produce type II radio bursts [46,55,60]. SDA is also a potential candidate for preacceleration for DSA. Although both suggestions seem plausible, additional ideas are required to account for the details of the observations.

A modification of DSA, called shock surfing acceleration [100], involves the particles being trapped in the shock front for much longer than the simple theory of SDA implies. In other modifications, the assumption that the

shock is a simple discontinuity is relaxed. Various additional possible acceleration mechanisms arise: due to a potential difference along field lines [49]; various instabilities that can lead to large-amplitude waves that can trap and preaccelerate particles ("surfatron" acceleration [66]); and nonstationarity that involves the shock continually breaking up and reforming [89].

Resonant Acceleration

MHD turbulence consists of a mixture of the Alfvén and fast modes. Fermi-type stochastic acceleration involves only the fast mode, and may be interpreted in terms of resonant damping at harmonic number $s = 0$. Resonances at $s \neq 0$ are possible and lead to dissipation of turbulence in both modes, with the energy going into the particles. For highly relativistic particles, resonances at high harmonics lead to a diffusion in momentum space, somewhat analogous to Fermi-type acceleration, but with a diffusion coefficient that depends on a higher power of p [58]. Although this acceleration must occur, for say GCRs due to turbulence in the interstellar medium, there is no strong case for it playing an important role in specific applications.

Acceleration During Magnetic Reconnection

Magnetic energy release through magnetic reconnection is known to be associated with powerful events, notably magnetospheric substorms and solar flares, that can lead to copious acceleration of fast particles. There have been various suggestions that reconnection can lead directly to acceleration of fast particles. For example, for magnetic reconnection in a current sheet, around the surface where \mathbf{B} changes sign, a test particle that enters the sheet typically emerges with a higher energy [91]. Although such energy gains occur, they do not appear to constitute an important acceleration mechanism. Both theory and observation suggest that the important acceleration associated with reconnection is indirect and spatially separated from the reconnection, and is not a direct consequence of the reconnection. The magnetic energy released in reconnection goes primarily into an Alfvénic flux and into mass motions accelerated by the magnetic tension in the reconnected field lines.

A different application of heating in current sheets has been proposed in connection with pulsar winds [21,65]. Beyond the light cylinder there is a wind containing wound up magnetic field lines in a nearly azimuthal direction, with the sign of \mathbf{B} reversing periodically with radial distance. The wind accelerates particles at a termination shock, where it encounters the surrounding synchrotron nebula. An outstanding problem is that the energy flux is

Poynting dominated near the light cylinder, but must be kinetic-energy dominated before it reaches the termination shock. The suggestion is that magnetic energy dissipation in the current sheets separating the regions of opposite \mathbf{B} provide the necessary conversion of magnetic energy into particle energy [21,65]. An alternative suggestion is that the number of current carriers becomes inadequate leading to the development of an accelerating field [71].

Future Directions

Three generic acceleration mechanisms are widely recognized, and there are also numerous other specific mechanisms that are important in specific applications. The pioneering works of Fermi [30,31] anticipated two of the three generic mechanisms: stochastic acceleration and diffusive shock acceleration (DSA), respectively. The third generic mechanism, acceleration by a parallel electric field, E_{\parallel} , was suggested even earlier [94], but remains the least adequately understood of the three.

The most important acceleration mechanism in astrophysics is DSA. It is the accepted mechanism for the acceleration of GCRs and for relativistic electrons in most synchrotron sources. The favored location for the acceleration of GCRs is at the shocks in young supernova remnants [43]. Despite the basic theory being well established there are essential details where our current understanding is incomplete and where further progress is to be expected. The well-established theory is a test-particle model for DSA at a single nonrelativistic shock. Several modifications to the theory are known to be important: the generalizations to relativistic shocks, and to multiple shocks, and the dynamical effects of the accelerated particles on the shock structure. There are several aspects of the required efficient scattering in DSA that are inadequately understood. One is the injection problem: resonant scattering is essential and requires that nonrelativistic ions and electrons have speeds $v > v_A$ and $v > 43v_A$, respectively. Preacceleration to above these thresholds is required before DSA can operate. For ions, the elemental abundances of GCRs suggests an injection mechanism that is insensitive to ionic species, whereas for flare-associated solar energetic particles there are extreme elemental anomalies (notably ^3He). For electrons the injection problem is more severe, and less understood. Second is the long-standing question of the ratio of relativistic electrons to ions [34]. In situ data on shocks in the heliosphere provide little evidence for DSA producing relativistic electrons. It is speculated that the shocks in synchrotron sources have much higher Mach numbers than those in the heliosphere, and that such shocks are more efficient in accelerating elec-

trons. A third aspect relates to resonant scattering of the highest energy particles. Unlike lower-energy particles, the resonant waves needed to scatter higher-energy particles cannot be generated by the particles themselves, and other sources of the required waves are speculative. A suggestion that at least alleviates this difficulty is that the lower-energy particles cause waves to grow through a nonresonant instability, which may lead to amplification of the magnetic field by a very large factor [11]: an increase in B increases the maximum energy to which DSA can operate. Fourth, spatial diffusion across field lines is postulated to occur in DSA at perpendicular shocks, but is incompletely understood [3].

Despite its apparent simplicity, acceleration by E_{\parallel} is the least understood of the three generic mechanisms. An important qualitative point is that $E_{\parallel} \neq 0$ is inconsistent with MHD, and acceleration by E_{\parallel} necessarily involves concepts that are relatively unfamiliar in the context of MHD. The best understood application is to acceleration of auroral electrons associated with magnetospheric substorms, but even in this case, how the E_{\parallel} develops remains uncertain. Existing models for acceleration by E_{\parallel} in ‘gaps’ in pulsars and in bulk energization of electrons in solar flares have serious flaws. Some of the present difficulties are alleviated by assuming that the relevant E_{\parallel} is oscillating: specific models include inertial Alfvén waves and an Alfvén resonator in the magnetosphere, and large-amplitude oscillations in pulsar magnetospheres. Further progress in understanding acceleration by E_{\parallel} is likely to come from more detailed in situ data on acceleration of auroral electrons.

Compared with the rapid evolution in ideas in other subfields of astrophysics, the ideas involved in understanding particle acceleration have developed only gradually over many decades. Further progress is certain to occur, but is likely to involve gradual acceptance and modification of the existing ideas outlined above, rather than radical new insights.

Appendix A: Quasilinear Equations

The quasilinear equations are written down here using a semiclassical formalism. The waves are regarded as a collection of wave quanta, with energy $\hbar\omega$ and momentum $\hbar\mathbf{k}$. Waves in an arbitrary mode, labeled M , have dispersion relation $\omega = \omega_M(\mathbf{k})$, polarization vector $\mathbf{e}_M(\mathbf{k})$ and ratio of electric to total energy, $R_M(\mathbf{k})$. Their distribution is described by their occupation number $N_M(\mathbf{k})$. A resonant interaction occurs when the gyroresonance condition is satisfied:

$$\omega - s\Omega - k_{\parallel}v_{\parallel} = 0, \quad (\text{A1})$$

where s is an integer, $\Omega = |q|B/m\gamma$ is the relativistic gyrofrequency, and $k_{\parallel}, v_{\parallel}$ are the components of \mathbf{k}, \mathbf{v} parallel to \mathbf{B} . Resonances at $s > 0$ are said to be via the normal Doppler effect, and those at $s < 0$ are said to be via the anomalous Doppler effect. Resonances at $s \leq 0$ are possible only for waves with refractive index greater than unity. The effect of a wave-particle interaction is described by the probability of spontaneous emission, $w_M(\mathbf{p}, \mathbf{k}, s)$, which is the probability per unit time that the particle emit a wave quantum in the wave mode M in the elemental range $d^3\mathbf{k}/(2\pi)^3$. For a particle of charge q and mass m , the probability of spontaneous emission is given by

$$w_M(\mathbf{p}, \mathbf{k}, s) = \frac{2\pi q^2 R_M(\mathbf{k})}{\varepsilon_0 \hbar \omega_M(\mathbf{k})} |\mathbf{e}_M^*(\mathbf{k}) \cdot \mathbf{V}(\mathbf{k}, \mathbf{p}; s)|^2 \cdot \delta(\omega_M(\mathbf{k}) - s\Omega - k_{\parallel}v_{\parallel}),$$

$$\mathbf{V}(\mathbf{k}, \mathbf{p}; s) = (v_{\perp} \frac{s}{k_{\perp}R} J_s(k_{\perp}R), -i\eta v_{\perp} J'_s(k_{\perp}R), v_{\parallel} J_s(k_{\perp}R)),$$

$$\Omega = \frac{\Omega_0}{\gamma}, \quad \Omega_0 = \frac{|q|B}{m}, \quad R = \frac{v_{\perp}}{\Omega} = \frac{p_{\perp}}{|q|B}.$$
(A2)

An advantage of the semiclassical formalism is that the Einstein coefficients imply that the probability of stimulated emission and true absorption are given by multiplying this probability by $N_M(\mathbf{k})$. This allows the effect on the occupation numbers of the waves and the particles to be written down using a simple bookkeeping argument. For the waves one finds

$$\frac{dN_M(\mathbf{k})}{dt} = \left(\frac{dN_M(\mathbf{k})}{dt} \right)_{\text{spont}} - \gamma_M(\mathbf{k})N_M(\mathbf{k}), \quad (\text{A3})$$

where the two terms on the right hand side describe spontaneous emission and absorption, respectively, with the absorption coefficient given by

$$\gamma_M(\mathbf{k}) = - \sum_{s=-\infty}^{\infty} \int d^3\mathbf{p} w_M(\mathbf{p}, \mathbf{k}, s) \hat{D}_s f(\mathbf{p}),$$

$$\hat{D}_s = \hbar \left(\frac{s\Omega}{v_{\perp}} \frac{\partial}{\partial p_{\perp}} + k_{\parallel} \frac{\partial}{\partial p_{\parallel}} \right)$$

$$= \frac{\hbar\omega}{v} \left(\frac{\partial}{\partial p} + \frac{\cos\alpha - n_M \beta \cos\theta}{p \sin\alpha} \frac{\partial}{\partial \alpha} \right). \quad (\text{A4})$$

The two forms correspond to, respectively, cylindrical and polar coordinates in momentum space.

The evolution of the distribution of particles due to the resonant interaction includes a term, neglected here, that describes the effect of spontaneous emission, and a quasi-linear diffusion equation that describes the effect of the in-

duced processes. In cylindrical and spherical polar coordinates, this equation is

$$\frac{df(\mathbf{p})}{dt} = \frac{1}{p_{\perp}} \frac{\partial}{\partial p_{\perp}} \cdot \left\{ p_{\perp} \left[D_{\perp\perp}(\mathbf{p}) \frac{\partial}{\partial p_{\perp}} + D_{\perp\parallel}(\mathbf{p}) \frac{\partial}{\partial p_{\parallel}} \right] f(\mathbf{p}) \right\}$$

$$+ \frac{\partial}{\partial p_{\parallel}} \left\{ \left[D_{\parallel\perp}(\mathbf{p}) \frac{\partial}{\partial p_{\perp}} + D_{\parallel\parallel}(\mathbf{p}) \frac{\partial}{\partial p_{\parallel}} \right] f(\mathbf{p}) \right\}$$

$$= \frac{1}{\sin\alpha} \frac{\partial}{\partial \alpha} \cdot \left\{ \sin\alpha \left[D_{\alpha\alpha}(\mathbf{p}) \frac{\partial}{\partial \alpha} + D_{\alpha p}(\mathbf{p}) \frac{\partial}{\partial p} \right] f(\mathbf{p}) \right\}$$

$$+ \frac{1}{p^2} \frac{\partial}{\partial p} \left\{ p^2 \left[D_{p\alpha}(\mathbf{p}) \frac{\partial}{\partial \alpha} + D_{pp}(\mathbf{p}) \frac{\partial}{\partial p} \right] f(\mathbf{p}) \right\}, \quad (\text{A5})$$

respectively, with the diffusion coefficients in either set of coordinates given by

$$D_{QQ'}(\mathbf{p}) = \sum_{s=-\infty}^{\infty} \int \frac{d^3\mathbf{k}}{(2\pi)^3} w_M(\mathbf{p}, \mathbf{k}, s) \Delta Q \Delta Q' N_M(\mathbf{k}),$$

$$\Delta Q = \hat{D}_s Q,$$

$$\Delta p_{\perp} = \frac{s\Omega}{v_{\perp}}, \quad \Delta p_{\parallel} = \hbar k_{\parallel},$$

$$\Delta \alpha = \frac{\hbar(\omega \cos\alpha - k_{\parallel}v)}{pv \sin\alpha}, \quad \Delta p = \frac{\hbar\omega}{v}. \quad (\text{A6})$$

Bibliography

Primary Literature

1. Achterberg A (1981) On the nature of small amplitude Fermi acceleration. *Astron Astrophys* 97:259–264
2. Achterberg A (1990) Particle acceleration by an ensemble of shocks. *Astron Astrophys* 231:251–258
3. Achterberg A, Ball L (1994) Particle acceleration at superluminal quasi-perpendicular shocks. *Appl SN1978K SN1987A. Astron Astrophys* 285:687–704
4. Achterberg A, Gallant YA, Kirk JG, Guthmann AW (2001) Particle acceleration by ultrarelativistic shocks: theory and simulations. *Mon Not Roy Astron Soc* 328:393–408
5. Armstrong JW, Rickett BJ, Spangler SR (1995) Electron density power spectra in the local interstellar medium. *J Astrophys* 443:209–221
6. Axford WI (1994) The origins of high-energy cosmic rays. *J Astrophys Suppl* 90:937–944
7. Axford WI, Leer E, Skadron G (1977) Acceleration of cosmic rays by shock waves. 15th Internat Cosmic Ray Conf Papers (Plovdiv) 11:132–137
8. Baade W, Zwicky F (1934) Cosmic rays from super-novae. *Proc Natl Acad Sci* 20:259

9. Ball L, Melrose DB (2001) Shock drift acceleration of electrons. *Publ Astron Soc Aust* 18:361–373
10. Bell AR (1978) The acceleration of cosmic rays in shock fronts 1&2. *Mon Not Roy Astron Soc* 182:147–156; 182:443–455
11. Bell AR (2004) Turbulent amplification of magnetic field and diffusive shock acceleration of cosmic rays. *Mon Not Roy Astron Soc* 353:550–558
12. Bell AR, Lucek SG (2001) Cosmic ray acceleration to very high energy through the non-linear amplification by cosmic rays of the seed magnetic field. *Mon Not Roy Astron Soc* 321:433–438
13. Bellan PM (1996) New model for ULF Pc5 pulsations: Alfvén cones. *Geophys Res Lett* 23:1717–1720
14. Benson RF, Calvert W, Klumpp DM (1980) Simultaneous wave and particle observations in the auroral kilometric radiation source region. *Geophys Res Lett* 7:959–962
15. Berger JM, Newcomb WA, Dawson JM, Frieman EA, Kulsrud RM, Lenard A (1958) Heating of a confined plasma by oscillating electromagnetic fields. *Phys Fluids* 1:301–307
16. Bicknell GV, Melrose DB (1982) In situ acceleration in extragalactic radio jets. *J Astrophys* 262:511–528
17. Blandford RD, Ostriker FR (1978) Particle acceleration by astrophysical shocks. *J Astrophys* 221:L29–L32
18. Block LP (1978) A double layer review. *Astrophys Space Sci* 55:59–83
19. Boström R, Gustafsson G, Holback B, Holmgren G, Koskinen H (1988) Characteristics of solitary waves and weak double layers in the magnetospheric plasma. *Phys Rev Lett* 61:82–85
20. Chaston CC, Bonnell JW, Carlson CW, Berthomier M, Peticolas LM, Roth I, McFadden JP, Ergun RE, Strangeway RJ (2002) Electron acceleration in the ionospheric Alfvén resonator. *J Geophys Res* 107:SMP 41–1–16
21. Coroniti FV (1990) Magnetically striped relativistic magnetohydrodynamic winds. The Crab Nebula revisited. *J Astrophys* 349:538–545
22. Cotton WD, Wittels JJ, Shapiro II, Marcaide J, Owen FN, Spangler SR, Rius A, Angulo C, Clark TA, Knight CA (1980) The very flat radio spectrum of 0735+178: A cosmic conspiracy? *J Astrophys* 238:L123–L128
23. Davis L Jr (1956) Modified Fermi mechanism for the acceleration of cosmic rays. *Phys Rev* 101:351–358
24. Dragt AJ (1961) Effect of hydromagnetic waves of the lifetime of Van Allen radiation protons. *J Geophys Res* 66:1641–1649
25. Drury LO'C (1983) An introduction to the theory of diffusive shock acceleration of energetic particles in tenuous plasma. *Rep Prog Phys* 46:973–1027
26. Duijveman A, Hoyng P, Ionson JA (1981) Fast plasma heating by anomalous and inertial resistivity effects in the solar atmosphere. *J Astrophys* 245:721–735
27. Dungey JW (1963) Loss of Van Allen electrons due to whistlers. *Planet Space Sci* 11:591–595
28. Ergun RE, Andersson L, Main D, Su Y-J, Newman DL, Goldman MV, Carlson CW, Hull AJ, McFadden JP, Mozer FS (2004) Auroral particle acceleration by strong double layers: the upward current region. *J Geophys Res* 109:A12220
29. Ergun RE, Su Y-J, Andersson L, Bagenal F, Delemere PA, Lysak RL, Strangeway RJ (2006) S bursts and the Jupiter ionospheric Alfvén resonator. *J Geophys Res* 111:A06212
30. Fermi E (1949) On the origin of cosmic radiation. *Phys Rev* 75:1169–1174
31. Fermi E (1954) Galactic magnetic field and the origin of cosmic radiation. *J Astrophys* 119:1–6
32. Fisk LA (1978) ³He-rich flares: a possible explanation. *J Astrophys* 224:1048–1055
33. Forbes TG, Malherbe JM (1986) A shock condensation mechanism for loop prominences. *J Astrophys* 302:L67–L70
34. Ginzburg VL, Syrovatskii SI (1964) The origin of cosmic rays. Pergamon Press, Oxford
35. Goertz CK (1980) Io's interaction with the plasma torus. *J Geophys Res* 85:2949–2956
36. Goertz CK, Boswell RW (1979) Magnetosphere-ionosphere coupling. *J Geophys Res* 84:7239–7246
37. Goldreich P, Lynden-Bell D (1969) Io, a Jovian unipolar inductor. *J Astrophys* 56:59–78
38. Gurevich AV (1960) On the amount of accelerated particles in an ionized gas under various acceleration mechanisms. *Sov Phys JETP* 11:1150–1157
39. Gurnett DA (1974) The earth as a radio source: terrestrial kilometric radiation. *J Geophys Res* 79:4227–4238
40. Hall DE, Sturrock PA (1967) Diffusion, scattering, and acceleration of particles by stochastic electromagnetic fields. *Plasma Phys* 10:2620–2628
41. Heavens AF, Drury LO'C (1989) Relativistic shocks and particle acceleration. *Mon Not Roy Astron Soc* 235:997–1009
42. Helliwell RA (1967) A theory of discrete VLF emissions from the magnetosphere. *J Geophys Res* 72:4773–4790
43. Hillas AM (2006) Cosmic rays: recent progress and some current questions. In: Klöckner H-R, Jarvis M, Rawlings S (eds) *Cosmology, galaxy formation and astroparticle physics on the pathway to the SKA*. astro-ph/0607109 (to be published)
44. Hillas AM (2006) The cosmic-ray knee and ensuing spectrum seen as a consequence of Bell's SNR shock acceleration process. *J Phys Conf Ser* 47:168–177
45. Holman GD (1985) Acceleration of runaway electrons and joule heating in solar flares. *J Astrophys* 293:584–594
46. Holman GD, Pesses ME (1983) Type II radio emission and the shock drift acceleration of electrons. *J Astrophys* 267:837–843
47. Ionson JA (1982) Resonant electrodynamic heating of stellar coronal loops: an LRC circuit analog. *J Astrophys* 254:318–334
48. Ionson JA (1985) Coronal heating by resonant (AC) and non-resonant (DC) mechanisms. *Astron Astrophys* 146:199–203
49. Jones FC, Ellison DC (1987) Noncoplanar magnetic fields, shock potentials, and ion deflection. *J Geophys Res* 92:11205–11207
50. Kaplan SA (1956) The theory of the acceleration of charged particles by isotropic gas magnetic turbulent fields. *Sov Phys JETP* 2:203–210
51. Kellermann KI, Pauliny-Toth IIK (1969) The spectra of opaque radio sources. *J Astrophys* 155:L71–L78
52. Kennel CF, Petschek HE (1966) Limit on stably trapped particle fluxes. *J Geophys Res* 71:1–28
53. Kirk JG (1994) Particle Acceleration. In: Benz AO, Courvoisier TJ-L (eds) *Saas-Fee Advanced Course 24, Plasma Astrophysics*. Springer, New York, pp 225–314
54. Knight S (1973) Parallel electric fields. *Planet Space Sci* 21:741–750
55. Knock SA, Cairns IH, Robinson PA, Kuncic Z (2001) Theory of type II radio emission from the foreshock of an interplanetary shock. *J Geophys Res* 106:25041–25052
56. Krinsky GF (1977) A regular mechanism for the acceleration

- of charged particles on the front of a shock wave. *Sov Phys Doklady* 234:1306–1308
57. Kulsrud R, Ferrari A (1971) The relativistic quasilinear theory of particle acceleration by hydromagnetic turbulence. *Astrophys Space Sci* 12:302–318
 58. Lacombe C (1977) Acceleration of particles and plasma heating by turbulent Alfvén waves in a radiogalaxy. *Astron Astrophys* 54:1–16
 59. Lee MA, Ryan JM (1986) Time-dependent shock acceleration of energetic solar particles. *J Astrophys* 303:829–842
 60. Leroy MM, Mangeney A (1984) A theory of energization of solar wind electrons by the Earth's bow shock. *Ann Geophys* 2:449–455
 61. Levinson A, Melrose D, Judge A, Luo Q (2005) Large-amplitude, pair-creating oscillations in pulsar magnetospheres. *J Astrophys* 631:456–465
 62. Lin RP (2006) Particle acceleration by the Sun: electrons, hard X-rays/gamma-rays. *Space Sci Rev* 124:233–248
 63. Lyons LR (1980) Generation of large-scale regions of auroral currents, electric potentials, and precipitation by the divergence of convection electric fields. *J Geophys Res* 85:17–24
 64. Lysak RL, Song Y (2005) Nonlocal interactions between electrons and Alfvén waves on auroral field lines. *J Geophys Res* 110:A10S06
 65. Lyubarsky YE, Kirk JG (2001) Reconnection in a striped pulsar wind. *J Astrophys* 547:437–448
 66. McClements KG, Dieckmann ME, Ynnerman A, Chapman SC, Dendy RO (2001) Surfatron and stochastic acceleration of electrons at supernova remnant shocks. *Phys Rev Lett* 87:255002
 67. Malkov MA, Drury LO'C (2001) Nonlinear theory of diffusive acceleration of particles by shock waves. *Rep Prog Phys* 64:429–481
 68. Marscher AP (1977) Structure of radio sources with remarkably flat spectra: PKS 0735+178. *Astron J* 82:781–784
 69. McLean DJ, Sheridan KV, Stewart RT, Wild JP (1971) Regular pulses from the Sun and a possible clue to the origin of solar cosmic rays. *Nature* 234:140–142
 70. Meisenheimer K, Röser H-J, Schlötelburg M (1996) *Astron Astrophys* 307:61–79
 71. Melatos A, Melrose DB (1996) Energy transport in a rotation-modulated pulsar wind. *Mon Not Roy Astron Soc* 279:1168–1190
 72. Melrose DB (1968) The emission and absorption of waves by charged particles in magnetized plasmas. *Astrophys Space Sci* 2:171–235
 73. Melrose DB (2005) Nonresonant Alfvén waves driven by cosmic rays. In: Li G, Zank GP, Russell CT (eds) *The physics of collisionless shocks*. AIP Conference Proceedings #781, American Institute of Physics, pp 135–140
 74. Melrose DB, Wentzel DG (1970) The interaction of cosmic-ray electrons with cosmic-ray protons. *J Astrophys* 161:457–476
 75. Melrose D, Couch A (1997) Effect of synchrotron losses on multiple diffusive shock acceleration. *Australian J Phys* 14:251–257
 76. Micono M, Zurlo N, Massaglia S, Ferrari A, Melrose DB (1999) Diffusive shock acceleration in extragalactic jets. *Astron Astrophys* 349:323–333
 77. Miller JA, Cargill PJ, Emslie AG, Holman GD, Dennis BR, LaRosa TN, Winglee RM, Benka SG, Tsuneta S (1997) Critical issues for understanding particle acceleration in impulsive solar flares. *J Geophys Res* 102:4631–4660
 78. Mozer FS, Cattell CA, Hudson MK, Lysak RL, Temerin M, Torbert RB (1980) Satellite measurements and theories of low altitude auroral particle acceleration. *Space Sci Rev* 27: 155–213
 79. Nelson GJ, Melrose DB (1985) Type II bursts. In: McLean DJ, Labrum NR (eds) *Solar Radiophysics*, chapter 13. Cambridge University Press, pp 333–359
 80. Newman DL, Goldman MV, Ergun RE, Mangeney A (2001) Formation of double layers and electron holes in a current-driven space plasma. *Phys Rev Lett* 87:255001
 81. Parizot E, Marcowith A, Ballet J, Gallant YA (2006) Observational constraints on energetic particle diffusion in young supernovae remnants: amplified magnetic field and maximum energy. *Astron Astrophys* 453:387–395
 82. Parker EN (1957) Acceleration of cosmic rays in solar flares. *Phys Rev* 107:830–836
 83. Parker EN, Tidman DA (1958) Suprathermal particles. *Phys Rev* 111:1206–1211
 84. Pesses ME (1981) On the conservation of the first adiabatic invariant in perpendicular shocks. *J Geophys Res* 86: 150–152
 85. Raadu MA (1989) The physics of double layers and their role in astrophysics. *Phys Rep* 178:25–97
 86. Reville B, Kirk JG, Duffy P (2006) A current-driven instability in parallel, relativistic shocks. *Plasma Phys Cont Fus* 48:1741–1747
 87. Schlickeiser R (1984) An explanation of abrupt cutoffs in the optical-infrared spectra of non-thermal sources. A new pile-up mechanism for relativistic electron spectra. *Astron Astrophys* 136:227–236
 88. Schlüter A (1957) Der Gyro-Relaxations-Effekt. *Z Nat* 12a:822–825
 89. Schmitz H, Chapman SC, Dendy RO (2002) Electron preacceleration mechanisms in the foot region of high Alfvénic Mach number shocks. *J Astrophys* 579:327–336
 90. Shen CS (1965) Transit acceleration of charged particles in an inhomogeneous electromagnetic field. *J Astrophys* 141: 1091–1104
 91. Speiser TW (1965) Particle trajectories in model current sheets, 1. Analytic solutions. *J Geophys Res* 70:1717–1788
 92. Stasiewicz K, et al. (2000) Small scale Alfvénic structures in the aurora. *Space Sci Rev* 92:423–533
 93. Sturrock PA (1966) Stochastic acceleration. *Phys Rev* 141:186–191
 94. Swann WFG (1933) A mechanism of acquirement of cosmic-ray energies by electrons. *Phys Rev* 43:217–220
 95. Thompson WB (1955) On the acceleration of cosmic-ray particles by magneto-hydrodynamic waves. *Proc Roy Soc Lond* 233:402–406
 96. Toptygin IN (1980) Acceleration of particles by shocks in a cosmic plasma. *Space Sci Rev* 26:157–213
 97. Tsytovich VN (1966) Statistical acceleration of particles in a turbulent plasma. *Sov Phys Uspeki* 9:370–404
 98. Tverskoï BA (1967) Contribution to the theory of Fermi statistical acceleration. *Sov Phys JETP* 25:317–325
 99. Tverskoï BA (1968) Theory of turbulent acceleration of charged particles in a plasma. *Sov Phys JETP* 26:821–828
 100. Ucer D, Shapiro VD (2001) Unlimited relativistic shock surfing acceleration. *Phys Rev Lett* 87:075001

101. van der Swaluw E, Achterberg A (2004) Non-thermal X-ray emission from young supernova remnants. *Astron Astrophys* 421:1021–1030
102. Vedin J, Rönmark K (2004) A linear auroral current-voltage relation in fluid theory. *Ann Geophys* 22:1719–1728
103. Vogt J (2002) Alfvén wave coupling in the auroral current circuit. *Surveys Geophys* 23:335–377
104. Völk HJ (2006) Shell-type supernova remnants. *astro-ph/0603502*
105. Wentzel DG (1961) Hydromagnetic waves and the trapped radiation Part 1. Breakdown of the adiabatic invariance. *J Geophys Res* 66:359–369
106. Wentzel DG (1964) Motion across magnetic discontinuities and Fermi acceleration of charged particles. *J Astrophys* 140:1013–1024
107. Wiersma J, Achterberg A (2004) Magnetic field generation in relativistic shocks. An early end of the exponential Weibel instability in electron-proton plasmas. *Astron Astrophys* 428:365–371
108. Wild JP, Smerd SF, Weiss AA (1963) Solar bursts. *Ann Rev Astron Astrophys* 1:291–366
109. Wygant JR, Keiling A, Cattell CA, Lysak RL, Temerin M, Mozer FS, Kletzing CA, Scudder JD, Streltsov V, Lotko W, Russell CT (2002) Evidence for kinetic Alfvén waves and parallel electron energization at 4–6 RE altitudes in the plasma sheet boundary layer. *J Geophys Res* 107:A900113
- Vogt J (2002) Alfvén wave coupling in the auroral current circuit. *Surv Geophys* 23:335–377

Books and Reviews

- Alfvén H, Fälthammar C-G (1963) *Cosmical Electrodynamics*. Oxford University Press, Oxford
- Aschwanden MJ (2004) *Physics of the solar corona*. Springer, Berlin
- Benz AO (1993) *Plasma astrophysics: kinetic processes in solar and stellar coronae*. Kluwer Academic Publishers, Dordrecht
- Berezinskii VS, Bulanov SV, Dogiel VA, Ginzburg VL, Ptuskin VS (1990) *Astrophysics of cosmic rays*. North Holland, Amsterdam
- Dorman LI (2006) *Cosmic ray interactions, propagation, and acceleration in space plasmas*. Springer, New York
- Drury LO'C (1983) An introduction to the theory of diffusive shock acceleration of energetic particles in tenuous plasmas. *Rep Prog Phys* 46:973–1027
- McLean DJ, Labrum NR (eds) (1986) *Solar Radiophysics*. Cambridge University Press, Cambridge
- Melrose DB (1980) *Plasma Astrophysics, vol I & II*. Gordon, New York
- Melrose DB (1986) *Instabilities in Space and Laboratory Plasmas*. Cambridge University Press
- Malkov MA, Drury LO'C (2001) Nonlinear theory of diffusive acceleration of particles by shock waves. *Rep Prog Phys* 64:429–481
- Michel FC (1991) *Physics of neutron star magnetospheres*. The University of Chicago Press, Chicago
- Priest ER, Forbes T (2000) *Magnetic reconnection—MHD theory and applications*. Cambridge University Press, Cambridge
- Schlickeiser R (2002) *Cosmic ray astrophysics*. Springer, Berlin
- Stone RG, Tsurutani BT (eds) (1985) *Collisionless shocks in the heliosphere: a tutorial review*. Geophysical Monograph 34. American Geophysical Union, Washington DC
- Sturrock PA (1980) *Solar flares*. Colorado Associated University Press, Boulder
- Svestka Z (1976) *Solar Flares*. D Reidel, Dordrecht

Adaptive Visual Servo Control

GUOQIANG HU¹, NICHOLAS GANS²,
WARREN E. DIXON²

¹ Department of Mechanical and Nuclear Engineering,
Kansas State University, Manhattan, USA

² Department of Mechanical and Aerospace Engineering,
University of Florida, Gainesville, USA

Article Outline

[Glossary](#)

[Definition of the Subject](#)

[Introduction](#)

[Visual Servo Control Methods](#)

[Compensating for Projection Normalization](#)

[Visual Servo Control via an Uncalibrated Camera](#)

[Future Directions](#)

[Bibliography](#)

Glossary

Camera-in-hand configuration

The camera-in-hand configuration refers to the case when the camera is attached to a moving robotic system (e. g., held by the robot end-effector).

Camera-to-hand configuration

The camera-to-hand configuration refers to the case when the camera is stationary and observing moving targets (e. g., a fixed camera observing a moving robot end-effector).

Euclidean reconstruction Euclidean reconstruction is the act of reconstructing the Euclidean coordinates of feature points based on the two-dimensional image information obtained from a visual sensor.

Extrinsic calibration parameters The extrinsic calibration parameters are defined as the relative position and orientation of the camera reference frame to the world reference frame (e. g., the frame affixed to the robot base). The parameters are represented by a rotation matrix and a translation vector.

Feature point Different computer vision algorithms have been developed to search images for distinguishing features in an image (e. g., lines, points, corners, textures). Given a three dimensional Euclidean point, its projection to the two dimensional image plane is called a feature point. Feature points are selected based on

contrast with surrounding pixels and the ability for the point to be tracked from frame to frame. Sharp corners (e.g., the windows of a building), or center of a homogenous region in the image (e.g., the center of car headlights) are typical examples of feature points.

Homography The geometric concept of homography is a one-to-one and on-to transformation or mapping between two sets of points. In computer vision, homography refers to the mapping between points in two Euclidean-planes (Euclidean homography), or to the mapping between points in two images (projective homography).

Image Jacobian The image-Jacobian is also called the interaction matrix, feature sensitivity matrix, or feature Jacobian. It is a transformation matrix that relates the change of the pixel coordinates of the feature points on the image to the velocity of the camera.

Intrinsic calibration parameters The intrinsic calibration parameters map the coordinates from the normalized Euclidean-space to the image-space. For the pinhole camera, this invertible mapping includes the image center, camera scale factors, and camera magnification factor.

Pose The position and orientation of an object is referred to as the pose. The pose has a translation component that is an element of \mathbb{R}^3 (i.e., Euclidean-space), and the rotation component is an element of the special orthogonal group $SO(3)$, though local mappings of rotation that exist in \mathbb{R}^3 (Euler angles, angle/axis) or \mathbb{R}^4 (unit quaternions). Pose of an object has specific meaning when describing the relative position and orientation of one object to another or the position and orientation of an object at different time instances. Pose is typically used to describe the position and orientation of one reference frame with respect to another frame.

Projection normalization The Euclidean coordinates of a point are projected onto a two-dimensional image via the projection normalization. For a given camera, a linear projective matrix can be used to map points in the Euclidean-space to the corresponding pixel coordinates on the image-plane. This relationship is not bijective because there exist infinite number of Euclidean points that have the same normalized coordinates, and the depth information is lost during the normalization.

Unit quaternion The quaternion is a four dimensional vector which is composed of real numbers (i.e., $q \in \mathbb{R}^4$). The unit quaternion is a quaternion subject to a nonlinear constraint that the norm is equal to 1. The unit quaternion can be used as a globally nonsingular representation of a rotation matrix.

Visual servo control system Control systems that use information acquired from an imaging source (e.g., a digital camera) in the feedback control loop are defined as visual servo control systems. In addition to providing feedback relating the local position and orientation (i.e., pose) of the camera with respect to some target, an image sensor can also be used to relate local sensor information to an inertial reference frame for global control tasks. Visual servoing requires multidisciplinary expertise to integrate a vision system with the controller for tasks including: selecting the proper imaging hardware; extracting and processing images at rates amenable to closed-loop control; image analysis and feature point extraction/tracking; recovering/estimating necessary state information from an image; and operating within the limits of the sensor range (i.e., field-of-view).

Lyapunov function A Lyapunov function is a continuously differentiable positive definite function with a negative definite or semi-definite time-derivative.

Definition of the Subject

Control systems that use information acquired from an imaging source in the feedback loop are defined as visual servo control systems. Visual servo control has developed into a large subset of robotics literature because of the enabling capabilities it can provide for autonomy. The use of an image sensor for feedback is motivated by autonomous task execution in an unstructured or unknown environments. In addition to providing feedback relating the local position and orientation (i.e., pose) of the camera with respect to some target, an image sensor can also be used to relate local sensor information to an inertial reference frame for global control tasks.

However, the use of image-based feedback adds complexity and new challenges for the control system design. Beyond expertise in design of mechanical control systems, visual servoing requires multidisciplinary expertise to integrate a vision system with the controller for tasks including: selecting the proper imaging hardware; extracting and processing images at rates amenable to closed-loop control; image analysis and feature point extraction/tracking; recovering/estimating necessary state information from an image; and operating within the limits of the sensor range (i.e., field-of-view). While each of the aforementioned tasks are active topics of research interest, the scope of this chapter is focused on issues associated with using reconstructed and estimated state information from an image to develop a stable closed-loop error system. That is, the topics in this chapter are based on the assumption that images

can be acquired, analyzed, and the resulting data can be provided to the controller without restricting the control rates.

Introduction

The different visual servo control methods can be divided into three main categories including: image-based, position-based, and approaches that use of blend of image and position-based approaches. Image-based visual servo control (e.g., [1,2,3,4,5]) consists of a feedback signal that is composed of pure image-space information (i.e., the control objective is defined in terms of an image pixel error). This approach is considered to be more robust to camera calibration and robot kinematic errors and is more likely to keep the relevant image features in the field-of-view than position-based methods because the feedback is directly obtained from the image without the need to transfer the image-space measurement to another space. A drawback of image-based visual servo control is that since the controller is implemented in the robot joint space, an image-Jacobian is required to relate the derivative of the image-space measurements to the camera's linear and angular velocities. However, the image-Jacobian typically contains singularities and local minima [6], and the controller stability analysis is difficult to obtain in the presence of calibration uncertainty [7]. Another drawback of image-based methods is that since the controller is based on image-feedback, the robot could be commanded to some configuration that is not physically possible. This issue is described as Chaumette's conundrum (see [6,8]).

Position-based visual servo control (e.g., [1,5,9,10,11]) uses reconstructed Euclidean information in the feedback loop. For this approach, the image-Jacobian singularity and local minima problems are avoided, and physically realizable trajectories are generated. However, the approach is susceptible to inaccuracies in the task-space reconstruction if the transformation is corrupted (e.g., uncertain camera calibration). Also, since the controller does not directly use the image features in the feedback, the commanded robot trajectory may cause the feature points to leave the field-of-view. A review of these two approaches is provided in [5,12,13].

The third class of visual servo controllers use some image-space information combined with some reconstructed information as a means to combine the advantages of these two approaches while avoiding their disadvantages (e.g., [8,14,15,16,17,18,19,20,21,22,23]). One particular approach was coined 2.5D visual servo control in [15,16,17,18] because this class of controllers exploits some two dimensional image feedback and recon-

structed three-dimensional feedback. This class of controllers is also called homography-based visual servo control in [19,20,21,22] because of the underlying reliance of the construction and decomposition of a homography.

A key issue that is pervasive to all visual servo control methods is that the image-space is a two-dimensional projection of the three-dimensional Euclidean-space. To compensate for the lack of depth information from the two-dimensional image data, some control methods exploit additional sensors (e.g., laser and sound ranging technologies) along with sensor fusion methods or the use of additional cameras in a stereo configuration that triangulate on corresponding images. However, the practical drawbacks of incorporating additional sensors include: increased cost, increased complexity, decreased reliability, and increased processing burden. Geometric knowledge of the target or scene can be used to estimate depth, but this information is not available for many unstructured tasks.

Motivated by these practical constraints, direct approximation and estimation techniques (cf. [4,5]) and dynamic adaptive compensation (cf. [19,20,21,22,24,25,26,27,28]) have been developed. For example, an adaptive kinematic controller has been developed in [24] to achieve a uniformly ultimately bounded regulation control objective provided conditions on the translation velocity and bounds on the uncertain depth parameters are satisfied. In [24], a three dimensional depth estimation procedure has been developed that exploits a prediction error provided a positive definite condition on the interaction matrix is satisfied. In [20,26,27], homography-based visual servo controllers have been developed to asymptotically regulate a manipulator end-effector and a mobile robot, respectively, by developing an adaptive update law that actively compensates for an unknown depth parameter. In [19,21], adaptive homography-based visual servo controllers have been developed to achieve asymptotic tracking control of a manipulator end-effector and a mobile robot, respectively. In [22,28], adaptive homography-based visual servo controllers via a quaternion formulation have been developed to achieve asymptotic regulation and tracking control of a manipulator end-effector, respectively.

In addition to normalizing the three-dimensional coordinates, the projection from the Euclidean-space to the image-space also involves a transformation matrix that may contain uncertainties. Specifically, a camera model (e.g., the pinhole model) is often required to relate pixel coordinates from an image to the (normalized) Euclidean coordinates. The camera model is typically assumed to be exactly known (i.e., the intrinsic calibration parameters are assumed to be known); however, de-

spite the availability of several popular calibration methods (cf. [29,30,31,32,33,34,35]), camera calibration can be time-consuming, requires some level of expertise, and has inherent inaccuracies. If the calibration parameters are not exactly known, performance degradation and potential unpredictable response from the visual servo controller may occur.

Motivated by the desire to incorporate robustness to camera calibration, different control approaches that do not depend on exact camera calibration have been proposed (cf. [18,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51]). Efforts such as [36,37,38,39,40] have investigated the development of methods to estimate the image and robot manipulator Jacobians. These methods are composed of some form of recursive Jacobian estimation law and a control law. Specifically, a visual servo controller is developed in [36] based on a weighted recursive least-squares update law to estimate the image Jacobian. In [37], a Broyden–Jacobian estimator is applied and a nonlinear least-square optimization method is used for the visual servo control development. In [38], the authors used a nullspace-biased Newton-step visual servo strategy with a Broyden Jacobian estimation for online singularity detection and avoidance in an uncalibrated visual servo control problem. In [39,40] a recursive least-squares algorithm is implemented for Jacobian estimation, and a dynamic Gauss–Newton method is used to minimize the squared error in the image plane.

In [41,42,43,44,45,46,47], robust and adaptive visual servo controllers have been developed. The development of traditional adaptive control methods to compensate for the uncertainty in the transformation matrix is inhibited because of the time-varying uncertainty injected in the transformation from the normalization of the Euclidean coordinates. As a result, initial adaptive control results were limited to scenarios where the optic axis of the camera was assumed to be perpendicular with the plane formed by the feature points (i. e., the time-varying uncertainty is reduced to a constant uncertainty). Other adaptive methods were developed that could compensate for the uncertain calibration parameters assuming an additional sensor (e. g., ultrasonic sensors, laser-based sensors, additional cameras) could be used to measure the depth information.

More recent approaches exploit geometric relationships between multiple spatiotemporal views of an object to transform the time-varying uncertainty into known time-varying terms multiplied by an unknown constant [18,48,49,50,51,52]. In [48], an on-line calibration algorithm was developed for position-based visual servoing. In [49], an adaptive controller was developed for image-

based dynamic control of a robot manipulator. One problem with methods based on the image-Jacobian is that the estimated image-Jacobian may contain singularities. The development in [49] exploits an additional potential force function to drive the estimated parameters away from the values that result in a singular Jacobian matrix. In [52], an adaptive homography-based controller was proposed to address problems of uncertainty in the intrinsic camera calibration parameters and lack of depth measurements. In Subsect. “[Adaptive Homography-Based Visual Servo Control Approach](#)”, a combined high-gain control approach and an adaptive control strategy are proposed to regulate the robot end-effector to a desired pose asymptotically with considering this time-varying scaling factor.

Robust control approaches based on static best-guess estimation of the calibration matrix have been developed to solve the uncalibrated visual servo regulation problem (cf. [18,44,50,51]). Specifically, under a set of assumptions on the rotation and calibration matrix, a kinematic controller was developed in [44] that utilizes a constant, best-guess estimate of the calibration parameters to achieve local set-point regulation for the six degree-of-freedom visual servo control problem. Homography-based visual servoing methods using best-guess estimation are used in [18,50,51] to achieve asymptotic or exponential regulation with respect to both camera and hand-eye calibration errors for the six degree-of-freedom problem.

This chapter is organized to provide the basic underlying principles of both historic and recent visual servo control methods with a progression to open research issues due to the loss of information through the image projection and the disturbances due to uncertainty in the camera calibration. Section “[Visual Servo Control Methods](#)” provides an overview of the basic methodologies. This section includes topics such as: geometric relationships and Euclidean reconstruction using the homography in Subsect. “[Geometric Relationships](#)”; and image-based visual servo control, position-based visual servo control, and hybrid visual servo control approaches in Subsects. “[Image-based Visual Servo Control](#)”–“[Homography-Based Visual Servo Control](#)”. In Sect. “[Compensating for Projection Normalization](#)”, an adaptive technique that actively compensates for unknown depth parameters in visual servo control is presented. Specifically, this adaptive compensation approach has been combined with the homography-based visual servo control scheme and image-based visual servo control scheme to generate asymptotic results in Subsects. “[Adaptive Depth Compensation in Homography-Based Visual Servo Control](#)” and “[Adaptive Depth Compensation in Image-Based Visual Servo Control](#)”, respectively. In Sect. “[Visual Servo Con-](#)

control via an Uncalibrated Camera”, several visual control approaches that do not require exact camera calibration knowledge are presented. For example, in Subsect. “**Jacobian Estimation Approach**”, a visual servo controller that is based on a Jacobian matrix estimator is described. In Subsect. “**Robust Control Approach**”, a robust controller based on static best-guess estimation of the calibration matrix is developed to solve the regulation problem. In Subsect. “**Adaptive Image-Based Visual Servo Control Approach**”, an adaptive controller for image-based dynamic regulation control of a robot manipulator is described based on a depth-independent interaction matrix. In Subsect. “**Adaptive Homography-Based Visual Servo Control Approach**”, a combined high-gain control approach and an adaptive control strategy are used to regulate the robot end-effector to a desired pose based on the homography technique.

Visual Servo Control Methods

The development in this chapter is focused on the development of visual servo controllers for robotic systems. To develop the controllers, the subsequent development is based on the assumption that images can be acquired, analyzed, and the resulting data can be provided to the controller without restricting the control rates. The sensor data from image-based feedback are feature points. Feature points are pixels in an image that can be identified and tracked between images so that the motion of the camera/robot can be discerned from the image. Image processing techniques can be used to select coplanar and non-collinear feature points within an image. For simplicity, the development in this chapter is based on the assumption that four stationary coplanar and non-collinear feature points [53] denoted by $O_i \forall i = 1, 2, 3, 4$ can be determined from a feature point tracking algorithm (e.g., Kanade–Lucas–Tomasi (KLT) algorithm discussed in [54,55]).

The plane defined by the four feature points is denoted by π as depicted in Fig. 1. The assumption that feature points lie on a plane is not restrictive and does not reduce the generality of the results. For example, if four coplanar target points are not available then the subsequent development can also exploit methods such as the virtual parallax algorithm (e.g., see [16,56]) to create virtual planes from non-planar feature points.

Geometric Relationships

The camera-in-hand configuration is depicted in Fig. 1. In Fig. 1, \mathcal{F} denotes a coordinate frame that is considered to be affixed to the single current camera viewing the object,

and a stationary coordinate frame \mathcal{F}^* denotes a constant (a priori determined) desired camera position and orientation that is defined as a desired image. The Euclidean coordinates of the feature points O_i expressed in the frames \mathcal{F} and \mathcal{F}^* are denoted by $\bar{m}_i(t) \in \mathbb{R}^3$ and $\bar{m}_i^* \in \mathbb{R}^3$, respectively, as

$$\bar{m}_i \triangleq [x_i(t) \quad y_i(t) \quad z_i(t)]^T, \quad \bar{m}_i^* \triangleq [x_i^* \quad y_i^* \quad z_i^*]^T, \quad (1)$$

where $x_i(t), y_i(t), z_i(t) \in \mathbb{R}$ denote the Euclidean coordinates of the i th feature point, and $x_i^*, y_i^*, z_i^* \in \mathbb{R}$ denote the Euclidean coordinates of the corresponding feature points in the desired/reference image. From standard Euclidean geometry, relationships between $\bar{m}_i(t)$ and \bar{m}_i^* can be determined as

$$\bar{m}_i = x_f + R\bar{m}_i^*, \quad (2)$$

where $R(t) \in SO(3)$ denotes the orientation of \mathcal{F}^* with respect to \mathcal{F} , and $x_f(t) \in \mathbb{R}^3$ denotes the translation vector from \mathcal{F} to \mathcal{F}^* expressed in the coordinate frame \mathcal{F} . The normalized Euclidean coordinate vectors, denoted by $m_i(t) \in \mathbb{R}^3$ and $m_i^* \in \mathbb{R}^3$, are defined as

$$m_i = [m_{xi} \quad m_{yi} \quad 1]^T \triangleq \left[\frac{x_i}{z_i} \quad \frac{y_i}{z_i} \quad 1 \right]^T, \quad (3)$$

$$m_i^* = [m_{xi}^* \quad m_{yi}^* \quad 1]^T \triangleq \left[\frac{x_i^*}{z_i^*} \quad \frac{y_i^*}{z_i^*} \quad 1 \right]^T. \quad (4)$$

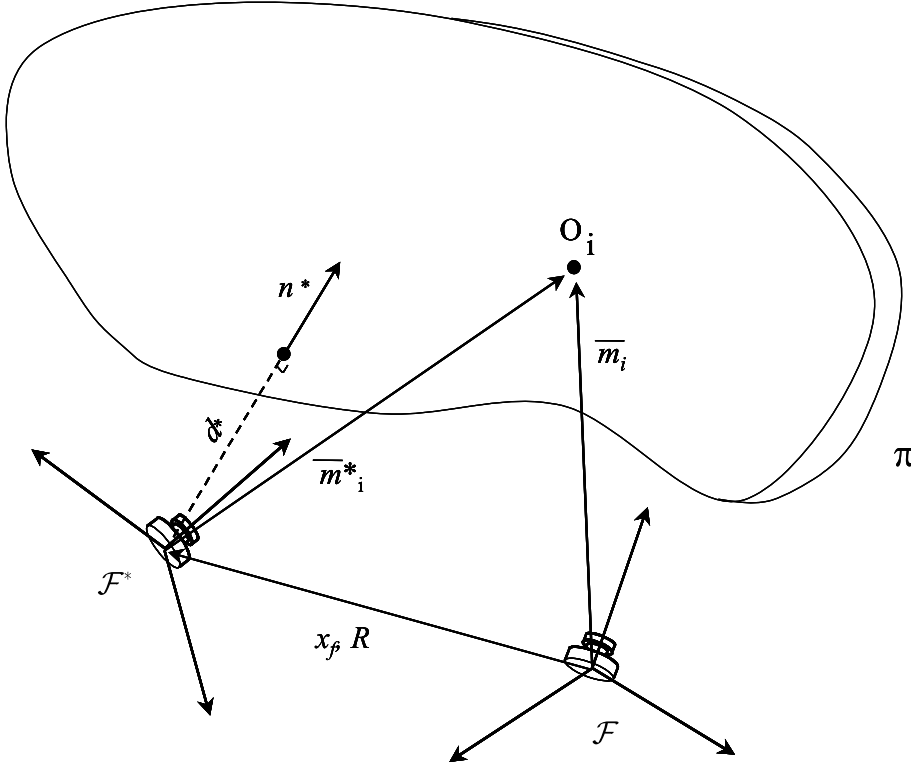
Based on the Euclidean geometry in Eq. (2), the relationships between $m_i(t)$ and m_i^* can be determined as [53]

$$m_i = \underbrace{\frac{z_i^*}{z_i}}_{\alpha_i} \underbrace{\left(R + \frac{x_f}{d^*} n^{*T} \right)}_H m_i^*, \quad (5)$$

where $\alpha_i(t) \in \mathbb{R}$ is a scaling term, and $H(t) \in \mathbb{R}^{3 \times 3}$ denotes the Euclidean homography. The Euclidean homography is composed of: a scaled translation vector which is equal to the translation vector $x_f(t)$ divided by the distance $d^* \in \mathbb{R}$ from the origin of \mathcal{F}^* to the plane π ; the rotation between \mathcal{F} and \mathcal{F}^* denoted by $R(t)$; and $n^* \in \mathbb{R}^3$ denoting a constant unit normal to the plane π .

Each feature point O_i on π also has a pixel coordinate $p_i(t) \in \mathbb{R}^3$ and $p_i^* \in \mathbb{R}^3$ expressed in the image coordinate frame for the current image and the desired image denoted by

$$p_i \triangleq [u_i \quad v_i \quad 1]^T, \quad p_i^* \triangleq [u_i^* \quad v_i^* \quad 1]^T, \quad (6)$$



Adaptive Visual Servo Control, Figure 1

Coordinate frame relationships between a camera viewing a planar patch at different spatiotemporal instances

where $u_i(t), v_i(t), u_i^*, v_i^* \in \mathbb{R}$. The pixel coordinates $p_i(t)$ and p_i^* are related to the normalized task-space coordinates $m_i(t)$ and m_i^* by the following global invertible transformation (i. e., the pinhole camera model)

$$p_i = A m_i, \quad p_i^* = A m_i^*, \quad (7)$$

where $A \in \mathbb{R}^{3 \times 3}$ is a constant, upper triangular, and invertible intrinsic camera calibration matrix that is explicitly defined as [57]

$$A \triangleq \begin{bmatrix} \alpha & -\alpha \cot \varphi & u_0 \\ 0 & \frac{\beta}{\sin \varphi} & v_0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix}, \quad (8)$$

where $a_{11}, a_{12}, a_{13}, a_{22}, a_{23} \in \mathbb{R}$. In Eq. (8), $u_0, v_0 \in \mathbb{R}$ denote the pixel coordinates of the principal point (i. e., the image center that is defined as the frame buffer coordinates of the intersection of the optical axis with the image plane), $\alpha, \beta \in \mathbb{R}$ represent the product of the camera scaling factors and the focal length, and $\varphi \in \mathbb{R}$ is the skew angle between the camera axes. Based on the physical meaning, the diagonal calibration elements are positive

(i. e., $a_{11}, a_{22} > 0$). Based on Eq. (7), the Euclidean relationship in Eq. (5) can be expressed in terms of the image coordinates as

$$p_i = \alpha_i \underbrace{(A H A^{-1})}_{G} p_i^*, \quad (9)$$

where $G(t) \in \mathbb{R}^{3 \times 3}$ denotes the projective homography.

Based on the feature point pairs $(p_i^*, p_i(t))$ with $i = 1, 2, 3, 4$, the projective homography can be determined up to a scalar multiple (i. e., $\frac{G(t)}{g_{33}(t)}$ where $g_{33}(t) \in \mathbb{R}$ denotes the bottom right element of $G(t)$) [19]. Various methods can then be applied (e. g., see [58,59]) to decompose the Euclidean homography to obtain the rotation matrix $R(t)$, scaled translation $\frac{x_f(t)}{d^*}$, normal vector n^* , and the depth ratio $\alpha_i(t)$.

Image-Based Visual Servo Control

As described previously, image-based visual servo control uses pure image-space information (i. e., pixel coordinates) in the control development. The image error for the feature point O_i between the current and desired poses

is defined as

$$e_{ii}(t) = \begin{bmatrix} u_i(t) - u_i^* \\ v_i(t) - v_i^* \end{bmatrix} \in \mathbb{R}^2.$$

The derivative of $e_{ii}(t)$ can be written as (e.g., see [3,4,5])

$$\dot{e}_{ii} = L_{ii} \begin{bmatrix} v_c \\ \omega_c \end{bmatrix}, \quad (10)$$

where $L_{ii}(m_i, z_i) \in \mathbb{R}^{2n \times 6}$ is defined as

$$L_{ii} = \begin{bmatrix} a_{11} & a_{12} \\ 0 & a_{22} \end{bmatrix} \times \begin{bmatrix} -\frac{1}{z_i} & 0 & \frac{m_{xi}}{z_i} & m_{xi}m_{yi} & -(1+m_{xi}^2) & m_{yi} \\ 0 & -\frac{1}{z_i} & \frac{m_{yi}}{z_i} & 1+m_{yi}^2 & -m_{xi}m_{yi} & -m_{xi} \end{bmatrix},$$

and $v_c(t), \omega_c(t) \in \mathbb{R}^3$ are linear and angular velocities of the camera, respectively, expressed in the camera coordinate frame \mathcal{F} . If the image error is defined based on the normalized pixel coordinates as

$$e_{ii}(t) = \begin{bmatrix} m_{xi}(t) - m_{xi}^* \\ m_{yi}(t) - m_{yi}^* \end{bmatrix} \in \mathbb{R}^2$$

then the derivative of $e_{ii}(t)$ can be written as (e.g., see [13])

$$\dot{e}_{ii} = \begin{bmatrix} -\frac{1}{z_i} & 0 & \frac{m_{xi}}{z_i} & m_{xi}m_{yi} & -(1+m_{xi}^2) & m_{yi} \\ 0 & -\frac{1}{z_i} & \frac{m_{yi}}{z_i} & 1+m_{yi}^2 & -m_{xi}m_{yi} & -m_{xi} \end{bmatrix} \times \begin{bmatrix} v_c \\ \omega_c \end{bmatrix}. \quad (11)$$

The image error for n ($n \geq 3$) feature points is defined as

$$e_i(t) = [e_{i1}^T(t) \quad e_{i2}^T(t) \quad \cdots \quad e_{in}^T(t)]^T \in \mathbb{R}^{2n}.$$

Based on Eq. (10), the open-loop error system for $e_i(t)$ can be written as

$$\dot{e}_i = L_i \begin{bmatrix} v_c \\ \omega_c \end{bmatrix}, \quad (12)$$

where the matrix

$$L_i(t) = [L_{i1}^T(t) \quad L_{i2}^T(t) \quad \cdots \quad L_{in}^T(t)]^T \in \mathbb{R}^{2n \times 6}$$

is called the image Jacobian, interaction matrix, feature sensitivity matrix, or feature Jacobian (cf. [2,3,4,5]). When the image Jacobian matrix is nonsingular at the desired pose (i.e., $e_i(t) = 0$), and the depth information $z_i(t)$ is

available, a locally exponentially result (i.e., $e_i(t) \rightarrow 0$ exponentially) can be obtained by using the controller

$$\begin{bmatrix} v_c \\ \omega_c \end{bmatrix} = -k_i L_i^+ e_i, \quad (13)$$

where $k_i \in \mathbb{R}$ is a positive control gain, and $L_i^+(t)$ is the pseudo inverse of $L_i(t)$. See [13] for details regarding the stability analysis.

In the controller Eq. (13), $z_i(t)$ is unknown. Direct approximation and estimation techniques (cf. [4,5]) have been developed to estimate $z_i(t)$. For example, an approximate estimate of $L_i(t)$ using constant z_i^* at the desired pose has been used in [4]. Besides the direct approximation and estimation approaches, dynamic adaptive compensation approaches (cf. [19,20,21,22,24,25,26,27,28]) have also been developed recently to compensate for this unknown depth information, which will be represented in Sect. “Compensating for Projection Normalization”.

Position-Based Visual Servo Control

The feedback loop in position-based visual servo control (e.g., [1,5,9,10,11]) uses Euclidean pose information that is reconstructed from the image information. Various methods can be applied (cf. [5,60,61]) to compute the pose estimate based on the image information and a priori geometric knowledge of the environment. The resulting outcome of these methods are the translation vector $x_f(t)$ and rotation matrix $R(t)$ between the current and desired poses of the robot end-effector. The rotation matrix can be further decomposed as a rotation axis $u(t) \in \mathbb{R}^3$ and a rotation angle $\theta(t) \in \mathbb{R}$.

The pose error $e_p(t) \in \mathbb{R}^6$ can be defined as

$$e_p(t) = \begin{bmatrix} x_f \\ u\theta \end{bmatrix},$$

where the corresponding open-loop error system can then be determined as

$$\dot{e}_p = L_p \begin{bmatrix} v_c \\ \omega_c \end{bmatrix}, \quad (14)$$

where $L_p(t) \in \mathbb{R}^{6 \times 6}$ is defined as

$$L_p = \begin{bmatrix} R & 0 \\ 0 & L_\omega \end{bmatrix}. \quad (15)$$

In Eq. (15), $L_\omega(u(t), \theta(t)) \in \mathbb{R}^{3 \times 3}$ is defined as [15]

$$L_\omega = I_3 - \frac{\theta}{2} [u]_\times + \left(1 - \frac{\text{sinc}(\theta)}{\text{sinc}^2\left(\frac{\theta}{2}\right)} \right) [u]_\times^2. \quad (16)$$

Based on the open-loop error system in Eq. (14), an example position-based visual servo controller can be designed as

$$\begin{bmatrix} v_c \\ \omega_c \end{bmatrix} = -k_p L_p^{-1} e_p \quad (17)$$

provided that $L_p(t)$ is nonsingular, where $k_p \in \mathbb{R}$ is a positive control gain. The controller in Eq. (17) yields an asymptotically stable result in the sense that $e_p(t)$ asymptotically converges to zero. See [13] for details regarding the stability analysis.

A key issue with position-based visual servo control is the need to reconstruct the Euclidean information to find the pose estimate from the image. Given detailed geometric knowledge of the scene (such as a CAD model) approaches such as [5,60,61] can deliver pose estimates. When geometric knowledge is not available, homography-based Euclidean reconstruction is also a good option. Given a set of feature points as stated in Subsect. “[Geometric Relationships](#)” and based on Eqs. (5) and (7), the homography $H(t)$ between the current pose and desired pose can be obtained if the camera calibration matrix A is known. Various methods can then be applied (e.g., see [58,59]) to decompose the homography to obtain the rotation matrices $R(t)$ and the scaled translation vector $\frac{x_f(t)}{d^*}$. If $\frac{x_f(t)}{d^*}$ is used in the control development, then either d^* should be estimated directly or adaptive control techniques would be required to compensate for/identify the unknown scalar.

Homography-Based Visual Servo Control

The 2.5D visual servo control approach developed in [15, 16,17,18] decomposes the six-DOF motion into translation and rotation components and uses separate translation and rotation controllers to achieve the asymptotic regulation. This class of controllers is also called homography-based visual servo control in [19,20,21,22] because of the underlying reliance of the construction and decomposition of a homography.

The translation error signal $e_v(t) \in \mathbb{R}^3$ can be chosen as a combination of image information (or normalized Euclidean information) and reconstructed Euclidean information as

$$e_v = \begin{bmatrix} m_{xi} - m_{xi}^* & m_{yi} - m_{yi}^* & \ln\left(\frac{z_i}{z_i^*}\right) \end{bmatrix}^T, \quad (18)$$

where the current and desired Euclidean coordinates are related to the chosen feature point O_i . The rotation error signal $e_\omega(t) \in \mathbb{R}^3$ is typically chosen in terms of the reconstructed rotation parameterization as

$$e_\omega = u\theta.$$

The corresponding translation and rotation error system can be written as

$$\dot{e}_v = L_v v_c + L_{v\omega} \omega_c \quad (19)$$

$$\dot{e}_\omega = L_\omega \omega_c, \quad (20)$$

where $L_\omega(t)$ was defined in Eq. (16), and $L_v(t), L_{v\omega}(t) \in \mathbb{R}^{3 \times 3}$ are defined as

$$L_v = -\frac{\alpha_i}{z_i^*} \begin{bmatrix} 1 & 0 & -m_{xi} \\ 0 & 1 & -m_{yi} \\ 0 & 0 & 1 \end{bmatrix}, \quad (21)$$

$$L_{v\omega} = \begin{bmatrix} m_{xi}m_{yi} & -1 - m_{xi}^2 & m_{yi} \\ 1 + m_{yi}^2 & -m_{xi}m_{yi} & -m_{xi} \\ -m_{yi} & m_{xi} & 0 \end{bmatrix}.$$

The rotation and translation controllers can be designed as

$$\begin{aligned} \omega_c &= -kL_\omega^{-1} e_\omega = -ke_\omega \\ v_c &= -L_v^{-1} (ke_v - kL_{v\omega} e_\omega) \end{aligned} \quad (22)$$

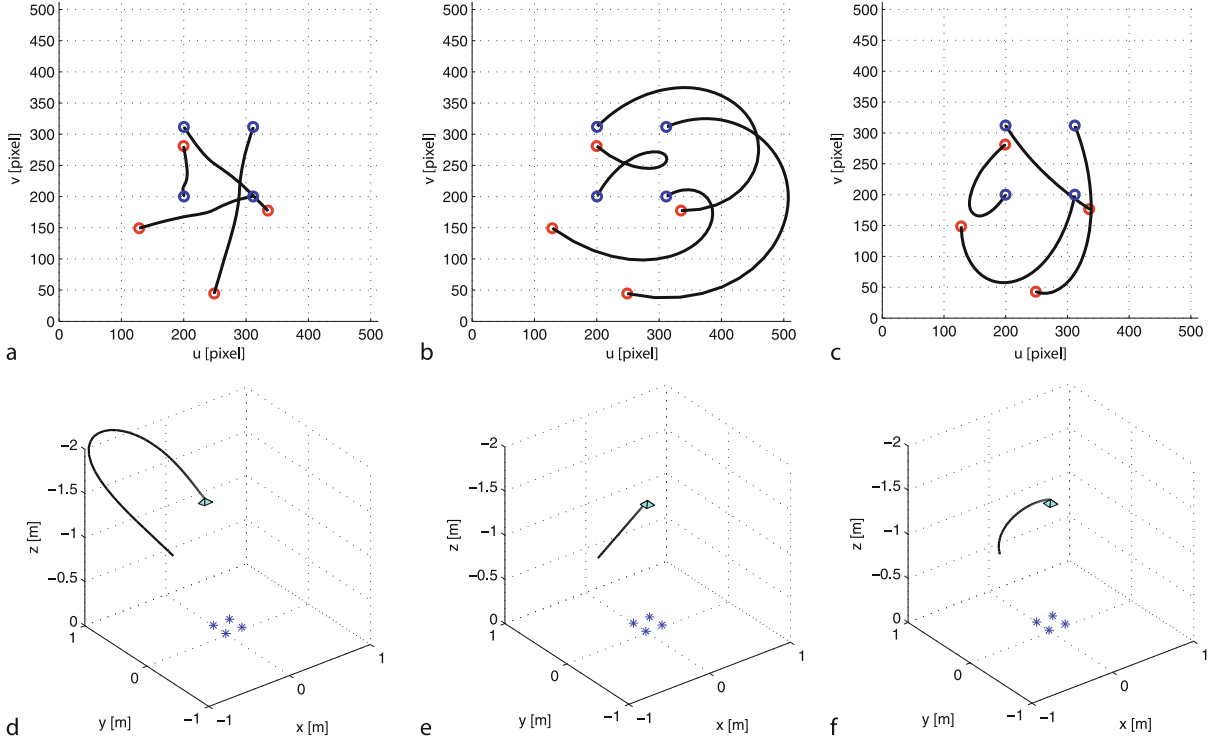
to yield an asymptotically stable result.

There are some other control approaches that combine image-space and reconstructed Euclidean information in the error system such as [8,23]. The partitioned visual servo control approach proposed in [8] decouples the z-axis motions (including both the translation and rotation components) from the other degrees of freedom and derives separate controllers for these z-axis motions. The switched visual servo control approach proposed in [23] partitions the control along the time axis, instead of the usual partitioning along spatial degrees of freedom. This system uses both position-based and image-based controllers, and a high-level decision maker invokes the appropriate controller based on the values of a set of Lyapunov functions.

Simulation Comparisons

The controllers given in Eqs. (13), (17), and (22) are implemented based on the same initial conditions. The errors asymptotically converge for all three controllers. The different image-space and Euclidean trajectories in the simulation are shown in Fig. 2.

Subplots a–c of Fig. 2 are two-dimensional image-space trajectories of the four feature points generated by the controllers Eqs. (13), (17), and (22), respectively. The trajectories start from the initial positions and asymptotically converge to the desired positions. A comparison of the results indicate that the image-based visual servo control and homography-based visual servo control have better performance in the image-space (see subplots a and c).



Adaptive Visual Servo Control, Figure 2

Simulation results for image-based, position-based, and combined visual servo control approaches

Also, since the position-based visual servo controller does not use image-features in the feedback, the feature points may leave the camera's field-of-view (see subplot b).

Subplots d–f of Fig. 2 are the three-dimensional Euclidean trajectories of the camera generated by the controllers Eqs. (13), (17), and (22), respectively. Since the position-based visual servo control uses the Euclidean information directly in the control loop, it has the best performance in the Euclidean trajectory (subplot e), followed by the homography-based controller (subplot f). The image-based visual servo controller in subplot d generates the longest Euclidean trajectory. Since image-based visual servo controller doesn't guarantee the Euclidean performance, it may generate a physically unimplementable trajectory.

Compensating for Projection Normalization

A key issue that impacts visual servo control is that the image-space is a two-dimensional projection of the three-dimensional Euclidean-space. To compensate for the lack of depth information from the two-dimensional image data, some researchers have focused on the use of alternate sensors (e. g., laser and sound ranging technologies).

Other researchers have explored the use of a camera-based vision system in conjunction with other sensors along with sensor fusion methods or the use of additional cameras in a stereo configuration that triangulate on corresponding images. However, the practical drawbacks of incorporating additional sensors include: increased cost, increased complexity, decreased reliability, and increased processing burden. Recently, some researchers have developed adaptive control methods that can actively adapt for the unknown depth parameter (cf. [19,20,21,22,24,25,26,27,28]) along with methods based on direct estimation and approximation (cf. [4,5]).

Adaptive Depth Compensation in Homography-Based Visual Servo Control

As in [19,20,21,22,26,27,28], homography-based visual servo controllers have been developed to achieve asymptotic regulation or tracking control of a manipulator end-effector or a mobile robot, by developing an adaptive update law that actively compensates for an unknown depth parameter. The open-loop error system and control development are discussed in this section.

Open-Loop Error System The translation error signal $e_v(t)$ is defined as in Subsect “Homography-Based Visual Servo Control”. However, the rotation error signal is defined as a unit quaternion $q(t) \triangleq [q_0(t) \ q_v^T(t)]^T \in \mathbb{R}^4$ where $q_0(t) \in \mathbb{R}$ and $q_v(t) \in \mathbb{R}^3$ (e.g., see [22,28]). The corresponding translation and rotation error systems are determined as

$$\dot{z}_i^* \dot{e}_v = -\alpha_i \tilde{L}_v v_c + z_i^* L_v \omega \omega_c \quad (23)$$

and

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_v \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -q_v^T \\ q_0 I_3 + q_v^\times \end{bmatrix} \omega_c, \quad (24)$$

where the notation $q_v^\times(t)$ denotes the skew-symmetric form of the vector $q_v(t)$ (e.g., see [22,28]), and $\tilde{L}_v(t) \in \mathbb{R}^{3 \times 3}$ is defined as

$$\tilde{L}_v = \begin{bmatrix} 1 & 0 & -m_{xi} \\ 0 & 1 & -m_{yi} \\ 0 & 0 & 1 \end{bmatrix}. \quad (25)$$

The structure of Eq. (25) indicates that $\tilde{L}_v(t)$ is measurable and can be used in the controller development directly.

Controller and Adaptive Law Design The rotation and translation controllers can be developed as (e.g., see [15,19,22,28])

$$\omega_c = -K_\omega q_v, \quad (26)$$

$$v_c = \frac{1}{\alpha_i} \tilde{L}_v^{-1} (K_v e_v + \dot{z}_i^* L_v \omega \omega_c), \quad (27)$$

where $K_\omega, K_v \in \mathbb{R}^{3 \times 3}$ denote two diagonal matrices of positive constant control gains. In Eq. (27), the parameter estimate $\dot{z}_i^*(t) \in \mathbb{R}$ is developed for the unknown constant z_i^* , and is defined as (e.g., see [19,22,28])

$$\dot{z}_i^* = \gamma_i e_v^T L_v \omega \omega_c, \quad (28)$$

where $\gamma_i \in \mathbb{R}$ denotes a positive constant adaptation gain. Based on Eqs. (23)–(28), the closed-loop error system is obtained as

$$\dot{q}_0 = \frac{1}{2} q_v^T K_\omega q_v, \quad (29)$$

$$\dot{q}_v = -\frac{1}{2} (q_0 I_3 + q_v^\times) K_\omega q_v, \quad (30)$$

$$\dot{z}_i^* \dot{e}_v = -K_v e_v + \tilde{z}_i L_v \omega \omega_c, \quad (31)$$

where $\tilde{z}_i(t) \in \mathbb{R}$ denotes the following parameter estimation error:

$$\tilde{z}_i = z_i^* - \hat{z}_i^*. \quad (32)$$

Closed-Loop Stability Analysis The controller given in Eqs. (26) and (27) along with the adaptive update law in Eq. (28) ensures asymptotic translation and rotation regulation in the sense that

$$\|q_v(t)\| \rightarrow 0 \quad \text{and} \quad \|e_v(t)\| \rightarrow 0 \quad \text{as} \quad t \rightarrow \infty. \quad (33)$$

To prove the stability result, a Lyapunov function candidate $V(t) \in \mathbb{R}$ can be defined as

$$V = \frac{z_i^*}{2} e_v^T e_v + q_v^T q_v + (1 - q_0)^2 + \frac{1}{2\gamma_i} \tilde{z}_i^2. \quad (34)$$

The time-derivative of $V(t)$ can be determined as

$$\begin{aligned} \dot{V} = & e_v^T (-K_v e_v + \tilde{z}_i L_v \omega \omega_c) - q_v^T (q_0 I_3 + q_v^\times) K_\omega q_v \\ & - (1 - q_0) q_v^T K_\omega q_v - \tilde{z}_i e_v^T L_v \omega \omega_c, \end{aligned} \quad (35)$$

where Eqs. (29)–(32) were utilized. The following negative semi-definite expression is obtained after simplifying Eq. (35):

$$\dot{V} = -e_v^T K_v e_v - q_v^T K_\omega q_v. \quad (36)$$

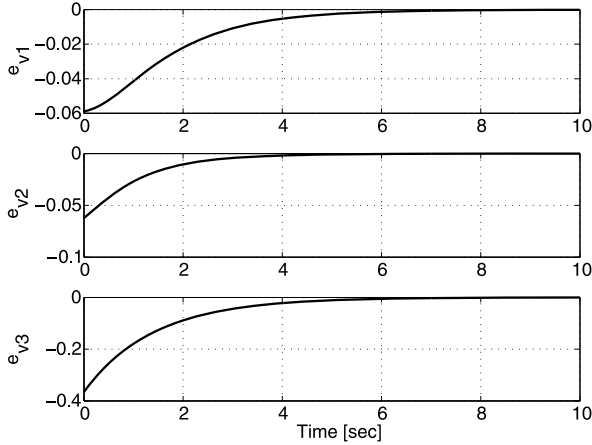
Barbalat’s Lemma [62] can then be used to prove the result given in Eq. (33).

Based on the controller in Eq. (26) and (27) and the adaptive law in Eq. (28), the resulting asymptotic translation and rotation errors are plotted in Fig. 3 and Fig. 4, respectively. The image-space trajectory is shown in Fig. 5, and also in Fig. 6 in a three-dimensional format, where the vertical axis is time. The parameter estimate for z_i^* is shown in Fig. 7.

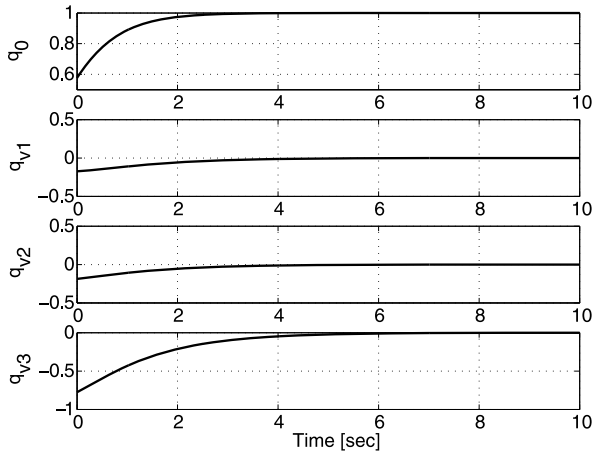
Adaptive Depth Compensation in Image-Based Visual Servo Control

This section provides an example of how an image-based visual servo controller can be developed that adaptively compensates for the uncertain depth. In this section, only three feature points are required to be tracked in the image to develop the error system (i.e., to obtain an image-Jacobian with at least equal rows than columns), but at least four feature points are required to solve for the unknown depth ratio $\alpha_i(t)$.

Open-Loop Error System The image error $e_i(t)$ defined in Subsect. “Image-based Visual Servo Control” can be



Adaptive Visual Servo Control, Figure 3

Unitless translation error $e_v(t)$ 

Adaptive Visual Servo Control, Figure 4

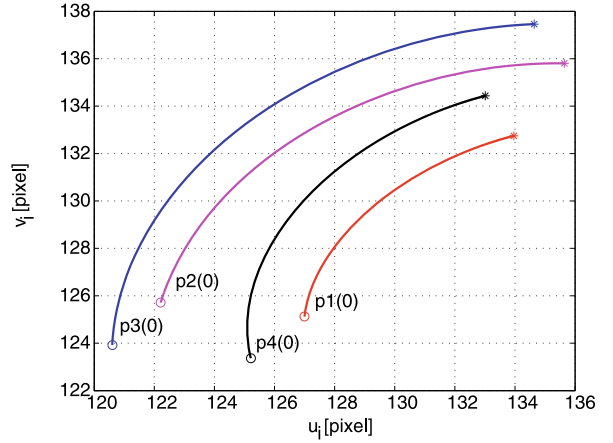
Quaternion rotation error $q(t)$

rewritten as

$$e_i(t) = \begin{bmatrix} m_{x1}(t) - m_{x1}^* \\ m_{y1}(t) - m_{y1}^* \\ m_{x2}(t) - m_{x2}^* \\ m_{y2}(t) - m_{y2}^* \\ m_{x3}(t) - m_{x3}^* \\ m_{y3}(t) - m_{y3}^* \end{bmatrix} \triangleq \begin{bmatrix} e_{x1} \\ e_{y1} \\ e_{x2} \\ e_{y2} \\ e_{x3} \\ e_{y3} \end{bmatrix}, \quad (37)$$

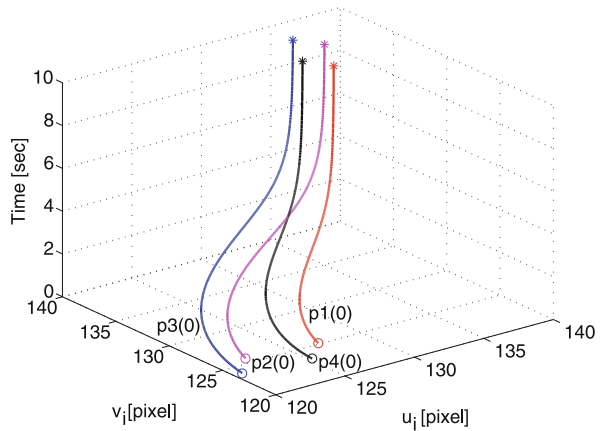
when three feature points are used for notation simplicity. The definition of image error $e_i(t)$ is expandable to more than three feature points. The time derivative of $e_i(t)$ is given by

$$\dot{e}_i(t) = L_i \begin{bmatrix} v_c \\ \omega_c \end{bmatrix}, \quad (38)$$



Adaptive Visual Servo Control, Figure 5

Image-space error in pixels between $p(t)$ and p^* . In the figure, "O" denotes the initial positions of the 4 feature points in the image, and "*" denotes the corresponding final positions of the feature points

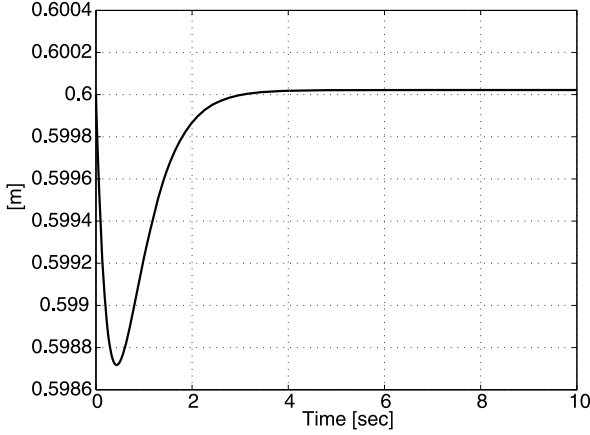


Adaptive Visual Servo Control, Figure 6

Image-space error in pixels between $p(t)$ and p^* shown in a 3D graph. In the figure, "O" denotes the initial positions of the 4 feature points in the image, and "*" denotes the corresponding final positions of the feature points

where the image Jacobian $L_i(t)$ is given by

$$L_i = \begin{bmatrix} \frac{\alpha_1}{z_1^*} & 0 & \frac{\alpha_1 m_{x1}}{z_1^*} & -m_{x1} m_{y1} & 1 + m_{x1}^2 & -m_{y1} \\ 0 & \frac{\alpha_1}{z_1^*} & \frac{\alpha_1 m_{y1}}{z_1^*} & -1 - m_{y1}^2 & m_{x1} m_{y1} & m_{x1} \\ \frac{\alpha_2}{z_2^*} & 0 & \frac{\alpha_2 m_{x2}}{z_2^*} & -m_{x2} m_{y2} & 1 + m_{x2}^2 & -m_{y2} \\ 0 & \frac{\alpha_2}{z_2^*} & \frac{\alpha_2 m_{y2}}{z_2^*} & -1 - m_{y2}^2 & m_{x2} m_{y2} & m_{x2} \\ \frac{\alpha_3}{z_3^*} & 0 & \frac{\alpha_3 m_{x3}}{z_3^*} & -m_{x3} m_{y3} & 1 + m_{x3}^2 & -m_{y3} \\ 0 & \frac{\alpha_3}{z_3^*} & \frac{\alpha_3 m_{y3}}{z_3^*} & -1 - m_{y3}^2 & m_{x3} m_{y3} & m_{x3} \end{bmatrix}, \quad (39)$$



Adaptive Visual Servo Control, Figure 7
Adaptive on-line estimate of z_i^*

where the depth ratio $\alpha_i(t)$, $i = 1, 2, 3$ is defined in (5). Let $\phi = [\phi_1 \ \phi_2 \ \phi_3]^T \in \mathbb{R}^3$ denote the unknown constant parameter vector as

$$\phi = \left[\frac{1}{z_1^*} \quad \frac{1}{z_2^*} \quad \frac{1}{z_3^*} \right]^T. \quad (40)$$

Even though the error system (38) is not linearly parameterizable in ϕ , an adaptive law can still be developed based on the following property in Paragraph “Controller and Adaptive Law Design”.

Controller and Adaptive Law Design Based on an open-loop error system as in Eq. (38), the controller is developed as

$$\begin{bmatrix} v_c \\ \omega_c \end{bmatrix} = -k_i \hat{L}_i^T e_i, \quad (41)$$

where $k_i \in \mathbb{R}$ is the control gain,

$$\hat{\phi}(t) = [\hat{\phi}_1(t) \ \hat{\phi}_2(t) \ \hat{\phi}_3(t)]^T \in \mathbb{R}^3$$

is the adaptive estimation of the unknown constant parameter vector defined in Eq. (40), and $\hat{L}_i(t) \in \mathbb{R}^{6 \times 6}$ is an estimation of the image Jacobian matrix as

$$\hat{L}_i = \begin{bmatrix} \alpha_1 \hat{\phi}_1 & 0 & \alpha_1 m_{x1} \hat{\phi}_1 & -m_{x1} m_{y1} & 1 + m_{x1}^2 & -m_{y1} \\ 0 & \alpha_1 \hat{\phi}_1 & \alpha_1 m_{y1} \hat{\phi}_1 & -1 - m_{y1}^2 & m_{x1} m_{y1} & m_{x1} \\ \alpha_2 \hat{\phi}_2 & 0 & \alpha_2 m_{x2} \hat{\phi}_2 & -m_{x2} m_{y2} & 1 + m_{x2}^2 & -m_{y2} \\ 0 & \alpha_2 \hat{\phi}_2 & \alpha_2 m_{y2} \hat{\phi}_2 & -1 - m_{y2}^2 & m_{x2} m_{y2} & m_{x2} \\ \alpha_3 \hat{\phi}_3 & 0 & \alpha_3 m_{x3} \hat{\phi}_3 & -m_{x3} m_{y3} & 1 + m_{x3}^2 & -m_{y3} \\ 0 & \alpha_3 \hat{\phi}_3 & \alpha_3 m_{y3} \hat{\phi}_3 & -1 - m_{y3}^2 & m_{x3} m_{y3} & m_{x3} \end{bmatrix}. \quad (42)$$

The adaptive law for $\hat{\phi}(t)$ in Eq. (42) is designed as

$$\dot{\hat{\phi}} = -\Gamma e_M \hat{L}_{iM} \hat{L}_i^T e_i, \quad (43)$$

where $\Gamma \in \mathbb{R}^{3 \times 3}$ is a positive diagonal matrix, $e_M(t) \in \mathbb{R}^{3 \times 6}$ and $\hat{L}_{iM}(t) \in \mathbb{R}^{6 \times 6}$ are defined as

$$e_M = \begin{bmatrix} e_{x1} & e_{y1} & 0 & 0 & 0 & 0 \\ 0 & 0 & e_{x2} & e_{y2} & 0 & 0 \\ 0 & 0 & 0 & 0 & e_{x3} & e_{y3} \end{bmatrix}$$

$$\hat{L}_{iM} = \begin{bmatrix} \alpha_1 & 0 & m_{x1} \alpha_1 & 0 & 0 & 0 \\ 0 & \alpha_1 & m_{y1} \alpha_1 & 0 & 0 & 0 \\ \alpha_2 & 0 & m_{x2} \alpha_2 & 0 & 0 & 0 \\ 0 & \alpha_2 & m_{y2} \alpha_2 & 0 & 0 & 0 \\ \alpha_3 & 0 & m_{x3} \alpha_3 & 0 & 0 & 0 \\ 0 & \alpha_3 & m_{y3} \alpha_3 & 0 & 0 & 0 \end{bmatrix}.$$

The adaptive law is motivated by the following two properties:

$$L_i - \hat{L}_i = \tilde{L}_i = \Phi_M \hat{L}_{iM}, \quad (44)$$

$$e_i^T \Phi_M \hat{L}_{iM} = \tilde{\phi}^T e_M \hat{L}_{iM}. \quad (45)$$

In Eqs. (44) and (45), the adaptive estimation error $\tilde{\phi}(t) \in \mathbb{R}^3$ and $\Phi_M(t) \in \mathbb{R}^{6 \times 6}$ are defined as

$$\tilde{\phi} = [\tilde{\phi}_1 \ \tilde{\phi}_2 \ \tilde{\phi}_3]^T$$

$$= [\phi_1 - \hat{\phi}_1 \ \phi_2 - \hat{\phi}_2 \ \phi_3 - \hat{\phi}_3]^T,$$

$$\Phi_M = \text{diag} \{ \tilde{\phi}_1, \tilde{\phi}_1, \tilde{\phi}_2, \tilde{\phi}_2, \tilde{\phi}_3, \tilde{\phi}_3 \}.$$

Under the controller Eq. (41) and the adaptive law Eq. (43), the closed-loop error system can be written as

$$\begin{aligned} \dot{e}_i &= -k_i L_i \hat{L}_i^T e_i = -k_i \hat{L}_i \hat{L}_i^T e_i - k_i \tilde{L}_i \hat{L}_i^T e_i \\ &= -k_i \hat{L}_i \hat{L}_i^T e_i - k_i \Phi_M \hat{L}_{iM} \hat{L}_i^T e_i. \end{aligned} \quad (46)$$

Closed-Loop Stability Analysis The controller given in Eq. (41) and the adaptive update law in Eq. (43) ensures asymptotic regulation in the sense that

$$\|e_i(t)\| \rightarrow 0 \quad \text{as } t \rightarrow \infty. \quad (47)$$

To prove the stability result, a Lyapunov function candidate $V(t) \in \mathbb{R}$ can be defined as

$$V = \frac{1}{2} e_i^T e_i + \frac{1}{2} k_i \tilde{\phi}^T \Gamma^{-1} \tilde{\phi}.$$

The time-derivative of $V(t)$ can be determined as

$$\begin{aligned} \dot{V} &= e_i^T \dot{e}_i - k_i \tilde{\phi}^T \Gamma^{-1} \dot{\tilde{\phi}} \\ &= e_i^T (-k_i \hat{L}_i \hat{L}_i^T e_i - k_i \Phi_M \hat{L}_{iM} \hat{L}_i^T e_i) \\ &\quad - k_i \tilde{\phi}^T \Gamma^{-1} (\Gamma e_M \hat{L}_{iM} e_i) \\ &= -k_i e_i^T \hat{L}_i \hat{L}_i^T e_i - k_i e_i^T \Phi_M \hat{L}_{iM} \hat{L}_i^T e_i + k_i \tilde{\phi}^T e_M \hat{L}_{iM} \hat{L}_i^T e_i \end{aligned}$$

Based on Eqs. (44) and (45)

$$e_i^T \Phi_M \hat{L}_{iM} \hat{L}_i^T = \tilde{\phi}^T e_M \hat{L}_{iM} \hat{L}_i^T,$$

and $\dot{V}(t)$ can be further simplified as

$$\begin{aligned} \dot{V} &= -k_i e_i^T \hat{L}_i \hat{L}_i^T e_i - k_i \tilde{\phi} e_M \hat{L}_{iM} \hat{L}_i^T e_i + k_i \tilde{\phi} e_M \hat{L}_{iM} \hat{L}_i^T e_i \\ &= -k_i e_i^T \hat{L}_i \hat{L}_i^T e_i. \end{aligned}$$

Barbalat's Lemma [62] can then be used to conclude the result given in Eq. (47).

Visual Servo Control via an Uncalibrated Camera

Motivated by the desire to incorporate robustness to camera calibration, different control approaches (e.g., Jacobian estimation-based least-square optimization, robust control, and adaptive control) that do not depend on exact camera calibration have been proposed (cf. [18,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,63]).

Jacobian Estimation Approach

The uncalibrated visual servo controllers that are based on a Jacobian matrix estimator have been developed in [36,37,38,39,40]. The camera calibration matrix and kinematic model is not required with the Jacobian matrix estimation. This type of visual servo control scheme is composed of a recursive Jacobian estimation law and a control law. Specifically, in [36], a weighted recursive least-square update law is used for Jacobian estimation, and a visual servo controller is developed via the Lyapunov approach. In [37], a Broyden Jacobian estimator is applied for Jacobian estimation and a nonlinear least-square optimization method is used for the visual servo controller development. In [39,40] a recursive least-squares algorithm is implemented for Jacobian estimation, and a dynamic Gauss–Newton method is used to minimize the squared error in the image plane to get the controller.

Let the robot end-effector features $y(\theta) \in \mathbb{R}^m$ be a function of the robot joint angle vector $\theta(t) \in \mathbb{R}^n$, and let the target features $y^*(t) \in \mathbb{R}^m$ be a function of time. The tracking error in the image-plane can then be defined as

$$f(\theta, t) = y(\theta) - y^*(t) \in \mathbb{R}^m.$$

The trajectory $\theta(t)$ that causes the end-effector to follow the target is established by minimizing the squared image error

$$F(\theta, t) = \frac{1}{2} f^T(\theta, t) f(\theta, t).$$

By using the dynamic Gauss–Newton method, the joint angle vector update law (controller) can be designed as [40]

$$\theta_{k+1} = \theta_k - (\hat{J}_k^T \hat{J}_k)^{-1} \hat{J}_k^T \left(f_k + \frac{\partial f_k}{\partial t} h_t \right), \quad (48)$$

where h_t is the increment of t . In Eq. (48), the estimated Jacobian $\hat{J}(t)$, can be determined by multiple recursive least-square Jacobian estimation algorithms (e.g., Broyden method [37], dynamic Broyden method [40], or an exponentially weighted recursive least-square algorithm [40]).

Robust Control Approach

Robust control approaches based on static best-guess estimation of the calibration matrix have been developed to solve the uncalibrated visual servo regulation problem (cf. [18,44,50,51]). Specifically, a kinematic controller was developed in [44] that utilizes a constant, best-guess estimate of the calibration parameters to achieve local set-point regulation for the six-DOF problem; although, several conditions on the rotation and calibration matrix are required. Model-free homography-based visual servoing methods using best-guess estimation are used in [18,50] to achieve exponential or asymptotic regulation with respect to both camera and hand-eye calibration errors for the six-DOF problem. A quaternion-based robust controller is developed in [51] that achieved a regulation control objective.

State Estimation Let $\hat{A} \in \mathbb{R}^{3 \times 3}$ be a constant best-guess estimate of the camera calibration matrix A . The normalized coordinate estimates, denoted by $\hat{m}_i(t)$, $\hat{m}_i^* \in \mathbb{R}^3$, can be expressed as

$$\hat{m}_i = \hat{A}^{-1} p_i = \tilde{A} m_i, \quad (49)$$

$$\hat{m}_i^* = \hat{A}^{-1} p_i^* = \tilde{A} m_i^*, \quad (50)$$

where the upper triangular calibration error matrix $\tilde{A} \in \mathbb{R}^{3 \times 3}$ is defined as

$$\tilde{A} = \hat{A}^{-1} A. \quad (51)$$

Since $m_i(t)$ and m_i^* can not be exactly determined, the estimates in Eqs. (49) and (50) can be substituted into Eq. (5) to obtain the following relationship:

$$\hat{m}_i = \alpha_i \hat{H} \hat{m}_i^*, \quad (52)$$

where $\hat{H}(t) \in \mathbb{R}^{3 \times 3}$ denotes the estimated Euclidean homography [18] defined as

$$\hat{H} = \tilde{A} \hat{H} \tilde{A}^{-1}. \quad (53)$$

Since $\hat{m}_i(t)$ and \hat{m}_i^* can be determined from Eqs. (49) and (50), a set of twelve linear equations can be developed from the four image point pairs, and Eq. (52) can be used to solve for $\hat{H}(t)$. If additional information (e. g., at least four vanishing points) is provided (see [64] for a description of how to determine vanishing points in an image), various techniques (e. g., see [58,59]) can be used to decompose $\hat{H}(t)$ to obtain the estimated rotation and translation components as

$$\hat{H} = \tilde{A}R\tilde{A}^{-1} + \tilde{A}x_h n^{*T} \tilde{A}^{-1} = \hat{R} + \hat{x}_h \hat{n}^{*T}, \quad (54)$$

where $\hat{R}(t) \in \mathbb{R}^{3 \times 3}$ and $\hat{x}_h(t), \hat{n}^* \in \mathbb{R}^3$ denote the estimate of $R(t)$, $x_h(t)$ and n^* , respectively, as [18]

$$\hat{R} = \tilde{A}R\tilde{A}^{-1}, \quad (55)$$

$$\hat{x}_h = \sigma \tilde{A}x_h, \quad (56)$$

$$\hat{n}^* = \frac{1}{\sigma} \tilde{A}^{-T} n^*, \quad (57)$$

where $\sigma \in \mathbb{R}$ denotes the positive constant

$$\sigma = \|\tilde{A}^{-T} n^*\|. \quad (58)$$

Based on Eq. (55), the rotation matrix estimate $\hat{R}(t)$ is similar to $R(t)$. By exploiting the properties of similar matrices (i. e., similar matrices have the same trace and eigenvalues), the following estimates can be determined as [18]

$$\hat{\theta} = \theta, \quad \hat{u} = \mu \tilde{A}u,$$

where $\hat{\theta}(t) \in \mathbb{R}$ and $\hat{u}(t) \in \mathbb{R}^3$ denote the estimates of $\theta(t)$ and $u(t)$, respectively, and $\mu \in \mathbb{R}$ is defined as

$$\mu = \frac{1}{\|\tilde{A}u\|}.$$

Alternatively, if the rotation matrix is decomposed as a unit quaternion. The quaternion estimate

$$\hat{q}(t) \triangleq [\hat{q}_0(t) \quad \hat{q}_v^T(t)]^T \in \mathbb{R}^4$$

can be related to the actual quaternion $q(t)$ as [51]

$$\hat{q}_0 = q_0, \quad \hat{q}_v = \pm \mu \tilde{A}q_v, \quad (59)$$

where

$$\mu = \frac{\|q_v\|}{\|\tilde{A}q_v\|}. \quad (60)$$

Since the sign ambiguity in Eq. (59) does not affect the control development and stability analysis [51], only the positive sign in Eq. (59) needs to be considered in the following control development and stability analysis.

Control Development and Stability Analysis The rotation open-loop error system can be developed by taking the time derivative of $q(t)$ as

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_v \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -q_v^T \\ q_0 I_3 + q_v^\times \end{bmatrix} \omega_c. \quad (61)$$

Based on the open-loop error system in (61) and the subsequent stability analysis, the angular velocity controller is designed as

$$\omega_c = -k_\omega \hat{q}_v, \quad (62)$$

where $k_\omega \in \mathbb{R}$ denotes a positive control gain. Substituting Eq. (62) into Eq. (61), the rotation closed-loop error system can be developed as

$$\dot{q}_0 = \frac{1}{2} k_\omega q_v^T \hat{q}_v \quad (63)$$

$$\dot{q}_v = -\frac{1}{2} k_\omega (q_0 I_3 + q_v^\times) \hat{q}_v. \quad (64)$$

The open-loop translation error system can be determined as [50]

$$\dot{e} = -\frac{1}{z_i^*} v_c - \omega_c^\times e + [m_i^*]^\times \omega_c, \quad (65)$$

where the translation error $e(t) \in \mathbb{R}^3$ is defined as

$$e = \frac{z_i}{z_i^*} m_i - m_i^*.$$

The translation controller is designed as (see also [18,50])

$$v_c = k_v \hat{e}, \quad (66)$$

where $k_v \in \mathbb{R}$ denotes a positive control gain, and $\hat{e}(t) \in \mathbb{R}^3$ is defined as

$$\hat{e} = \frac{z_i}{z_i^*} \hat{m}_i - \hat{m}_i^*. \quad (67)$$

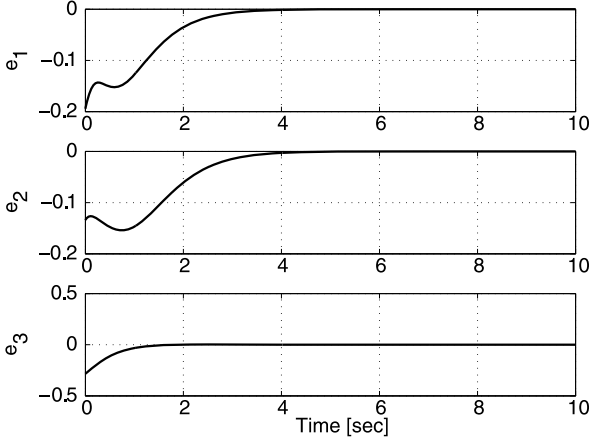
In Eq. (67), $\hat{m}_i(t)$ and \hat{m}_i^* can be computed from Eq. (49) and (50), respectively, and the ratio $(z_i(t))/z_i^*$ can be computed from the decomposition of the estimated Euclidean homography in Eq. (52).

After substituting Eqs. (62) and (66) into Eq. (65), the resulting closed-loop translation error system can be determined as

$$\dot{e} = \left(-k_v \frac{1}{z_i^*} \tilde{A} + [k_\omega \hat{q}_v]^\times \right) e - k_\omega [m_i^*]^\times \hat{q}_v. \quad (68)$$

The controller given in Eqs. (62) and (66) ensures asymptotic regulation in the sense that

$$\|q_v(t)\| \rightarrow 0, \quad \|e(t)\| \rightarrow 0 \quad \text{as } t \rightarrow \infty \quad (69)$$



Adaptive Visual Servo Control, Figure 8
Unitless translation error between $m_1(t)$ and m_1^*

provided k_v is selected sufficiently large, and the following inequalities are satisfied (cf. [18,50,51])

$$\lambda_{\min} \left\{ \frac{1}{2} (\tilde{A} + \tilde{A}^T) \right\} \geq \lambda_0, \quad \lambda_{\max} \left\{ \frac{1}{2} (\tilde{A} + \tilde{A}^T) \right\} \leq \lambda_1,$$

where $\lambda_0, \lambda_1 \in \mathbb{R}$ are positive constants, and $\lambda_{\min}\{\cdot\}$ and $\lambda_{\max}\{\cdot\}$ denote the minimal and maximal eigenvalues of $1/2(\tilde{A} + \tilde{A}^T)$, respectively.

To prove the stability result, a Lyapunov function candidate $V(t) \in \mathbb{R}$ can be defined as

$$V = q_v^T q_v + (1 - q_0)^2 + e^T e. \quad (70)$$

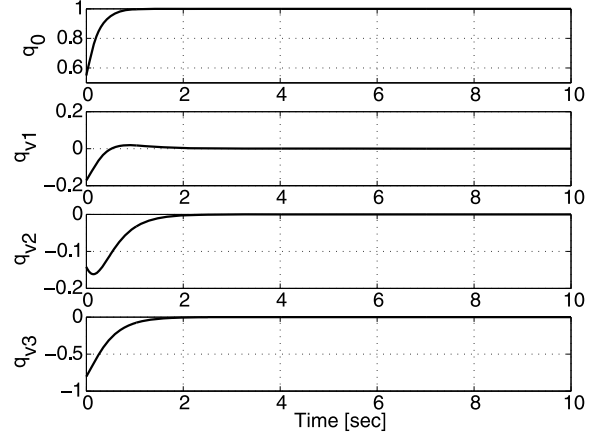
The time derivative of Eq. (70) can be proven to be negative semi-definite as [51]

$$\dot{V} \leq -\frac{k_\omega \lambda_0 \mu}{K_{v1}} \|q_v\|^2 - \frac{\lambda_0}{z_i^*} \|e\|^2, \quad (71)$$

where K_{v1} is a positive large enough number. Using the signal chasing argument in [51], Barbalat's Lemma [62] can then be used to prove the result given in Eq. (69). Based on the controller in Eqs. (62) and (66), the resulting asymptotic translation and rotation errors are plotted in Fig. 8 and Fig. 9, respectively.

Adaptive Image-Based Visual Servo Control Approach

In [49], an adaptive controller for image-based dynamic regulation control of a robot manipulator is proposed using a fixed camera whose intrinsic and extrinsic parameters are not known. A novel depth-independent interaction matrix (cf. [49,65,66]) is proposed therein to map the



Adaptive Visual Servo Control, Figure 9
Quaternion rotation error $q(t)$

visual signals onto the joints of the robot manipulator. Due to the depth-independent interaction matrix, the closed-loop dynamics can be linearly parameterized in the constant unknown camera parameters. An adaptive law can then be developed to estimate these parameters online.

Open-Loop Error System Denote the homogeneous coordinate of the feature point expressed in the robot base frame as

$$\bar{x}(t) = [x(t) \ y(t) \ z(t) \ 1]^T \in \mathbb{R}^4,$$

and denote its coordinate in the camera frame as

$${}^c \bar{x}(t) = [{}^c x(t) \ {}^c y(t) \ {}^c z(t)]^T \in \mathbb{R}^3.$$

The current and desired positions of the feature point on the image plane are respectively defined as

$$p(t) = [u(t) \ v(t) \ 1]^T \in \mathbb{R}^3 \quad \text{and}$$

$$p_d = [u_d \ v_d \ 1]^T \in \mathbb{R}^3.$$

The image error between the current and desired positions of the feature point on the image plane, denoted as $e_i(t) \in \mathbb{R}^3$, is defined as

$$e_i = p - p_d = [u \ v \ 1]^T - [u_d \ v_d \ 1]^T. \quad (72)$$

Under the camera perspective projection model, $p(t)$ can be related to $\bar{x}(t)$ as

$$p = \frac{1}{c_z} M \bar{x}(t), \quad (73)$$

where $M \in \mathbb{R}^{3 \times 4}$ is a constant perspective projection matrix that depends only on the intrinsic and extrinsic calibration parameters. Based on Eq. (73), the depth of the

feature point in the camera frame is given by

$${}^c z(t) = m_3^T \bar{x}(t), \quad (74)$$

where $m_3 \in \mathbb{R}^{1 \times 4}$ is the third row of M . Based on Eqs. (72)–(74), the derivative of image error $\dot{e}_i(t)$ is related to $\dot{\bar{x}}(t)$ as

$$\dot{e}_i = \dot{p} = \frac{1}{c_z} (M - p m_3^T) \dot{\bar{x}}. \quad (75)$$

The robot kinematics are given by

$$\dot{\bar{x}} = J(q) \dot{q}, \quad (76)$$

where $q(t) \in \mathbb{R}^n$ denotes the joint angle of the manipulator, and $J(q) \in \mathbb{R}^{4 \times n}$ denotes the manipulator Jacobian. Based on Eqs. (75) and (76),

$$\dot{e}_i = \frac{1}{c_z} (M - p m_3^T) J(q) \dot{q}. \quad (77)$$

The robot dynamics are given by

$$H(q) \ddot{q} + \left(\frac{1}{2} \dot{H}(q) + C(q, \dot{q}) \right) \dot{q} + g(q) = \tau. \quad (78)$$

In Eq. (78), $H(q) \in \mathbb{R}^{n \times n}$ is the positive-definite and symmetric inertia matrix, $C(q, \dot{q}) \in \mathbb{R}^{n \times n}$ is a skew-symmetric matrix, $g(q) \in \mathbb{R}^n$ denotes the gravitational force, and $\tau(t) \in \mathbb{R}^n$ is the joint input of the robot manipulator.

Controller and Adaptive Law Design As stated in Property 2 of [49], the perspective projection matrix M can only be determined up to a scale, so it is reasonable to fix one component of M and treat the others as unknown constant parameters. Without loss of generality, let m_{34} be fixed, and define the unknown parameter vector as

$$\phi = [m_{11} \ m_{12} \ m_{13} \ m_{14} \ m_{21} \ m_{22} \ m_{23} \ m_{24} \ m_{31} \ m_{32} \ m_{33}]^T \in \mathbb{R}^{11}.$$

To design the adaptive law, the estimated projection error of the feature point is defined as

$$\begin{aligned} e(t_j, t) &= p(t_j) \hat{m}_3^T(t) \bar{x}(t_j) - \hat{M}(t) \bar{x}(t_j) \\ &= W(\bar{x}(t_j), p(t_j)) \tilde{\phi} \end{aligned} \quad (79)$$

at time instant t_j . In Eq. (79), $W(\bar{x}(t_j), p(t_j)) \in \mathbb{R}^{3 \times 11}$ is a constant regression matrix and the estimation error $\tilde{\phi}(t) \in \mathbb{R}^{11}$ is defined as $\tilde{\phi}(t) = \phi - \hat{\phi}(t)$ with $\hat{\phi}(t) \in \mathbb{R}^{11}$ equal to the estimation of ϕ . The controller can then be designed as [49]

$$\tau = g(q) - K_1 \dot{q} - J^T(q) \left(\hat{M}^T - \hat{m}_3 p^T + \frac{1}{2} \hat{m}_3 e_i \right) B e_i, \quad (80)$$

where $K_1 \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{3 \times 3}$ are positive-definite gain

matrices. The adaptive law is designed as

$$\begin{aligned} \dot{\hat{\phi}} &= -\Gamma^{-1} \left(Y^T(q, p) \dot{q} \right. \\ &\quad \left. + \sum_{j=1}^m W^T(\bar{x}(t_j), p(t_j)) K_3 e(t_j, t) \right), \end{aligned} \quad (81)$$

where $\Gamma \in \mathbb{R}^{11 \times 11}$ and $K_3 \in \mathbb{R}^{3 \times 3}$ are positive-definite gain matrices, and the regression matrix $Y(q, p)$ is determined as

$$\begin{aligned} &-J^T(q) \left[(\hat{M}^T - \hat{m}_3 p^T - M^T + m_3 p^T) \right. \\ &\quad \left. + \frac{1}{2} (\hat{m}_3 - m_3) e_i^T \right] B e_i = Y(q, p) \tilde{\phi}. \end{aligned}$$

As stated in [49], if the five positions (corresponding to five time instant $t_j, j = 1, 2, \dots, 5$) of the feature point for the parameter adaptation are so selected that it is impossible to find three collinear projections on the image plane, under the control of the proposed controller Eq. (80) and the adaptive algorithm Eq. (81) for parameter estimation, then the image error of the feature point is convergent to zero, i. e., $\|e_i(t)\| \rightarrow 0$ as $t \rightarrow \infty$ (see [49] for the closed-loop stability analysis).

Adaptive Homography-Based Visual Servo Control Approach

Problem Statement and Assumptions In this section, a combined high-gain control approach and an adaptive control strategy are used to regulate the robot end-effector to a desired pose asymptotically. A homography-based visual servo control approach is used to address the six-DOF regulation problem. A high-gain controller is developed to asymptotically stabilize the rotation error system and the fact that the diagonal elements of the camera calibration matrix are positive. The translation error system can be linearly parameterized in terms of some unknown constant parameters which are determined by the unknown depth information and the camera calibration parameters. An adaptive controller is developed to stabilize the translation error by compensating for the unknown depth information and intrinsic camera calibration parameters. A Lyapunov-based analysis is used to examine the stability of the developed controller.

The following development is based on two mild assumptions.

Assumption 1: The bounds of a_{11} and a_{22} (defined in Subsect. “Geometric Relationships”) are assumed to be known as

$$\underline{\zeta}_{a_{11}} < a_{11} < \bar{\zeta}_{a_{11}}, \quad \underline{\zeta}_{a_{22}} < a_{22} < \bar{\zeta}_{a_{22}}. \quad (82)$$

The absolute values of a_{12} , a_{13} , and a_{23} are upper bounded as

$$|a_{12}| < \bar{\zeta}_{a_{12}}, \quad |a_{13}| < \bar{\zeta}_{a_{13}}, \quad |a_{23}| < \bar{\zeta}_{a_{23}}. \quad (83)$$

In (82) and (83), $\bar{\zeta}_{a_{11}}$, $\bar{\zeta}_{a_{11}}$, $\bar{\zeta}_{a_{22}}$, $\bar{\zeta}_{a_{22}}$, $\bar{\zeta}_{a_{12}}$, $\bar{\zeta}_{a_{13}}$ and $\bar{\zeta}_{a_{23}}$ are positive constants.

Assumption 2: The reference plane is within the camera field-of-view and not at infinity. That is, there exist positive constants $\bar{\zeta}_{z_i}$ and $\bar{\zeta}_{z_i}$ such that

$$\bar{\zeta}_{z_i} < z_i(t) < \bar{\zeta}_{z_i}. \quad (84)$$

Euclidean Reconstruction Using Vanishing Points Based on Eqs. (5)–(7), the homography relationship based on measurable pixel coordinates is

$$p_i = \alpha_i A H A^{-1} p_i^*. \quad (85)$$

Since A is unknown, standard homography computation and decomposition algorithms can't be applied to extract the rotation and translation from the homography. If some additional information can be applied (e. g., four vanishing points), the rotation matrix can be obtained. For the vanishing points, $d^* = \infty$, so that

$$H = R + \frac{x_f}{d^*} n^{*T} = R. \quad (86)$$

Based on Eq. (86), the relationship in Eq. (85) can be expressed as

$$p_i = \alpha_i \bar{R} p_i^*, \quad (87)$$

where $\bar{R}(t) \in \mathbb{R}^{3 \times 3}$ is defined as

$$\bar{R} = A R A^{-1}. \quad (88)$$

For the four vanishing points, twelve linear equations can be obtained based on Eq. (87). After normalizing $\bar{R}(t)$ by one nonzero element (e. g., $\bar{R}_{33}(t) \in \mathbb{R}$ which is assumed to be the third row third column element of $\bar{R}(t)$ without loss of generality) twelve equations can be used to solve for twelve unknowns. The twelve unknowns are given by the eight unknown elements of the normalized $\bar{R}(t)$, denoted by $\bar{R}_n(t) \in \mathbb{R}^{3 \times 3}$ defined as

$$\bar{R}_n \triangleq \frac{\bar{R}}{\bar{R}_{33}}, \quad (89)$$

and the four unknowns are given by $\bar{R}_{33}(t) \alpha_i(t)$. From the definition of $\bar{R}_n(t)$ in Eq. (89), the fact that

$$\det(\bar{R}) = \det(A) \det(R) \det(A^{-1}) = 1 \quad (90)$$

can be used to conclude that

$$\bar{R}_{33}^3 \det(\bar{R}_n) = 1, \quad (91)$$

and hence,

$$\bar{R} = \frac{\bar{R}_n}{\sqrt[3]{\det(\bar{R}_n)}}. \quad (92)$$

After $\bar{R}(t)$ is obtained, the original four feature points on the reference plane can be used to determine the depth ratio $\alpha_i(t)$.

Open-Loop Error System

Rotation Error System If the rotation matrix $R(t)$ introduced in (5) were known, then the corresponding unit quaternion $q(t) \triangleq [q_0(t) \quad q_v^T(t)]^T$ can be calculated using the numerically robust method presented in [28] and [67] based on the corresponding relationships

$$R(q) = (q_0^2 - q_v^T q_v) I_3 + 2q_v q_v^T - 2q_0 q_v^\times, \quad (93)$$

where I_3 is the 3×3 identity matrix, and the notation $q_v^\times(t)$ denotes the skew-symmetric form of the vector $q_v(t)$. Given $R(t)$, the quaternion $q(t)$ can also be written as

$$q_0 = \frac{1}{2} \sqrt{1 + \text{tr}(R)}, \quad q_v = \frac{1}{2} u \sqrt{3 - \text{tr}(R)}, \quad (94)$$

where $u(t) \in \mathbb{R}^3$ is a unit eigenvector of $R(t)$ with respect to the eigenvalue 1. The open-loop rotation error system for $q(t)$ can be obtained as [68]

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_v \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -q_v^T \\ q_0 I_3 + q_v^\times \end{bmatrix} \omega_c, \quad (95)$$

where $\omega_c(t) \in \mathbb{R}^3$ defines the angular velocity of the camera expressed in \mathcal{F} .

The quaternion $q(t)$ given in Eq. (93)–(95) is not measurable since $R(t)$ is unknown. However, since $\bar{R}(t)$ can be determined as described in Eq. (92), the same algorithm as shown in Eq. (94) can be used to determine a corresponding measurable quaternion $(\bar{q}_0(t), \bar{q}_v^T(t))^T$ as

$$\bar{q}_0 = \frac{1}{2} \sqrt{1 + \text{tr}(\bar{R})}, \quad \bar{q}_v = \frac{1}{2} \bar{u} \sqrt{3 - \text{tr}(\bar{R})}, \quad (96)$$

where $\bar{u}(t) \in \mathbb{R}^3$ is a unit eigenvector of $\bar{R}(t)$ with respect to the eigenvalue 1. Based on Eq. (88), $\text{tr}(\bar{R}) = \text{tr}(A R A^{-1}) = \text{tr}(R)$, where $\text{tr}(\cdot)$ denotes the trace of a matrix. Since $R(t)$ and $\bar{R}(t)$ are similar matrices, the relationship between $(q_0(t), q_v^T(t))^T$ and $(\bar{q}_0(t), \bar{q}_v^T(t))^T$ can be determined as

$$\bar{q}_0 = q_0, \quad \bar{q}_v = \frac{\|q_v\|}{\|A q_v\|} A q_v \triangleq \gamma A q_v, \quad (97)$$

where $\gamma(t) \in \mathbb{R}$ is a positive, unknown, time-varying scalar.

The square of $\gamma(t)$ is

$$\gamma^2 = \frac{q_v^T q_v}{(Aq_v)^T Aq_v} = \frac{q_v^T q_v}{q_v^T A^T A q_v}. \quad (98)$$

Since A is of full rank, the symmetric matrix $A^T A$ is positive definite. Hence, the Rayleigh–Ritz theorem can be used to conclude that

$$\lambda_{\min}(A^T A) q_v^T q_v \leq q_v^T A^T A q_v \leq \lambda_{\max}(A^T A) q_v^T q_v. \quad (99)$$

From Eqs. (98) and (99), it can be concluded that

$$\frac{1}{\lambda_{\max}(A^T A)} \leq \gamma^2 = \frac{q_v^T q_v}{(Aq_v)^T Aq_v} \leq \frac{1}{\lambda_{\min}(A^T A)}. \quad (100)$$

$$\sqrt{\frac{1}{\lambda_{\max}(A^T A)}} \leq \gamma \leq \sqrt{\frac{1}{\lambda_{\min}(A^T A)}}.$$

Based on Eq. (100), there exist positive bounding constants $\underline{\zeta}_\gamma, \bar{\zeta}_\gamma \in \mathbb{R}$ that satisfy the following inequalities:

$$\underline{\zeta}_\gamma < \gamma(t) < \bar{\zeta}_\gamma. \quad (101)$$

The inverse of the relationship between $\bar{q}_v(t)$ and $q_v(t)$ in Eq. (97) can also be developed as

$$q_v = \frac{1}{\gamma} A^{-1} \bar{q}_v$$

$$= \frac{1}{\gamma} \begin{bmatrix} \frac{1}{a_{11}} \bar{q}_{v1} - \frac{a_{12}}{a_{11} a_{22}} \bar{q}_{v2} - \left(\frac{a_{13}}{a_{11}} - \frac{a_{12} a_{23}}{a_{11} a_{22}} \right) \bar{q}_{v3} \\ \frac{1}{a_{22}} \bar{q}_{v2} - \frac{a_{23}}{a_{22}} \bar{q}_{v3} \\ \bar{q}_{v3} \end{bmatrix}. \quad (102)$$

Translation Error System The translation error, denoted by $e(t) \in \mathbb{R}^3$, is defined as

$$e(t) = p_e(t) - p_e^*, \quad (103)$$

where $p_e(t), p_e^* \in \mathbb{R}^3$ are defined as

$$p_e = [u_i \quad v_i \quad -\ln(\alpha_i)]^T, \quad p_e^* = [u_i^* \quad v_i^* \quad 0]^T, \quad (104)$$

where $i \in \{1, \dots, 4\}$. The translation error $e(t)$ is measurable since the first two elements are image coordinates, and $\alpha_i(t)$ is obtained from the homography decomposition. The open-loop translation error system can be obtained by taking the time derivative of $e(t)$ and multiplying the resulting expression by z_i^* as

$$z_i^* \dot{e} = -\alpha_i A_e v_c + z_i^* A_e (A^{-1} p_i)^x \omega_c, \quad (105)$$

where $v_c(t) \in \mathbb{R}^3$ defines the linear velocity of the camera expressed in \mathcal{F} , and $A_e(t) \in \mathbb{R}^{3 \times 3}$ is defined as

$$A_e = \begin{bmatrix} a_{11} & a_{12} & a_{13} - u_i \\ 0 & a_{22} & a_{23} - v_i \\ 0 & 0 & 1 \end{bmatrix}.$$

Rotation Controller Development and Stability Analysis Based on the relationship in Eq. (97), the open-loop error system in Eq. (95), and the subsequent stability analysis, the rotation controller is designed as

$$\begin{aligned} \omega_{c1} &= -k_{\omega 1} \bar{q}_{v1} = -(k_{\omega 11} + 2) \bar{q}_{v1} \\ \omega_{c2} &= -k_{\omega 2} \bar{q}_{v2} = -(k_{\omega 21} + k_{\omega 22} + 1) \bar{q}_{v2} \\ \omega_{c3} &= -k_{\omega 3} \bar{q}_{v3} = -(k_{\omega 31} + k_{\omega 32} + k_{\omega 33}) \bar{q}_{v3}, \end{aligned} \quad (106)$$

where $k_{\omega i} \in \mathbb{R}, i = 1, 2, 3$ and $k_{\omega ij} \in \mathbb{R}, i, j = 1, 2, 3, j < i$, are positive constants. The expressed form of the controller in Eq. (106) is motivated by the use of completing the squares in the subsequent stability analysis. In Eq. (106), the damping control gains $k_{\omega 21}, k_{\omega 31}, k_{\omega 32}$ are selected according to the following sufficient conditions to facilitate the subsequent stability analysis

$$\begin{aligned} k_{\omega 21} &> \frac{1}{4} k_{\omega 1}^2 \frac{\bar{\zeta}_{a_{12}}^2}{\underline{\zeta}_{a_{11}} \underline{\zeta}_{a_{22}}}, \\ k_{\omega 31} &> \frac{1}{4} k_{\omega 1}^2 \frac{1}{\underline{\zeta}_{a_{11}}} \left(\frac{\bar{\zeta}_{a_{12}} \bar{\zeta}_{a_{23}}}{\underline{\zeta}_{a_{22}}} + \bar{\zeta}_{a_{13}} \right)^2, \\ k_{\omega 32} &> \frac{1}{4} k_{\omega 2}^2 \frac{\bar{\zeta}_{a_{23}}^2}{\underline{\zeta}_{a_{22}}}, \end{aligned} \quad (107)$$

where $\underline{\zeta}_{a_{11}}, \bar{\zeta}_{a_{11}}, \underline{\zeta}_{a_{22}}, \bar{\zeta}_{a_{22}}, \bar{\zeta}_{a_{12}}, \bar{\zeta}_{a_{13}}$ and $\bar{\zeta}_{a_{23}}$ are defined in Eqs. (82) and (83), and $k_{\omega 11}, k_{\omega 22}, k_{\omega 33}$ are feedback gains that can be selected to adjust the performance of the rotation control system.

Provided the sufficient gain conditions given in Eq. (107) are satisfied, the controller in Eq. (106) ensures asymptotic regulation of the rotation error in the sense that

$$\|q_v(t)\| \rightarrow 0, \quad \text{as } t \rightarrow \infty. \quad (108)$$

To prove the stability result, a Lyapunov function candidate $V_1(t) \in \mathbb{R}$ can be defined as

$$V_1 \triangleq q_v^T q_v + (1 - q_0)^2. \quad (109)$$

Based on the open-loop error system in Eq. (95), the time-derivative of $V_1(t)$ can be determined as

$$\begin{aligned} \dot{V}_1 &= 2q_v^T \dot{q}_v - 2(1 - q_0) \dot{q}_0 \\ &= q_v^T \omega_c = q_{v1} \omega_{c1} + q_{v2} \omega_{c2} + q_{v3} \omega_{c3}. \end{aligned} \quad (110)$$

After substituting Eq. (102) for $q_v(t)$, substituting Eq. (106) for $\omega_c(t)$, and completing the squares, the expression in Eq. (110) can be written as

$$\dot{V}_1 < -\frac{1}{\xi_\gamma} \left(k_{\omega 11} \frac{1}{a_{11}} \tilde{q}_{v1}^2 + k_{\omega 22} \frac{1}{a_{22}} \tilde{q}_{v2}^2 + k_{\omega 33} \tilde{q}_{v3}^2 \right). \quad (111)$$

Barbalat's lemma [62] can now be used to conclude that $\|q_v(t)\| \rightarrow 0$ as $t \rightarrow \infty$.

Translation Control Development and Stability Analysis After some algebraic manipulation, the translation error system in Eq. (105) can be rewritten as

$$\begin{aligned} \frac{z_i^*}{a_{11}} \dot{e}_1 &= -\alpha_i v_{c1} + Y_1(\alpha_i, u_i, v_i, \omega_c, v_{c2}, v_{c3}) \phi_1 \\ \frac{z_i^*}{a_{22}} \dot{e}_2 &= -\alpha_i v_{c2} + Y_2(\alpha_i, u_i, v_i, \omega_c, v_{c3}) \phi_2 \\ z_i^* \dot{e}_3 &= -\alpha_i v_{c3} + Y_3(\alpha_i, u_i, v_i, \omega_c) \phi_3, \end{aligned} \quad (112)$$

where $\phi_1 \in \mathbb{R}^{n_1}$, $\phi_2 \in \mathbb{R}^{n_2}$, and $\phi_3 \in \mathbb{R}^{n_3}$ are vectors of constant unknown parameters, and $Y_1(\cdot) \in \mathbb{R}^{1 \times n_1}$, $Y_2(\cdot) \in \mathbb{R}^{1 \times n_2}$, and $Y_3(\cdot) \in \mathbb{R}^{1 \times n_3}$ are known regressor vectors.

The control strategy used in [52] is to design $v_{c3}(t)$ to stabilize $e_3(t)$, and then design $v_{c2}(t)$ to stabilize $e_2(t)$ given $v_{c3}(t)$, and then design $v_{c1}(t)$ to stabilize $e_1(t)$ given $v_{c3}(t)$ and $v_{c2}(t)$. Following this design strategy, the translation controller $v_c(t)$ is designed as [52]

$$\begin{aligned} v_{c3} &= \frac{1}{\alpha_i} (k_{v3} e_3 + Y_3(\alpha_i, u_i, v_i, \omega_c) \hat{\phi}_3) \\ v_{c2} &= \frac{1}{\alpha_i} (k_{v2} e_2 + Y_2(\alpha_i, u_i, v_i, \omega_c, v_{c3}) \hat{\phi}_2) \\ v_{c1} &= \frac{1}{\alpha_i} (k_{v1} e_1 + Y_1(\alpha_i, u_i, v_i, \omega_c, v_{c2}, v_{c3}) \hat{\phi}_1), \end{aligned} \quad (113)$$

where the depth ratio $\alpha_i(t) > 0 \forall t$. In (113), $\hat{\phi}_1(t) \in \mathbb{R}^{n_1}$, $\hat{\phi}_2(t) \in \mathbb{R}^{n_2}$, $\hat{\phi}_3(t) \in \mathbb{R}^{n_3}$ denote adaptive estimates that are designed according to the following adaptive update laws to cancel the respective terms in the subsequent stability analysis

$$\dot{\hat{\phi}}_1 = \Gamma_1 Y_1^T e_1, \quad \dot{\hat{\phi}}_2 = \Gamma_2 Y_2^T e_2, \quad \dot{\hat{\phi}}_3 = \Gamma_3 Y_3^T e_3, \quad (114)$$

where $\Gamma_1 \in \mathbb{R}^{n_1 \times n_1}$, $\Gamma_2 \in \mathbb{R}^{n_2 \times n_2}$, $\Gamma_3 \in \mathbb{R}^{n_3 \times n_3}$ are diagonal matrices of positive constant adaptation gains. Based on Eqs. (112) and (113), the closed-loop translation error

system is

$$\begin{aligned} \frac{z_i^*}{a_{11}} \dot{e}_1 &= -k_{v1} e_1 + Y_1(\alpha_i, u_i, v_i, \omega_c, v_{c2}, v_{c3}) \tilde{\phi}_1 \\ \frac{z_i^*}{a_{22}} \dot{e}_2 &= -k_{v2} e_2 + Y_2(\alpha_i, u_i, v_i, \omega_c, v_{c3}) \tilde{\phi}_2 \\ z_i^* \dot{e}_3 &= -k_{v3} e_3 + Y_3(\alpha_i, u_i, v_i, \omega_c) \tilde{\phi}_3, \end{aligned} \quad (115)$$

where $\tilde{\phi}_1(t) \in \mathbb{R}^{n_1}$, $\tilde{\phi}_2(t) \in \mathbb{R}^{n_2}$, $\tilde{\phi}_3(t) \in \mathbb{R}^{n_3}$ denote the intrinsic calibration parameter mismatch defined as

$$\begin{aligned} \tilde{\phi}_1(t) &= \phi_1 - \hat{\phi}_1(t), \quad \tilde{\phi}_2(t) = \phi_2 - \hat{\phi}_2(t), \\ \tilde{\phi}_3(t) &= \phi_3 - \hat{\phi}_3(t). \end{aligned}$$

The controller given in Eq. (113) along with the adaptive update law in Eq. (114) ensures asymptotic regulation of the translation error system in the sense that

$$\|e(t)\| \rightarrow 0 \quad \text{as } t \rightarrow \infty.$$

To prove the stability result, a Lyapunov function candidate $V_2(t) \in \mathbb{R}$ can be defined as

$$\begin{aligned} V_2 &= \frac{1}{2} \frac{z_i^*}{a_{11}} e_1^2 + \frac{1}{2} \frac{z_i^*}{a_{22}} e_2^2 + \frac{1}{2} z_i^* e_3^2 + \frac{1}{2} \tilde{\phi}_1^T \Gamma_1^{-1} \tilde{\phi}_1 \\ &\quad + \frac{1}{2} \tilde{\phi}_2^T \Gamma_2^{-1} \tilde{\phi}_2 + \frac{1}{2} \tilde{\phi}_3^T \Gamma_3^{-1} \tilde{\phi}_3. \end{aligned} \quad (116)$$

After taking the time derivative of Eq. (116) and substituting for the closed-loop error system developed in Eq. (115), the following simplified expression can be obtained:

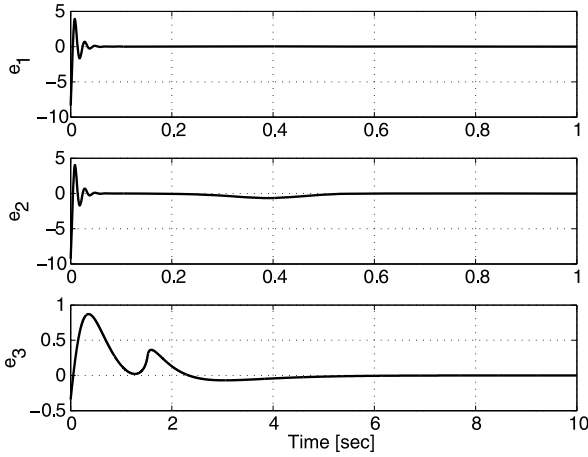
$$\dot{V}_2 = -k_{v1} e_1^2 - k_{v2} e_2^2 - k_{v3} e_3^2. \quad (117)$$

Barbalat's lemma [62] can now be used to show that $e_1(t)$, $e_2(t)$, $e_3(t) \rightarrow 0$ as $t \rightarrow \infty$.

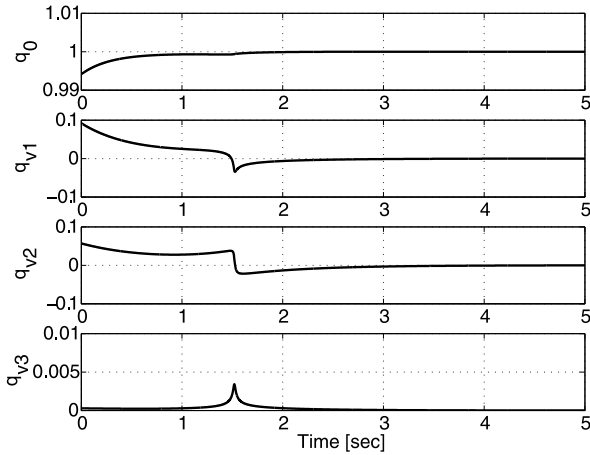
The controller given in Eqs. (106) and (113) along with the adaptive update law in Eq. (114) ensures asymptotic translation and rotation regulation in the sense that

$$\|q_v(t)\| \rightarrow 0 \quad \text{and} \quad \|e(t)\| \rightarrow 0 \quad \text{as } t \rightarrow \infty,$$

provided the control gains satisfy the sufficient conditions given in Eq. (107). Based on the controller in Eqs. (106) and (113) and the adaptive law in Eq. (114), the resulting asymptotic translation and rotation errors are plotted in Figs. 10 and 11, respectively. The image-space trajectory is shown in Fig. 12, and it is also shown in Fig. 13 in a three dimensional format.



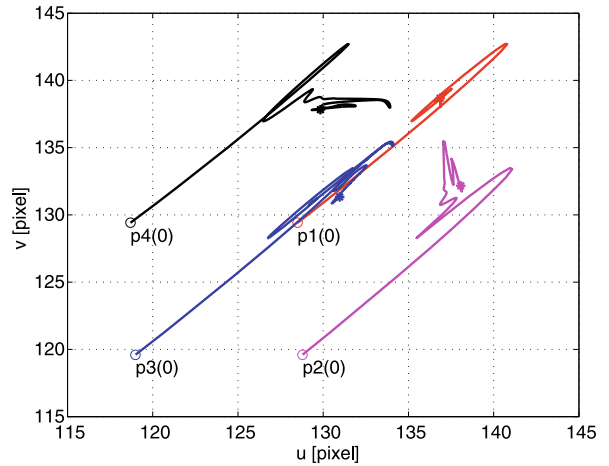
Adaptive Visual Servo Control, Figure 10
Unitless translation error $e(t)$



Adaptive Visual Servo Control, Figure 11
Quaternion rotation error $q(t)$

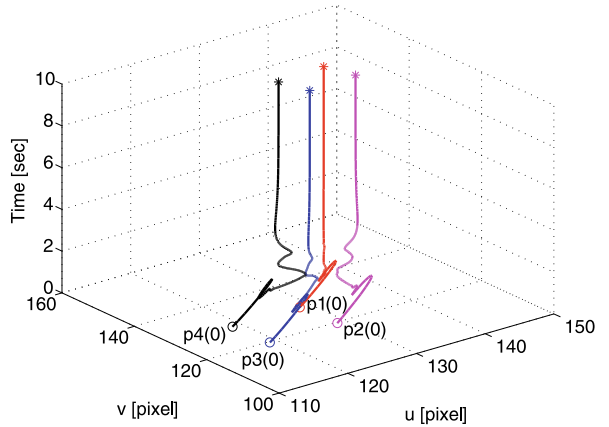
Future Directions

Image-based visual servo control is problematic because it may yield impossible Euclidean trajectories for the robotic system, and the image-Jacobian can contain singularities. Position-based visual servo control also has deficits in that the feedback signal does not exploit the image information directly, so the resulting image-trajectory may take the feature points out of the field-of-view. Homography-based visual servo control combines the positive aspects of the former methods; however, this approach requires the homography construction and decomposition. The homography construction is sensitive to image noise and there exists a sign ambiguity in the homography decomposition. Open problems that are areas of future research include



Adaptive Visual Servo Control, Figure 12

Image-space error in pixels between $p(t)$ and p^* . In the figure, "O" denotes the initial positions of the 4 feature points in the image, and "*" denotes the corresponding final positions of the feature points



Adaptive Visual Servo Control, Figure 13

Image-space error in pixels between $p(t)$ and p^* shown in a 3D graph. In the figure, "O" denotes the initial positions of the 4 feature points in the image, and "*" denotes the corresponding final positions of the feature points

new methods to combine image-based and reconstructed information in the feedback loop without the homography decomposition, perhaps using methods that directly servo on the homography feedback without requiring the homography decomposition.

The success of visual servo control for traditional robotic systems provides an impetus for other autonomous systems such as the navigation and control of unmanned air vehicles (UAV). For new applications, new hurdles may arise. For example, the inclusion of the dynamics in traditional robotics applications is a straight-

forward task through the use of integrator backstepping. However, when a UAV is maneuvering (including roll, pitch, yaw motions) to a desired pose, the feature points may be required to leave the field-of-view because of the unique flight dynamics. Other examples include systems that are hyper-redundant, such as robotic elephant trunks, where advanced path-planning must be incorporated with the controller to account for obstacle avoidance and maintaining a line-of-sight to a target.

Bibliography

- Weiss LE, Sanderson AC, Neuman CP (1987) Dynamic sensor-based control of robots with visual feedback. *IEEE J Robot Autom* RA-3:404–417
- Feddema J, Mitchell O (1989) Vision-guided servoing with feature-based trajectory generation. *IEEE Trans Robot Autom* 5:691–700
- Hashimoto K, Kimoto T, Ebine T, Kimura H (1991) Manipulator control with image-based visual servo. *Proc IEEE Int Conf Robot Autom*, pp 2267–2272
- Espiau B, Chaumette F, Rives P (1992) A new approach to visual servoing in robotics. *IEEE Trans Robot Autom* 8:313–326
- Hutchinson S, Hager G, Corke P (1996) A tutorial on visual servo control. *IEEE Trans Robot Autom* 12:651–670
- Chaumette F (1998) Potential problems of stability and convergence in image-based and position-based visual servoing. In: Kriegman D, Hager G, Morse A (eds), *The confluence of vision and control LNCIS series*, vol 237. Springer, Berlin, pp 66–78
- Espiau B (1993) Effect of camera calibration errors on visual servoing in robotics. 3rd Int Symp. Exp Robot, pp 182–192
- Corke P, Hutchinson S (2001) A new partitioned approach to image-based visual servo control. *IEEE Trans Robot Autom* 17:507–515
- Wilson WJ, Hulls CW, Bell GS (1996) Relative end-effector control using cartesian position based visual servoing. *IEEE Trans Robot Autom* 12:684–696
- Martinet P, Gallice J, Khadraoui D (1996) Vision based control law using 3D visual features. *Proc World Autom Cong*, pp 497–502
- Daucher N, Dhome M, Lapresté J, Rives G (1997) Speed command of a robotic system by monocular pose estimate. *Proc Int Conf Intell Robot Syst*, pp 55–62
- Hashimoto K (2003) A review on vision-based control of robot manipulators. *Adv Robot* 17:969–991
- Chaumette F, Hutchinson S (2006) Visual servo control part i: Basic approaches. *IEEE Robot Autom Mag* 13:82–90
- Deguchi K (1998) Optimal motion control for image-based visual servoing by decoupling translation and rotation. *Proc Int Conf Intell Robot Syst*, pp 705–711
- Malis E, Chaumette F, Bodet S (1999) 2 1/2D visual servoing. *IEEE Trans Robot Autom* 15:238–250
- Malis E, Chaumette F (2000) 2 1/2D visual servoing with respect to unknown objects through a new estimation scheme of camera displacement. *Int J Comput Vis* 37:79–97
- Chaumette F, Malis E (2000) 2 1/2D visual servoing: a possible solution to improve image-based and position-based visual servoings. *Proc IEEE Int Conf Robot Autom*, pp 630–635
- Malis E, Chaumette F (2002) Theoretical improvements in the stability analysis of a new class of model-free visual servoing methods. *IEEE Trans Robot Autom* 18:176–186
- Chen J, Dawson DM, Dixon WE, Behal A (2005) Adaptive homography-based visual servo tracking for a fixed camera configuration with a camera-in-hand extension. *IEEE Trans Control Syst Technol* 13:814–825
- Fang Y, Dixon WE, Dawson DM, Chawda P (2005) Homography-based visual servoing of wheeled mobile robots. *IEEE Trans Syst Man Cybern – Part B: Cybern* 35:1041–1050
- Chen J, Dixon WE, Dawson DM, McIntyre M (2006) Homography-based visual servo tracking control of a wheeled mobile robot. *IEEE Trans Robot* 22:406–415
- Hu G, Gupta S, Fitz-coy N, Dixon WE (2006) Lyapunov-based visual servo tracking control via a quaternion formulation. *Proc IEEE Conf Decision Contr*, pp 3861–3866
- Gans N, Hutchinson S (2007) Stable visual servoing through hybrid switched-system control. *IEEE Trans Robot* 23:530–540
- Conticelli F, Allotta B (2001) Nonlinear controllability and stability analysis of adaptive image-based systems. *IEEE Trans Robot Autom* 17:208–214
- Conticelli F, Allotta B (2001) Discrete-time robot visual feedback in 3-D positioning tasks with depth adaptation. *IEEE/ASME Trans Mechatron* 6:356–363
- Fang Y, Behal A, Dixon WE, Dawson DM (2002) Adaptive 2.5D visual servoing of kinematically redundant robot manipulators. *Proc IEEE Conf Decision Contr*, pp 2860–2865
- Fang Y, Dawson DM, Dixon WE, de Queiroz MS (2002) 2.5D visual servoing of wheeled mobile robots. *Proc IEEE Conf Decision Contr*, pp 2866–2871
- Hu G, Dixon WE, Gupta S, Fitz-coy N (2006) A quaternion formulation for homography-based visual servo control. *Proc IEEE Int Conf Robot Autom*, pp 2391–2396
- Tsai RY (1987) A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE J Robot Autom* 3:323–344
- Tsai RY (1989) *Synopsis of recent progress on camera calibration for 3D machine vision*. MIT Press, Cambridge
- Robert L (1996) Camera calibration without feature extraction. *Comput Vis Image Underst* 63:314–325
- Heikkilä J, Silven O (1997) A four-step camera calibration procedure with implicit image correction. *Proc IEEE Conf Comput Vis Pattern Recogn*, pp 1106–1112
- Clarke TA, Fryer JG (1998) The development of camera calibration methods and models. *Photogrammetric Rec* 16:51–66
- Sturm PF, Maybank SJ (1999) On plane-based camera calibration: A general algorithm, singularities, applications. *Proc IEEE Conf Comput Vis Pattern Recogn*, pp 432–437
- Zhang Z (1999) Flexible camera calibration by viewing a plane from unknown orientations. *Proc IEEE Int Conf Comput Vis*, pp 666–673
- Hosoda K, Asada M (1994) Versatile visual servoing without knowledge of true jacobian. *Proc IEEE/RSJ Int Conf Intell Robot Syst*, pp 186–193
- Jagersand M, Fuentes O, Nelson R (1997) Experimental evaluation of uncalibrated visual servoing for precision manipulation. *Proc Int Conf Robot Autom*, pp 2874–2880
- Shahamiri M, Jagersand M (2005) Uncalibrated visual servoing using a biased newton method for on-line singularity detection and avoidance. *Proc IEEE/RSJ Int Conf Intell Robot Syst*, pp 3953–3958

39. Piepmeier JA, Lipkin H (2003) Uncalibrated eye-in-hand visual servoing. *Int J Robot Res* 22:805–819
40. Piepmeier JA, McMurray GV, Lipkin H (2004) Uncalibrated dynamic visual servoing. *IEEE Trans Robot Autom* 24:143–147
41. Kelly R (1996) Robust asymptotically stable visual servoing of planar manipulator. *IEEE Trans Robot Autom* 12:759–766
42. Bishop B, Spong MW (1997) Adaptive calibration and control of 2D monocular visual servo system. *Proc IFAC Symp Robot Contr*, pp 525–530
43. Hsu L, Aquino PLS (1999) Adaptive visual tracking with uncertain manipulator dynamics and uncalibrated camera. *Proc IEEE Conf Decision Contr*, pp 1248–1253
44. Taylor CJ, Ostrowski JP (2000) Robust vision-based pose control. *Proc IEEE Int Conf Robot Autom*, pp 2734–2740
45. Zergeroglu E, Dawson DM, de Queiroz M, Behal A (2001) Vision-based nonlinear tracking controllers in the presence of parametric uncertainty. *IEEE/ASME Trans Mechatron* 6:322–337
46. Dixon WE, Dawson DM, Zergeroglu E, Behal A (2001) Adaptive tracking control of a wheeled mobile robot via an uncalibrated camera system. *IEEE Trans Syst Man Cybern – Part B: Cybern* 31:341–352
47. Astolfi A, Hsu L, Netto M, Ortega R (2002) Two solutions to the adaptive visual servoing problem. *IEEE Trans Robot Autom* 18:387–392
48. Ruf A, Tonko M, Horaud R, Nagel H-H (1997) Visual tracking of an end-effector by adaptive kinematic prediction. *Proc IEEE/RSJ Int Conf Intell Robot Syst*, pp 893–898
49. Liu Y, Wang H, Wang C, Lam K (2006) Uncalibrated visual servoing of robots using a depth-independent interaction matrix. *IEEE Trans Robot* 22:804–817
50. Fang Y, Dixon WE, Dawson DM, Chen J (2003) An exponential class of model-free visual servoing controllers in the presence of uncertain camera calibration. *Proc IEEE Conf Decision Contr*, pp 5390–5395
51. Hu G, Gans N, Dixon WE (2007) Quaternion-based visual servo control in the presence of camera calibration error. *Proc IEEE Multi-Conf Syst Contr*, pp 1492–1497
52. Chen J, Behal A, Dawson DM, Dixon WE (2003) Adaptive visual servoing in the presence of intrinsic calibration uncertainty. *Proc IEEE Conf Decision Contr*, pp 5396–5401
53. Faugeras O (1993) Three-dimensional computer vision: A geometric viewpoint. MIT Press, Cambridge
54. Shi J, Tomasi C (1994) Good features to track. *Proc IEEE Conf Comput Vis Pattern Recogn*, pp 593–600
55. Tomasi C, Kanade T (1991) Detection and tracking of point features. Technical report. Carnegie Mellon University, Pittsburgh
56. Boufama B, Mohr R (1995) Epipole and fundamental matrix estimation using virtual parallax. *Proc IEEE Int Conf Comput Vis*, pp (1030)–1036
57. Hartley R, Zisserman A (2000) Multiple view geometry in computer vision. Cambridge University Press, New York
58. Faugeras O, Lustman F (1988) Motion and structure from motion in a piecewise planar environment. *Int J Pattern Recogn Artif Intell* 2:485–508
59. Zhang Z, Hanson AR (1995) Scaled euclidean 3D reconstruction based on externally uncalibrated cameras. *IEEE Symp Comput Vis*, pp 37–42
60. DeMenthon D, Davis L (1995) Model-based object pose in 25 lines of code. *Int J Comput Vis* 15:123–141
61. Quan L, Lan Z-D (1999) Linear n -point camera pose determination. *IEEE Trans Pattern Anal Machine Intell* 21:774–780
62. Slotine JJ, Li W (1991) Applied nonlinear control. Prentice Hall, Inc., Englewood Cliff
63. Zachi A, Hsu L, Ortega R, Lizarralde F (2006) Dynamic control of uncertain manipulators through immersion and invariance adaptive visual servoing. *Int J Robot Res* 25:1149–1159
64. Almansa A, Desolneux A, Vamech S (2003) Vanishing point detection without any a priori information. *IEEE Trans Pattern Analysis and Machine Intelligence* 25:502–507
65. Liu Y, Wang H, Lam K (2005) Dynamic visual servoing of robots in uncalibrated environments. *Proc IEEE Int Conf Robot Autom*, pp 3142–3148
66. Liu Y, Wang H, Zhou D (2006) Dynamic tracking of manipulators using visual feedback from an uncalibrated fixed camera. *Proc IEEE Int Conf Robot Autom*, pp 4124–4129
67. Shuster M (1993) A survey of attitude representations. *J Astron Sci* 41:439–518
68. Dixon WE, Behal A, Dawson DM, Nagarkatti S (2003) Nonlinear control of engineering systems: A Lyapunov-based approach. Birkhäuser, Boston

Additive Cellular Automata

BURTON VOORHEES

Center for Science, Athabasca University,
Athabasca, Canada

Article Outline

[Glossary](#)

[Definition of the Subject](#)

[Introduction](#)

[Notation and Formal Definitions](#)

[Additive Cellular Automata in One Dimension](#)

[d-Dimensional Rules](#)

[Future Directions](#)

[Bibliography](#)

Glossary

Cellular automata Cellular automata are dynamical systems that are discrete in space, time, and value. A state of a cellular automaton is a spatial array of discrete cells, each containing a value chosen from a finite alphabet. The state space for a cellular automaton is the set of all such configurations.

Alphabet of a cellular automaton The alphabet of a cellular automaton is the set of symbols or values that can appear in each cell. The alphabet contains a distinguished symbol called the null or quiescent symbol, usually indicated by 0, which satisfies the condition of an additive identity: $0 + x = x$.

Cellular automata rule The rule, or update rule of a cellular automaton describes how any given state is transformed into its successor state. The update rule of a cellular automaton is described by a rule table, which defines a local neighborhood mapping, or equivalently as a global update mapping.

Additive cellular automata An additive cellular automaton is a cellular automaton whose update rule satisfies the condition that its action on the sum of two states is equal to the sum of its actions on the two states separately.

Linear cellular automata A linear cellular automaton is a cellular automaton whose update rule satisfies the condition that its action on the sum of two states separately equals action on the sum of the two states plus its action on the state in which all cells contain the quiescent symbol. Note that some researchers reverse the definitions of additivity and linearity.

Neighborhood The neighborhood of a given cell is the set of cells that contribute to the update of value in that cell under the specified update rule.

Rule table The rule table of a cellular automaton is a listing of all neighborhoods together with the symbol that each neighborhood maps to under the local update rule.

Local maps of a cellular automaton The local mapping for a cellular automaton is a map from the set of all neighborhoods of a cell to the automaton alphabet.

State transition diagram The state transition diagram (STD) of a cellular automaton is a directed graph with each vertex labeled by a possible state and an edge directed from a vertex x to a vertex y if and only if the state labeling vertex x maps to the state labeling vertex y under application of the automaton update rule.

Transient states A transient state of a cellular automaton is a state that can at most appear only once in the evolution of the automaton rule.

Cyclic states A cyclic state of a cellular automaton is a state lying on a cycle of the automaton update rule, hence it is periodically revisited in the evolution of the rule.

Basins of attraction The basins of attraction of a cellular automaton are the equivalences classes of cyclic states together with their associated transient states, with two states being equivalent if they lie on the same cycle of the update rule.

Predecessor state A state x is the predecessor of a state y if and only if x maps to y under application of the cellular automaton update rule. More specifically, a state x is an n th order predecessor of a state y if it maps to y under n applications of the update rule.

Garden-of-Eden A Garden-of-Eden state is a state that has no predecessor. It can be present only as an initial condition.

Surjectivity A mapping is surjective (or onto) if every state has a predecessor.

Injectivity A mapping is injective (one-to-one) if every state in its domain maps to a unique state in its range. That is, if states x and y both map to a state z then $x = y$.

Reversibility A mapping X is reversible if and only if a second mapping X^{-1} exists such that if $X(x) = y$ then $X^{-1}(y) = x$. For finite state spaces reversibility and injectivity are identical.

Definition of the Subject

Cellular automata are discrete dynamical systems in which an extended array of symbols from a finite alphabet is iteratively updated according to a specified local rule. Originally developed by John von Neumann [1,2] in 1948, following suggestions from Stanislaw Ulam, for the purpose of showing that self-replicating automata could be constructed. Von Neumann's construction followed a complicated set of reproduction rules but later work showed that self-reproducing automata could be constructed with only simple update rules, e.g. [3]. More generally, cellular automata are of interest because they show that highly complex patterns can arise from the application of very simple update rules. While conceptually simple, they provide a robust modeling class for application in a variety of disciplines, e.g. [4], as well as fertile grounds for theoretical research. Additive cellular automata are the simplest class of cellular automata. They have been extensively studied from both theoretical and practical perspectives.

Introduction

A wide variety of cellular automata applications, in a number of differing disciplines, has appeared in the past fifty years, see, e.g. [5,6,7,8]. Among other things, cellular automata have been used to model growth and aggregation processes [9,10,11,12]; discrete reaction-diffusion systems [13,14,15,16,17]; spin exchange systems [18,19]; biological pattern formation [20,21]; disease processes and transmission [22,23,24,25,26]; DNA sequences, and gene interactions [27,28]; spiral galaxies [29]; social interaction networks [30]; and forest fires [31,32]. They have been used for language and pattern recognition [33,34,35,36,37,38,39]; image processing [40,41]; as parallel computers [42,43,44,45,46,47]; parallel multipliers [48]; sorters [49]; and prime number sieves [50].

In recent years, cellular automata have become important for VLSI logic circuit design [51]. Circuit designers need “simple, regular, modular, and cascable logic circuit structure to realize a complex function” and cellular automata, which show a significant advantage over linear feedback shift registers, the traditional circuit building block, satisfy this need (see [52] for an extensive survey). Cellular automata, in particular additive cellular automata, are of value for producing high-quality pseudorandom sequences [53,54,55,56,57]; for pseudoexhaustive and deterministic pattern generation [58,59,60,61,62,63]; for signature analysis [64,65,66]; error correcting codes [67,68]; pseudoassociative memory [69]; and cryptography [70].

In this discussion, attention focuses on the subclass of additive cellular automata. These are the simplest cellular automata, characterized by the property that the action of the update rule on the sum of two states is equal to the sum of the rule acting on each state separately. Hybrid additive rules (i. e., with different cells evolving according to different additive rules) have proved particularly useful for generation of pseudorandom and pseudoexhaustive sequences, signature analysis, and other circuit design applications, e. g. [52,71,72].

The remainder of this article is organized as follows: Sect. “Notation and Formal Definitions” introduces definitions and notational conventions. In Sect. “Additive Cellular Automata in One Dimension”, consideration is restricted to one-dimensional rules. The influence of boundary conditions on the evolution of one-dimensional rules, conditions for rule additivity, generation of fractal space-time outputs, equivalent forms of rule representation, injectivity and reversibility, transient lengths, and cycle periods are discussed using several approaches. Taking X as the global operator for an additive cellular automata, a method for analytic solution of equations of the form $X(\mu) = \beta$ is described. Section “ d -Dimensional Rules” describes work on d -dimensional rules defined on tori. The discrete baker transformation is defined and used to generalize one-dimensional results on transient lengths, cycle periods, and similarity of state transition diagrams. Extensive references to the literature are provided throughout and a set of general references is provided at the end of the bibliography.

Notation and Formal Definitions

Let $S(\mathcal{L}) = \{s_i\}$ be the set of lattice sites of a d -dimensional lattice \mathcal{L} with n_r equal to the number of lattice sites on dimension r . Denote by \mathcal{A} a finite symbols set with $|\mathcal{A}| = p$ (usually prime). An \mathcal{A} -configuration on \mathcal{L} is a surjective map $\nu : \mathcal{A} \mapsto S(\mathcal{L})$ that assigns a symbol from \mathcal{A} to each

site in $S(\mathcal{L})$. In this way, every \mathcal{A} -configuration defines a size $n_1 \times \cdots \times n_d$, d -dimensional matrix μ of symbols drawn from \mathcal{A} . Denote the set of all \mathcal{A} -configurations on \mathcal{L} by $\mathcal{E}(\mathcal{A}, \mathcal{L})$.

Each $s_i \in S(\mathcal{L})$ is labeled by an integer vector $\vec{i} = (i_1, \dots, i_d)$ where i_r is the number of sites along the r th dimension separating s_i from the assigned origin in \mathcal{L} . The shift operator on the r th dimension of \mathcal{L} is the map $\sigma_r : \mathcal{L} \mapsto \mathcal{L}$ defined by

$$\sigma_r(s_i) = s_j, \quad \vec{j} = (i_1, \dots, i_r - 1, \dots, i_d). \quad (1)$$

Equivalently, the shift maps the value at site \vec{i} to the value at site \vec{j} .

Let $\mu(s_i; t) = \mu(i_1, \dots, i_d; t) \in \mathcal{A}$ be the entry of μ corresponding to site s_i at iteration t for any discrete dynamical system having $\mathcal{E}(\mathcal{A}, \mathcal{L})$ as state space. Given a finite set of integer d -tuples $\mathcal{N} = \{(k_1, \dots, k_d)\}$, define the \mathcal{N} -neighborhood of a site $s_i \in S(\mathcal{L})$ as

$$N(s_i) = \{s_j \mid \vec{j} = \vec{i} + \vec{k}, \vec{k} \in \mathcal{N}\} \quad (2)$$

A neighborhood configuration is a surjective map $\nu : \mathcal{A} \mapsto N(s_0)$. Denote the set of all neighborhood configurations by $\mathcal{E}_{\mathcal{N}}(\mathcal{A})$.

The rule table for a cellular automata acting on the state space $\mathcal{E}(\mathcal{A}, \mathcal{L})$ with standard neighborhood $N(s_0)$ is defined by a map $x : \mathcal{E}_{\mathcal{N}}(\mathcal{A}) \mapsto \mathcal{A}$ (note that this map need not be surjective or injective). The value of x for a given neighborhood configuration is called the (value of the) rule component of that configuration. The map $x : \mathcal{E}_{\mathcal{N}}(\mathcal{A}) \mapsto \mathcal{A}$ induces a global map $X : \mathcal{E}(\mathcal{A}, \mathcal{L}) \mapsto \mathcal{E}(\mathcal{A}, \mathcal{L})$ as follows: For any given element $\mu(t) \in \mathcal{E}(\mathcal{A}, \mathcal{L})$, the set $C(s_i) = \{\mu(s_j; t) \mid s_j \in N(s_i)\}$ is a neighborhood configuration for the site s_i , hence the map $\mu(s_i; t) \mapsto x(C(s_i))$ for all s_i produces a new symbol $\mu(s_i; t + 1)$. The site s_i is called the mapping site. When taken over all mapping sites, this produces a matrix $\mu(t + 1)$ that is the representation of $X(\mu(t))$. A cellular automaton is indicated by reference to its rule table or to the global map defined by this rule table.

A cellular automaton with global map X is additive if and only if, for all pairs of states μ and β ,

$$X(\mu + \beta) = X(\mu) + X(\beta) \quad (3)$$

Addition of states is carried out site-wise mod(p) on the matrix representations of μ and β ; for example, for a one-dimensional six-site lattice with $p = 3$ the sum of 120112 and 021212 is 111021.

The definition for additivity given in [52] differs slightly from this standard definition. There, a binary valued cellular automaton is called “linear” if its local rule

only involves the XOR operation and “additive” if it involves XOR and/or XNOR. A rule involving XNOR can be written as the binary complement of a rule involving only XOR. In terms of the global operator of the rule, this means that it has the form $\mathbf{1} + X$ where X satisfies Eq. (3) and $\mathbf{1}$ represents the rule that maps every site to 1. Thus, $(\mathbf{1} + X)(\mu + \beta)$ equals $\mathbf{1} \dots \mathbf{1} + X(\mu + \beta)$ while

$$\begin{aligned} (\mathbf{1} + X)(\mu) + (\mathbf{1} + X)(\beta) \\ = \mathbf{1} \dots \mathbf{1} + \mathbf{1} \dots \mathbf{1} + X(\mu) + X(\beta) \\ = X(\mu) + X(\beta) \bmod(2). \end{aligned}$$

In what follows, an additive rule is defined strictly as one obeying Eq. (3), corresponding to rules that are “linear” in [52].

Much of the formal study of cellular automata has focused on the properties and forms of representation of the map $X : \mathcal{E}(\mathcal{A}, \mathcal{L}) \mapsto \mathcal{E}(\mathcal{A}, \mathcal{L})$. The structure of the state transition diagram (STD(X)) of this map is of particular interest.

Example 1 (Continuous Transformations of the Shift Dynamical System) Let \mathcal{L} be isomorphic to the set of integers \mathbb{Z} . Then $\mathcal{E}(\mathcal{A}, \mathbb{Z})$ is the set of infinite sequences with entries from \mathcal{A} . With the product topology induced by the discrete topology on \mathcal{A} and σ as the left shift map, the system $(\mathcal{E}(\mathcal{A}, \mathbb{Z}), \sigma)$ is the shift dynamical system on \mathcal{A} . The set of cellular automata maps $X : \mathcal{E}(\mathcal{A}, \mathbb{Z}) \mapsto \mathcal{E}(\mathcal{A}, \mathbb{Z})$ constitutes the class of continuous shift-commuting transformations of $(\mathcal{E}(\mathcal{A}, \mathbb{Z}), \sigma)$, a fundamental result of Hedlund [73].

Example 2 (Elementary Cellular Automata) Let \mathcal{L} be isomorphic to \mathbb{Z} with $\mathcal{A} = \{0, 1\}$ and $\mathcal{N} = \{-1, 0, 1\}$. The neighborhood of site s_i is $\{s_{i-1}, s_i, s_{i+1}\}$ and $\mathcal{E}_{\mathcal{N}}(\mathcal{A}) = \{000, 001, 010, 011, 100, 101, 110, 111\}$. In this one-dimensional case, the rule table can be written as $x_i = x(i_0 i_1 i_2)$ where $i_0 i_1 i_2$ is the binary form of the index i . Listing this gives the standard form for the rule table of an elementary cellular automata.

000	001	010	011	100	101	110	111
x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7

The standard labeling scheme for elementary cellular automata was introduced by Wolfram [74], who observed that the rule table for elementary rules defines the binary number $\sum_{i=0}^7 x_i 2^i$ and used this number to label the corresponding rule.

Example 3 (The Game of Life) This simple 2-dimensional cellular automata was invented by John Conway to illustrate a self-reproducing system. It was first presented in

1970 by Martin Gardner [75,76]. The game takes place on a square lattice, either infinite or toridal. The neighborhood of a cell consists of the eight cells surrounding it. The alphabet is $\{0,1\}$: a 1 in a cell indicates that cell is alive, a 0 indicates that it is dead. The update rules are: (a) If a cell contains a 0, it remains 0 unless exactly three of its neighbors contain a 1; (b) If a cell contains a 1 then it remains a 1 if and only if two or three of its neighbors are 1. This cellular automata produces a number of interesting patterns including a variety of fixed points (still life); oscillators (period 2); and moving patterns (gliders, spaceships); as well as more exotic patterns such as glider guns which generate a stream of glider patterns.

Additive Cellular Automata in One Dimension

Much of the work on cellular automata has focused on rules in one dimension ($d = 1$). This section reviews some of this work.

Boundary Conditions and Additivity

In the case of one-dimensional cellular automata, the lattice \mathcal{L} can be isomorphic to the integers; to the non-negative integers; to the finite set $\{0, \dots, n-1\} \in \mathbb{Z}$; or to the integers modulo an integer n . In the first case, there are no boundary conditions; in the remaining three cases, different boundary conditions apply. If \mathcal{L} is isomorphic to \mathbb{Z}_n , the integers mod(n), the boundary conditions are periodic and the lattice is circular (it is a p -adic necklace). This is called a cylindrical cellular automata [77] because evolution of the rule can be represented as taking place on a cylinder. If the lattice is isomorphic to $\{0, \dots, n-1\}$, null, or Dirichlet boundary conditions are set [78,79,80]. That is, the symbol assigned to all sites in \mathcal{L} outside of this set is the null symbol. When the lattice is isomorphic to the non-negative integers \mathbb{Z}^+ , null boundary conditions are set at the left boundary. In these latter two cases, the neighborhood structure assumed may influence the need for null conditions.

Example 4 (Elementary Rule 90) Let δ represent the global map for the elementary cellular automata rule 90, with rule table

000	001	010	011	100	101	110	111
0	1	0	1	1	0	1	0

For a binary string μ in \mathbb{Z} or \mathbb{Z}_n the action of rule 90 is defined by $[\delta(\mu)]_i = \mu_{i-1} + \mu_{i+1} \bmod(2)$, where all indices are taken mod(n) in the case of \mathbb{Z}_n . In the remaining

cases,

$$[\delta(\mu)]_i = \begin{cases} \mu_1 & i = 0 \\ \mu_{i-1} + \mu_{i+1} & 0 < i < n-1 \\ \mu_{n-2} & i = n-1 \\ \text{null conditions} & \end{cases} \quad (4)$$

$$[\delta(\mu)]_i = \begin{cases} \mu_1 & i = 0 \\ \mu_{i-1} + \mu_{i+1} & 0 < i \\ \text{half-infinite conditions} & \end{cases}$$

Note that \mathcal{L} and \mathcal{Z}_n are representations of the intervals $[-1, 1]$ and $[0, 1]$ respectively. Cellular automata rules are not quite functions on these intervals, however, since they are generally double valued on rational points having distinct representations as binary strings [81]. For example, both $0\bar{1}$ and $1\bar{0}$ in \mathcal{Z}^+ , where underlining indicates infinite repetition, are numerically $1/2$ but $\delta(0\bar{1}) = 1\bar{1}\bar{0} = 3/4$ while $\delta(1\bar{0}) = 0\bar{1}\bar{0} = 1/4$.

The state space $\mathcal{E}(\{0, 1\}, \mathcal{L})$ for binary valued one-dimensional cellular automata is just the set of binary sequences over the specified one-dimensional lattice. For the cases of \mathcal{Z} and \mathcal{Z}_n all such rules commute with the shift operator σ . When null boundary conditions are involved, however, commutativity fails at the boundary sites. For example, let X be the global operator for an elementary cellular automata operating on strings $\mu = \mu_0 \dots \mu_{n-1}$ of length n with null boundary conditions. Noting that $-1 = 1 \bmod(2)$, the commutator $[X, \sigma]$ has components

$$[[X, \sigma](\mu)]_i = [X\sigma(\mu) + \sigma X(\mu)]_i$$

$$= \begin{cases} X(0\mu_1\mu_2) + X(\mu_0\mu_1\mu_2) \bmod(2) & i = 0 \\ 0 & 0 < i < n-1 \\ X(\mu_{n-1}00) & i = n-1 \end{cases} \quad (5)$$

The relation $[X, \sigma] = \bar{0}$ can be preserved for rules defined on \mathcal{Z}^+ by altering the neighborhood structure. The case $\mathcal{N} = \{-1, 0, 1\}$ is called nearest neighbor because the value introduced at site i at the next iteration of a rule is determined by the values at site i and its immediately neighboring sites $i-1$ and $i+1$. Taking $\mathcal{N} = \{0, 1, 2\}$ yields left justified neighborhoods. This eliminates the need for null boundary conditions at the left boundary in \mathcal{Z}^+ .

Changes in the neighborhood structure of this sort are equivalent to changes in the mapping site. It is important to recognize that such changes can significantly alter the topological structure of the state transition diagram (STD) for a rule. If X represents the global map for a rule with

nearest neighbor neighborhoods, then the global map for the same rule with left justified neighborhoods is σX and the presence of the shift operator can change cycle periods. For example, the maximum cycle period for nearest neighbor rule 90 acting on \mathcal{Z}_6 is 2 while the maximum cycle period for this same rule with left justified neighborhoods is 3.

$100010 \mapsto 010100 \mapsto 100010$ nearest neighbor case

$100010 \mapsto 101000 \mapsto 001010 \mapsto 100010$ left justified case

The additivity condition of Eq. (3) has an expression in terms of rule table components. For $\mathcal{A} = \{0, 1\}$ this is given by:

Theorem 1 ([81])

A k -site rule $X : \mathcal{E}(\{0, 1\}, \mathcal{L}) \mapsto \mathcal{E}(\{0, 1\}, \mathcal{L})$ with rule components x_i is additive if and only if for $i = i_0 \dots i_{k-1}$ and $j = j_0 \dots j_{k-1}$, with $(i+j)_r = i_r + j_r \bmod(2)$, it is true that $x_i + x_j = x_{i+j} \bmod(2)$.

Corollary 1 ([81])

A k -site rule $X : \mathcal{E}(\{0, 1\}, \mathcal{L}) \mapsto \mathcal{E}(\{0, 1\}, \mathcal{L})$ is additive if and only if for all $i = i_0 \dots i_{k-1}$

$$x_i = \sum_{r=0}^{k-1} i_{k-r-1} x_{2^r} \bmod(2). \quad (6)$$

By Eq. (6), the k rule components x_{2^r} , $0 \leq r \leq k-1$, determine the set of k -site additive rules. Hence only 2^k of the possible 2^{2^k} k -site rules are additive, including the $\bar{0}$ rule that maps all sites to 0.

Additive Cellular Automata and Fractals

There is a direct connection between the space-time output patterns of additive cellular automata and self-similar fractal patterns [82,83,84,85,86,87,88]. The simplest examples are elementary rules 102 and 90. When acting on a doubly infinite sequence with the initial state $\bar{0}\bar{1}\bar{0}$, iteration of these rules yields the space-time output indicated in Fig. 1. In the case of rule 60, this output is the $\bmod(2)$ Pascal triangle while for rule 90 it consists of every other row of this triangle [89].

The pattern generated by rule 60 (or, equivalently, by rule 102) rescales to yield the fractal known as the Sierpinski gasket [90,91]. Direct connections between cellular automata outputs and the fractal generation schemes of matrix substitution systems and hierarchical iterated function systems are shown in [92,93,94,95,96,97,98,99]. In [100,101] the dimension spectrum associated to the

```

1
11
101
1111
10001
110011
1010101
11111111
10000001
110000011
1010000101
111100001111
1000100010001
11001100110011
101010101010101
111111111111111
1000000000000001
11000000000000011
101000000000000101
1111000000000001111
10001000000000010001
110011000000000110011
1010101000000001010101
11111111000000011111111
10000001000000010000001
110000011000000110000011
1010000101000001010000101
1111000011110000111100001111
10001000100010001000100010001
110011001100110011001100110011
10101010101010101010101010101
11111111111111111111111111111

```

Rule 60 Space-Time Output
(32 Iterations)

```

1
101
10001
110011
11111111
110000011
111100001111
11001100110011
111111111111111
11000000000000011
1111000000000001111
110011000000000110011
11111111000000011111111
1100000110000001100000011
1111000011110000111100001111
110011001100110011001100110011
1111000011110000111100001111
110011001100110011001100110011
11111111111111111111111111111

```

Rule 90 Space-Time Output
(16 Iterations)

Additive Cellular Automata, Figure 1

Space-time output of rules 60 and 90 from initial state with a single 1

space-time output of additive cellular automata is shown to be equal to the singularity spectrum of an associated multifractal.

Forms of Representation

Unless otherwise noted, the lattice in this section will be \mathbb{Z}_n with periodic boundary conditions. Several forms of representation for additive rules with periodic boundary conditions appear in the literature. Rules have been represented as dipolynomials over finite fields [102]; as recursion relations [103,104]; as circulant matrices [105,106]; and as polynomials in roots of unity or in powers of the left shift [81].

The global operator for a k -site additive rule with neighborhood structure $\mathcal{N} = \{-r, \dots, k-r-1\}$ can be written as

$$\mathcal{X} = \sigma^{-r} \sum_{s=0}^{k-1} a_s \sigma^s \quad a_s \in \mathcal{A}. \quad (7)$$

In the dipolynomial representation, a state $\mu = \mu_0 \dots \mu_{n-1}$ defines a dipolynomial $\mu(x)$ while the rule \mathcal{X} is represented by the dipolynomial equivalent of Eq. (7):

represented by the dipolynomial equivalent of Eq. (7):

$$\mu(x) = \sum_{s=0}^{n-1} \mu_s x^s, \quad a_s \in \mathcal{A} \quad (8)$$

$$\mathcal{X} = x^{-r} \sum_{s=0}^{k-1} a_s x^s, \quad a_s \in \mathcal{A}$$

The action of \mathcal{X} on a state μ is obtained by multiplication of the corresponding dipolynomials: $\mathcal{X}(x)\mu(x)$, with all products reduced mod(n) [102].

The rule \mathcal{X} is also representable in terms of powers of the shift operator, with the expression

$$\mathcal{X} = \sum_{s=0}^{k-1} a_s \sigma^s \quad a_s \in \mathcal{A} \quad (9)$$

where the coefficients in this equation differ from those in Eq. (7). For example, rule 90 acting on \mathbb{Z}_n is represented by $\sigma^{-1} + \sigma$ in the form of Eq. (7), and by $\sigma + \sigma^5$ in the form of Eq. (9) because $-1 = 5 \bmod(6)$.

When a rule \mathcal{X} acting on \mathbb{Z}_n is represented as in Eq. (9) the string (a_0, \dots, a_{n-1}) is directly connected to the representation of \mathcal{X} as a circulant matrix.

A right circulant matrix is a matrix in which each successive row is obtained from the row immediately above by shifting that row one unit to the right, with the final row entry shifted to the front. Thus,

$$\text{circ}(a_0, a_1, a_2, \dots, a_{n-2}, a_{n-1}) = \begin{pmatrix} a_0 & a_1 & a_2 & \cdots & a_{n-2} & a_{n-1} \\ a_{n-1} & a_0 & a_1 & \cdots & a_{n-3} & a_{n-2} \\ & & & \ddots & & \\ a_1 & a_2 & a_3 & \cdots & a_{n-1} & a_0 \end{pmatrix}$$

If $\mu \in \mathcal{E}(\mathcal{A}, \mathbb{Z}_n)$ is written as a column vector, the operation of \mathcal{X} on μ is given as multiplication by the right circulant matrix $\text{circ}(a_0, \dots, a_{n-1})$ with all terms reduced $\text{mod}(p)$. The value of this representation is that properties of circulant matrices are well known and this provides significant information about the cellular automata rule. This approach has a natural extension to the case of null boundary conditions [107,108], although the matrix involved is no longer a complete circulant. For example, the nearest neighbor form of rule 90 acting on \mathbb{Z}_6 with periodic boundary conditions has matrix representation

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \quad (10a)$$

while the matrix representation for null boundary conditions is

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \quad (10b)$$

The circulant representation of the left shift σ on \mathbb{Z}_n is

$$\sigma = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ & & & \ddots & & & & \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \end{pmatrix} \quad (10c)$$

The following lemmas [109] show the direct connection to Eq. (9) and to Hedlund's condition:

Lemma 1 ([109]) *An $n \times n$ matrix \mathbf{A} is circulant if and only if $[\mathbf{A}, \sigma] = 0$.*

Lemma 2 ([109]) *An $n \times n$ matrix \mathbf{A} is circulant if and only if $\mathbf{A} = P_A(\sigma)$ for some polynomial P_A of degree less than or equal to n . Further, if $\mathbf{A} = \text{circ}(a_0, \dots, a_{n-1})$ then*

$$P_A(\sigma) = \sum_{s=0}^{n-1} a_s \sigma^s \quad (11)$$

Other properties of circulant matrices provide links to the representation of rules over \mathbb{Z}_n by polynomials in the n th roots of unity. Let $\omega = e^{2\pi i/n}$ be the first complex n th root of unity.

Definition 1 The Fourier matrix of order n is the matrix

$$F_n = \frac{1}{\sqrt{n}} \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 & 1 \\ 1 & \omega^{n-1} & \omega^{n-2} & \cdots & \omega^2 & \omega \\ 1 & \omega^{n-2} & \omega^{n-4} & \cdots & \omega^4 & \omega^2 \\ & & & \ddots & & \\ 1 & \omega^2 & \omega^4 & \cdots & \omega^{n-4} & \omega^{n-2} \\ 1 & \omega & \omega^2 & \cdots & \omega^{n-2} & \omega^{n-1} \end{pmatrix} \quad (12)$$

Denote the Hermitian conjugate (transpose, complex conjugate) of this by F_n^* .

Lemma 3 ([109])

- a) *The Fourier matrix is unitary: $F_n F_n^* = F_n^* F_n = I$.*
- b) *The eigenvalues of F_n are ± 1 and $\pm i$ with multiplicities depending on n .*
- c) *The characteristic polynomials of F_n^* are*

$$\begin{aligned} (\lambda - 1)^2 (\lambda - i) (\lambda + 1) (\lambda^4 - 1)^{\frac{n}{4}-1} & \quad n = 0 \text{ mod}(4) \\ (\lambda - 1) (\lambda^4 - 1)^{(n-1)/4} & \quad n = 1 \text{ mod}(4) \\ (\lambda^2 - 1) (\lambda^4 - 1)^{(n-2)/4} & \quad n = 2 \text{ mod}(4) \\ (\lambda - i) (\lambda^2 - 1) (\lambda^4 - 1)^{(n-3)/4} & \quad n = 3 \text{ mod}(4) \end{aligned} \quad (13)$$

Every $n \times n$ circulant matrix \mathbf{A} is diagonalized by F_n . Further, if P_A is the polynomial defined by Eq. (11) then

$$F_n \mathbf{A} F_n^* = \Lambda(\mathbf{A}) = \text{diag}[P_A(1), P_A(\omega), \dots, P_A(\omega^{n-1})] \quad (14)$$

hence the r th eigenvalue of \mathbf{A} is $P_A(\omega^r)$. Define the $n \times n$ matrix $\pi_r = \text{diag}(0, \dots, 0, 1, 0, \dots, 0)$ with the 1 in the r th position and set $\Pi_r = F_n^* \pi_r F_n$. The matrices Π_r are Hermitian and satisfy the conditions

$$\Pi_r \Pi_s = \begin{cases} 0 & r \neq s \\ \Pi_s & r = s \end{cases}$$

$$\sum_{r=0}^{n-1} \Pi_r = I$$

Thus, they are orthogonal, idempotent, and form a resolution of unity. Hence, they are a complete set of projection matrices.

Lemma 4 ([109]) *Let $\mathbf{A} = \text{circ}(a_0, \dots, a_{n-1})$. Then*

$$\mathbf{A} = \sum_{r=0}^{n-1} P_A(\omega^r) \Pi_r \quad (15)$$

Representation of an additive rule in terms of complex polynomials yields an interesting result on injectivity. A rule $X : \mathcal{E}(\mathcal{A}, \mathcal{Z}) \mapsto \mathcal{E}(\mathcal{A}, \mathcal{Z})$ is *surjective* if every configuration has a predecessor. If the predecessor of a configuration having a predecessor is unique, the rule is *injective*. A rule that is both surjective and injective is bijective. For cellular automata, injectivity is equivalent to reversibility. If a rule X is reversible, there is an inverse rule; that is, a reversible rule X^{-1} such that if $\xi \in \mathcal{E}(\mathcal{A}, \mathcal{Z})$ and $X(\eta) = \xi$, then $X^{-1}(\xi) = \eta$ [110,111,112,113,114,115]. The question of whether or not a cellular automata rule is surjective or injective is decidable only in dimension one [112]. Injective additive rules are also called group rules [52]. Those with maximum period cycles provide an effective means of generating pseudorandom sequences. For a binary valued rule operating on strings of length n and all divisors of n , the maximal possible cycle period is $2^n - 1$ (i. e., every state but the 0 state is on the cycle). If a rule operating on strings of length n has periodic boundary conditions, the shift operator produces cycles of length n . As a result, no additive rule acting on strings with periodic boundary conditions can produce cycles of maximal period. If null boundary conditions are used, however, maximal period cycles can appear [79].

For a given cellular automata rule, the set of states having no predecessor is called the Garden-of-Eden (GoE). All additive rules acting on $\mathcal{E}(\mathcal{A}, \mathcal{Z})$ are surjective but this is not true of rules acting on $\mathcal{E}(\mathcal{A}, \mathcal{Z}_n)$. For additive rules acting on strings of length n with periodic boundary conditions there will be configurations having no predecessors. Nevertheless, these Garden-of-Eden states are not intrinsic to the rule since they do have predecessors when the state space $\mathcal{E}(\mathcal{A}, \mathcal{Z}_n)$ is embedded in $\mathcal{E}(\mathcal{A}, \mathcal{Z})$, as the following example shows.

Example 5 Let $X : \mathcal{E}(\{0, 1\}, \mathcal{Z}) \mapsto \mathcal{E}(\{0, 1\}, \mathcal{Z})$ be defined by the rule table

000	001	010	011	100	101	110	111
0	0	1	1	1	1	0	0

For $X : \mathcal{E}(\{0, 1\}, \mathcal{Z}_3) \mapsto \mathcal{E}(\{0, 1\}, \mathcal{Z}_3)$ the states $\{001, 010, 100, 111\}$ have no predecessors. It is easy to show, however, that these states do have predecessors in \mathcal{Z}_6 when \mathcal{Z}_3

and \mathcal{Z}_6 are embedded in \mathcal{Z} :

$$\begin{aligned} \{000111, 111000\} &\mapsto 001; \\ \{001110, 110001\} &\mapsto 010; \\ \{011100, 100011\} &\mapsto 100; \\ \{01, 10\} &\mapsto 1 \end{aligned}$$

While all additive rules are surjective in this sense, not all additive rules are injective. The condition for injectivity is easily seen using the representation in terms of n th roots of unity. From Eq. (14) a rule will be injective on $\mathcal{E}(\mathcal{A}, \mathcal{Z}_n)$ if and only if its diagonalized matrix representation is invertible, hence there can be no zero eigenvalues.

Lemma 5 ([81,116]) *Let $X : \mathcal{E}(\mathcal{A}, \mathcal{Z}_n) \mapsto \mathcal{E}(\mathcal{A}, \mathcal{Z}_n)$ be an additive rule represented by $\mathbf{A} = \text{circ}(a_0, \dots, a_{n-1})$. X is injective on $\mathcal{Z}_n \subset \mathcal{Z}$ if and only if $P_A(\omega^s) \neq 0 \pmod{p}$ for $0 \leq s \leq n-1$.*

If the rule in Lemma 5 is to be injective on \mathcal{Z} then it must be so on \mathcal{Z}_n for all n . Thus the complex polynomial $P_A(z)$ must be irreducible with respect to all n th roots of unity, for all n . This leads to the next theorem:

Theorem 2 ([81,116]) *Let $X : \mathcal{E}(\mathcal{A}, \mathcal{Z}) \mapsto \mathcal{E}(\mathcal{A}, \mathcal{Z})$ be an additive rule represented as in Eq. (9). X will be injective if and only if*

$$\lim_{\varepsilon \rightarrow 0} \frac{1}{2\pi i} \oint_{c(\varepsilon)} \frac{P'_A(z)}{P_A(z)} dz = 0 \quad (16)$$

where $P'_A(z)$ is the derivative of $P_A(z)$ and $c(\varepsilon)$ is a contour consisting of circles of radius $1 + \varepsilon$ and $1 - \varepsilon$.

This follows since the integral in Eq. (16) counts the number of zeros minus the number of poles of $P_A(z)$ contained within the contour. Since there are no poles, in the limit this counts the number of zeros on the unit circle.

Corollary 2 ([81]) *Let $X : \mathcal{E}(\{0, 1\}, \mathcal{Z}_n) \mapsto \mathcal{E}(\{0, 1\}, \mathcal{Z}_n)$ be defined by $X = \sum_{s=0}^{k-1} \sigma^s$.*

1. *If k is even, X is not injective.*
2. *If k is odd, X will be injective for all n such that $n \neq mk$ for any m .*

Table 1 lists all rules that are injective on $\mathcal{E}(\{0, 1\}, \mathcal{Z}_n)$ for at least some n , with $k \leq 5$.

Transient Lengths and Cycle Periods

For any cellular automata acting on a finite state space, every state eventually maps to a fixed point or cycle. If

Additive Cellular Automata, Table 1

Injective Rules for $k \leq 5$

X	k	injectivity condition
σ^4	5	none
σ^3	4	none
σ^2	3	none
σ	2	none
I	1	none
$\sigma^2 + \sigma^3 + \sigma^4$	5	$n \neq 3m$
$\sigma + \sigma^3 + \sigma^4$	5	none
$\sigma + \sigma^2 + \sigma^4$	5	none
$I + \sigma^3 + \sigma^4$	5	none
$I + \sigma^2 + \sigma^4$	5	$n \neq 3m$
$I + \sigma + \sigma^4$	5	none
$\sigma + \sigma^2 + \sigma^3$	4	$n \neq 3m$
$I + \sigma^2 + \sigma^3$	4	none
$I + \sigma + \sigma^3$	4	none
$I + \sigma + \sigma^2$	3	$n \neq 3m$
$I + \sigma + \sigma^2 + \sigma^3 + \sigma^4$	5	$n \neq 5m$

a rule is injective, it is reversible and every state is a fixed point, or is on a cycle. If not injective, there will be states without predecessors, Garden-of-Eden states. As indicate, however, if a rule is additive its Garden-of-Eden states are spurious in the sense that they do have predecessors if the state space is enlarged.

The following theorem lists several significant properties of cellular automata rules acting on $\mathcal{E}(\mathcal{A}, \mathbb{Z})$ or $\mathcal{E}(\mathcal{A}, \mathbb{Z}^+)$ with left justified neighborhoods.

Theorem 3 ([81]) *Let X be a k -site cellular automata rule acting on $\mathcal{E}(\mathcal{A}, \mathbb{Z})$ or on $\mathcal{E}(\mathcal{A}, \mathbb{Z}^+)$ with left justified neighborhoods. Then the following statements are equivalent: (a) X is surjective, (b) X has an empty Garden-of-Eden, (c) Every finite sequence $\mu_0 \dots \mu_{n-1}$ has exactly p^{k-1} pre-images and every state μ has at most p^{k-1} predecessors, (d) X maps eventually periodic states to eventually periodic states and non-periodic states to non-periodic states, (e) as a map of the interval $[0,1]$ X maps rationals to rationals and irrationals to irrationals.*

If $X : \mathcal{E}(\mathcal{A}, \mathbb{Z}_n) \mapsto \mathcal{E}(\mathcal{A}, \mathbb{Z}_n)$ is a k -site rule with $|\mathcal{A}| = p$ and either periodic or null boundary conditions, the state transition diagram, $\text{STD}(X)$ is a graph with p^n vertices labeled by the set of p -adic numbers $\{i_0, \dots, i_{n-1} | 0 \leq i_r \leq p-1\}$. An edge is directed from the vertex i_0, \dots, i_{n-1} to the vertex j_0, \dots, j_{n-1} if and only if $X(i_0, \dots, i_{n-1}) = j_0, \dots, j_{n-1}$. Each state μ maps to a unique state $X(\mu)$ so $\text{STD}(X)$ consists of a set of trees rooted on fixed points or cycles. States at the top of trees are Garden-of-Eden states.

If $h(X, n)$ is the maximum tree height, states at heights $\underline{h} \leq h(X, n)$ cannot appear after $h(X, n) - \underline{h} + 1$ iterations and after $h(X, n)$ iterations only fixed points and states on cycles remain. Thus, iteration of a non-injective rule on $\mathcal{E}(\mathcal{A}, \mathbb{Z}_n)$ decreases the number of available states with a corresponding reduction in entropy. On the other hand, non-injective additive rules acting on $\mathcal{E}(\mathcal{A}, \mathbb{Z}^+)$ do not reduce entropy [117] even though they do so on $\mathcal{E}(\mathcal{A}, \mathbb{Z}_n)$ for all n . The explanation for this apparent paradox is that the Garden-of-Eden states that appear in $\mathcal{E}(\mathcal{A}, \mathbb{Z}_n)$ are artifacts of the finite length of states in this space. When embedded in \mathbb{Z}^+ , states in \mathbb{Z}_n correspond to periodic configurations, hence to rational numbers in $[0,1]$ and the set of all rationals has measure 0 in the reals.

Parameters of interest for characterizing state transition diagrams of rules acting on $\mathcal{E}(\mathcal{A}, \mathbb{Z}_n)$ are the maximum tree height $h(X, n)$ and the cycle periods $c_s(X, n)$.

Theorem 4 ([102])

1. Trees rooted at all vertices on cycles or at fixed points of the STD for additive cellular automata are isomorphic to the tree rooted at the fixed point 0.
2. The periods of all cycles of an additive rule acting on \mathbb{Z}_n are divisors of the period for cycles obtained by starting from an initial state containing only a single 1.
3. Let $c(m)$ be the maximum cycle period for an additive cellular automaton acting on \mathbb{Z}_m and take $n = 2m$. Then $c(n)$ divides $2c(m)$.
4. Let $n = 2^k m$, m odd. The maximum cycle period $c(n)$ for an additive rule acting on \mathbb{Z}_n satisfies $c(n) | 2^{\text{ord}(n,m)} - 2^m$ where $\text{ord}(n, m) = \min\{r | 2^r \equiv 2^m \pmod{m}\}$

In most cases, the maximum cycle period equals $2^{\text{ord}(n,m)} - 2^m$ or, if the rule is symmetric, $2^{\text{sord}(n,m)} - 2^m$. In [104,118], Jen shows that when this is not the case, it is a number theoretic consequence of an anomalous shift that reduces the maximum cycle period. As indicated, the choice of mapping site can influence cycle periods and the effect of anomalous shifts is analogous. This is an immediate result of the next theorem.

Theorem 5 ([77,118]) *Let $X : \mathcal{E}(\mathcal{A}, \mathbb{Z}_n) \mapsto \mathcal{E}(\mathcal{A}, \mathbb{Z}_n)$ be the global mapping for a cellular automaton rule. A state μ is on a cycle of X if and only if there exist integers r and s such that $X^r(\mu) = \sigma^{-s}(\mu)$.*

In some cases $r = c(n)$ and $s = n$ so the theorem is not as strong as it might first appear. If $X^r(\mu) = \sigma^{-s}(\mu)$ for all states on cycles of maximum period, however, then $c(n) = rt$ where $t = \min\{j | js \equiv 0 \pmod{n}\}$. A change of mapping site is equivalent to multiplication by a power

of the shift σ . If μ is on a cycle of maximum period then $(\sigma^k \mathcal{X})^r(\mu) = \sigma^{k-s}(\mu)$ and the cycle period is rq with $q = \min\{j \mid j(k-s) = 0 \pmod{n}\}$. In general, there is no requirement that $q = t$.

Comprehensive results on cycle periods and maximum transient lengths in the case of null or periodic boundary conditions were first obtained by Elspasa [119] in work characterizing the cycle sets in the state transition diagrams of linear machines. Since then a number of researchers have independently derived similar results [120,121,122,123,124,125,126]. Let $\mathbf{A} = \text{circ}(a_0, \dots, a_{n-1})$ and let μ be a state in $\mathcal{E}(\mathcal{A}, \mathcal{Z}_n)$. The *minimal annihilating polynomial* of μ is the monic polynomial $P_\mu(z)$ such that $P_\mu(\mathbf{A})\mu = \mathbf{0} \pmod{p}$. This polynomial exists since \mathbf{A} always satisfies its characteristic equation. Let $P_\mu(z) = z^k \underline{P}_\mu(z)$ with $\underline{P}_\mu(0) \neq 0$. The order of $P_\mu(z)$, $\text{ord}(P_\mu(z))$, is defined as the smallest natural number c such that $\underline{P}_\mu(z) \mid (z^c - 1)$. The following theorem is given in [121,122]. Alternate versions appear in [123,124,126].

Theorem 6 *Let $\mu \in \mathcal{E}(\mathcal{A}, \mathcal{Z}_n)$ with minimal annihilating polynomial $P_\mu(z) = z^k \underline{P}_\mu(z)$. Then $\mathbf{A}^k \mu$ belongs to a cycle with period $c = \text{ord}(\underline{P}_\mu(z))$.*

Since the minimal annihilating polynomial always divides the minimal polynomial, which, for additive cellular automata represented by circulant matrices, or by the corresponding null boundary condition matrix, is the same as the characteristic polynomial of that matrix, all cycle periods and maximum transient lengths can be found from the characteristic polynomial. Hence, the maximum cycle period is the order of the characteristic polynomial since there always exists a state whose minimal annihilating polynomial is the minimal polynomial.

The questions of reachability, and conditions for states to be on cycles is also addressed using the formulation in terms of roots of unity. If an additive rule acting on $\mathcal{E}(\mathcal{A}, \mathcal{Z}_n)$ is represented by a circulant matrix \mathbf{A} and a state μ is represented by $\mu = \sum_{s=0}^{n-1} \mu_s \omega^s$, write

$$P_{\mathbf{A}}(\omega) = \prod_{j=1}^r \rho_j(\omega) \prod_{k=1}^s \Omega_k(\omega) = \rho(\omega) \Omega(\omega) \quad (17)$$

where $\rho(\omega)$ is a product of the irreducible factors of $P_{\mathbf{A}}(\omega)$ representing injective rules, and $\Omega(\omega)$ is a product of the irreducible factors representing non-injective rules. Diagonalization of \mathbf{A} yields $\Lambda(\mathbf{A}) = \Lambda(\rho) \Lambda(\Omega)$ and $\Lambda^{-1}(\rho)$ exists since $\rho(\omega)$ represents injective (hence reversible) rules. Let v denote the nullity of $\Lambda(\Omega)$.

Theorem 7 ([81,127]) *A state $\mu(\omega)$ is reachable by an additive cellular automata rule with circulant matrix rep-*

resentation \mathbf{A} if and only if $\Lambda(\Omega) \mid \mu(\omega)$. The fraction of reachable configurations is 2^{-v} . Further, if $d(\mu)$ is the in degree of the STD vertex labeled by μ then

$$d(\mu) = \begin{cases} 0 & \mu \text{ is a Garden-of-Eden state} \\ 2^v & \text{otherwise} \end{cases} \quad (18)$$

Example 6a (Rule 90 acting on $\mathcal{E}(\{0,1\}, \mathcal{Z}_n)$ with periodic boundary conditions) The circulant matrix for nearest neighbor rule 90 acting on \mathcal{Z}_n with periodic boundary conditions is $\mathbf{A}(\delta) = \text{circ}(0, 1, 0, \dots, 0, 1)$. The characteristic polynomial of this matrix satisfies the recurrence relation $Q_{n+1}(x) = xQ_n(x) + Q_{n-1}(x) \pmod{2}$. For $n = 2^s m$ with m odd, the characteristic polynomial of $\mathbf{A}(\delta)$ has the form

$$Q_n(x) = x^{2^s} r_m^{2^s+1}(x) \quad (19)$$

$$r_{2k+1}(x) = x r_{2k-1}(x) + r_{2k-3}(x)$$

and the minimal annihilating polynomial is [108]

$$\begin{aligned} x r_m(x) & \quad s = 0 \\ x^{2^{s-1}} r_m^{2^s}(x) & \quad s > 0 \end{aligned} \quad (20)$$

where $r_m(0) \neq 0$. For $n = 6$ ($s = 1, m = 3$), the circulant matrix representing rule 90 is given by Eq. (10a). The characteristic polynomial of this matrix, with coefficients reduced mod(2), is $x^2(x^4 + 1) = x^2(x + 1)^4$. Thus, the minimal polynomial is $x(x + 1)^2 = x(x^2 + 1)$ and $r_3(x) = (x + 1)$. The minimum integer c such that $(x^2 + 1)$ divides $(x^c + 1)$ is just $c = 2$, showing that the maximum cycle period is 2.

Example 6b (Rule 90 Represented by Roots of Unity, Acting on \mathcal{Z}_6) The Theorem 5 condition $\mathcal{X}^r(\mu) = \sigma^{-s}(\mu)$ for a state μ to be on a cycle can be written as $\Lambda^r(\mathcal{X})F_n(\mu) = \Lambda^{n-s}(\sigma)F_n(\mu)$. This gives the conditions for μ to be on a cycle as a set of linear equations [81]. For $n = 6$, elementary rule 90 has the form $\delta = \sigma + \sigma^5$, or in terms of the sixth root of unity, $\delta(\omega) = \omega + \omega^5$. Using $\omega^3 = -1$ with all sums taken mod(2)

$$\begin{aligned} \Lambda(\delta) &= \text{diag}(0, \omega + \omega^5, \omega^2 + \omega^4, 0, \omega^2 + \omega^4, \omega + \omega^5) \\ \Lambda^2(\delta) &= \text{diag}(0, \omega^2 + \omega^4, \omega^2 + \omega^4, 0, \omega^2 + \omega^4, \omega^2 + \omega^4) \\ \Lambda^3(\delta) &= \text{diag}(0, \omega + \omega^5, \omega^2 + \omega^4, 0, \omega^2 + \omega^4, \omega + \omega^5) \end{aligned} \quad (21)$$

hence $\Lambda^3(\delta) = \Lambda(\delta)$ or equivalently, $\Lambda(\delta)(\Lambda^2(\delta) + I) = 0 \pmod{2}$. Thus, the maximum tree height is one and the maximum cycle period is two. In addition, $v = 2$ so that 1/4 of the total of 64 states will be on cycles. Further,

for $n = 6$ (observing that $-1 = 1 \pmod{2}$)

$$F_6(\mu) = \frac{1}{\sqrt{6}} \begin{pmatrix} \mu_0 + \mu_1 + \mu_2 + \mu_3 + \mu_4 + \mu_5 \\ \mu_0 + \mu_1\omega^5 + \mu_2\omega^4 + \mu_3\omega^3 + \mu_4\omega^2 + \mu_5\omega \\ \mu_0 + \mu_1\omega^4 + \mu_2\omega^2 + \mu_3 + \mu_4\omega^4 + \mu_5\omega^2 \\ \mu_0 + \mu_1\omega^3 + \mu_2 + \mu_3\omega^3 + \mu_4 + \mu_5\omega^3 \\ \mu_0 + \mu_1\omega^2 + \mu_2\omega^4 + \mu_3 + \mu_4\omega^2 + \mu_5\omega^4 \\ \mu_0 + \mu_1\omega + \mu_2\omega^2 + \mu_3\omega^3 + \mu_4\omega^4 + \mu_5\omega^5 \end{pmatrix} \quad (22)$$

Since $\omega^2 + \omega^4 = \omega^3 = -1 = 1 \pmod{2}$, $\Lambda^2(\delta) = \text{diag}(0, 1, 1, 0, 1, 1)$ and the condition $\Lambda^2(\delta)F_6(\mu) = F_6(\mu)$ requires that $[F_6(\mu)]_0 = [F_6(\mu)]_3 = 0$ which reduces to $\mu_0 + \mu_2 + \mu_4 = \mu_1 + \mu_3 + \mu_5 = 0$ or equivalently, $\mu_4 = \mu_0 + \mu_2$ and $\mu_5 = \mu_1 + \mu_3$. Hence a state μ will be on a cycle if and only if it has the form $\mu = (\mu_0, \mu_1, \mu_2, \mu_3, \mu_0 + \mu_2, \mu_1 + \mu_3)$.

Computing Predecessor States

A problem of general interest for cellular automata is computation of predecessor states. For a rule $\mathcal{X} : \mathcal{E}(\mathcal{A}, \mathcal{L}) \mapsto \mathcal{E}(\mathcal{A}, \mathcal{L})$ with a state β given this requires solution of the equation $\mathcal{X}(\mu) = \beta$. It is always possible to construct solutions for this equation, or to show that none exist by a method of backward reconstruction based on the rule table.

Example 7 (Rule 60 Acting on \mathbb{Z}_4 With Periodic Boundary Conditions) Rule 60 is a 2-site rule, defined by $(00, 11) \mapsto 0, (01, 10) \mapsto 1$. Given the state 0110 the predecessors of this state can be computed as follows:

1. The initial 0 in 0110 can arise from either 00 or 11.
2. Starting with a 00, the next symbol in 0110 is a 1 and this can arise from a 01 or a 10, but this must also connect to the original 00 so only 01 is allowed, giving 001. Starting from a 11, on the other hand, the same reasoning requires 110.
3. The third symbol in 0110 is also a 1. To be consistent with 001 requires that 10 be selected, and to be consistent with 110 requires that 01 be selected, thus giving the two partially constructed possibilities as 0010 and 1101.
4. Finally, the fourth symbol must be a 0. This requires that the predecessor string conclude with either 00 or 11. Since the strings are in \mathbb{Z}_4 with periodic boundary conditions, the final symbol in the predecessor string must also be the first symbol in that string. Thus, both 0010 and 1101 are seen to be predecessors of 0110.

Other ways of computing predecessor states for finite strings is through the construction of a rule matrix [81]

or the use of de Bruijn diagrams [81,113]. Backward reconstruction, the rule matrix, and use of a de Bruijn diagram are valid methods for computing predecessor states for all one-dimensional rules. For additive rules, however, there is an analytic means for computing predecessor states, starting from left justified neighborhoods defined on $\mathcal{E}(\mathcal{A}, \mathbb{Z}_n)$ or $\mathcal{E}(\mathcal{A}, \mathbb{Z}^+)$ [81,128]. This can be illustrated for rules defined on $\mathcal{E}(\{0, 1\}, \mathbb{Z}^+)$. This method also works for rules defined on $\mathcal{E}(\{0, 1\}, \mathbb{Z}_n)$ if it is embedded in $\mathcal{E}(\{0, 1\}, \mathbb{Z}^+)$ as the subset of half-infinite periodic sequences with periods that divide n . Define operators $\mathcal{B} : \mathcal{E}(\{0, 1\}, \mathbb{Z}^+) \mapsto \mathcal{E}(\{0, 1\}, \mathbb{Z}^+)$ and $\sigma^{-1} : \mathcal{E}(\{0, 1\}, \mathbb{Z}^+) \mapsto \mathcal{E}(\{0, 1\}, \mathbb{Z}^+)$ by

$$[B(\mu)]_s = \sum_{i=0}^s \mu_i \pmod{2} \quad (23)$$

$$[\sigma^{-1}(\mu)]_s = \begin{cases} 0 & s = 0 \\ \mu_{s-1} & s > 0 \end{cases}$$

Theorem 8 ([81,128]) Let $D = I + \sigma$ be the global operator for elementary rule 60 acting on $\mathcal{E}(\{0, 1\}, \mathbb{Z}^+)$ and define the state $\alpha^{(s)}$ in \mathbb{Z}^+ by

$$[\alpha^{(s)}]_i = \begin{cases} 0 & i \neq s \\ 1 & i = s \end{cases}$$

1. The general solution of $D(\mu) = \beta$ is

$$\mu = a_0 B(\alpha^{(0)}) + B\sigma^{-1}(\beta).$$

2. The general solution of $D^k(\mu) = \beta$ is

$$\mu = \sum_{s=0}^{k-1} a_s B^{s+1}(\alpha^{(s)}) + B^k \sigma^{-k}(\beta),$$

where the coefficients a_s provide initial conditions.

The general technique for computing predecessors can be illustrated with the case of rule 150, expressed in left justified form as $\mathcal{X} = I + \sigma + \sigma^2$. To solve $(I + \sigma + \sigma^2)(\mu) = \beta$ define four sequences:

$$\begin{aligned} \mu_i^{(0)} &= \mu_{2i}, & \mu_i^{(1)} &= \mu_{2i+1} \\ \beta_i^{(0)} &= \beta_{2i}, & \beta_i^{(1)} &= \beta_{2i+1} \quad i = 0, 1, 2, \dots \end{aligned} \quad (24)$$

The equation $(I + \sigma + \sigma^2)(\mu) = \beta$ reduces to the pair of coupled equations

$$\begin{aligned} \mu_i^{(0)} + \mu_{i+1}^{(0)} &= \mu_i^{(1)} + \beta_i^{(0)} \\ \mu_i^{(1)} + \mu_{i+1}^{(1)} &= \mu_{i+1}^{(0)} + \beta_i^{(0)} \end{aligned} \quad (25)$$

and these can be written as

$$\begin{aligned} D(\mu^{(0)}) &= \mu^{(1)} + \beta^{(0)} \\ D(\mu^{(1)}) &= \sigma(\mu^{(0)}) + \beta^{(1)} \end{aligned} \quad (26)$$

From Theorem 8, Eq. (26) can be formally solved to obtain

$$\begin{aligned} \mu^{(0)} &= a_0 B(\alpha_{(0)}) + B\sigma^{-1}(\mu^{(1)} + \beta^{(0)}) \\ \mu^{(1)} &= a_1 B(\alpha_{(0)}) + B\sigma^{-1}(\sigma\mu^{(0)} + \beta^{(1)}) \end{aligned} \quad (27)$$

Substituting the second equation of (27) into the first, making use of the identity $\sigma^{-2}\sigma(\mu^{(0)}) = \sigma^{-1}(\mu^{(0)} + \mu_0^{(0)}\alpha^{(0)})$, rearranging terms and absorbing the term $\mu_0^{(0)}\alpha^{(0)}$ into the constant parameter, results in the equation

$$(I + B^2\sigma^{-1})(\mu^{(0)}) = B(a_0\alpha^{(0)} + a_1B(\alpha^{(1)})) + B\sigma^{-1}(\beta^{(0)}) + B^2\sigma^{-2}(\beta^{(1)}) \quad (28)$$

Theorem 9 ([81,128]) *The operator $(I + B^2\sigma^{-1})$ is invertible with $(I + B^2\sigma^{-1})^{-1} = I + C_{(2,1)}$ with*

$$[C_{(2,1)}(\xi)]_i = \begin{cases} 0 & i = 0 \\ \sum_{s=0}^{\lceil \frac{i-1}{3} \rceil} (\xi_{i-3s-2} + \xi_{i-3s-1}) & i > 0 \end{cases} \quad (29)$$

where all sums are taken mod(2), $\xi_r = 0$ for $r < 0$, and $\lceil x \rceil$ indicates the greatest integer less than or equal to x .

The solution for $\mu^{(0)}$ is substituted into the second equation of (27) yielding a solution for $\mu^{(1)}$. These are recombined to get the general solution for μ . This technique of reducing a single equation to a set of coupled equations involving simpler additive rules works in general although the form for partitioning of sequences is specific to the particular case. Computation of predecessors involves inversion of operators of the form $I + B^r\sigma^{-s}$. The general form for the inverse of this operator is $I + C_{(r,s)}$ where $C_{(r,s)}$ is the lower triangular matrix that is the solution of the equation

$$\sum_{m=j}^{j+r} \binom{r+1}{m-j+1} [C_{(r,s)}]_{im} + [C_{(r,s)}]_{i,j+s} = \delta_{i,j+s} \quad (30)$$

d-Dimensional Rules

Both [102,105] discuss the extension from one-dimensional to d -dimensional rules defined on tori. In [102] this

discussion uses a formalism of multinomials defined over finite fields. In [105], the one-dimensional analysis based on circulant matrices is generalized. The matrix formalism of state transitions is retained by defining a d -fold “circulant of circulants,” which is not, of itself, necessarily a circulant. Computation of the non-zero eigenvalues of this matrix yields results on transient lengths and cycle periods.

More recently, an extensive analysis of additive rules defined on multi-dimensional tori has appeared [129]. A d -dimensional integer vector $\vec{n} = (n_1, \dots, n_d)$ defines a discrete toroidal lattice $\mathcal{L}(\vec{n})$. Every d -dimensional matrix of size \vec{n} with entries in \mathcal{A} , $|\mathcal{A}| = p$ (prime), defines an additive rule acting on $\mathcal{E}(\mathcal{A}, \mathcal{L}(\vec{n}))$ as follows: Let \mathcal{T} and $\mu(t)$ be elements of $\mathcal{E}(\mathcal{A}, \mathcal{L}(\vec{n}))$ with \mathcal{X} the rule defined by \mathcal{T} and $\mu(t)$ a state at time t . The state transition defined by \mathcal{X} is $\mu(t+1) = \mathcal{X}(\mu(t))$ and this is given by

$$\begin{aligned} [\mu(t+1)]_{i_1 \dots i_d} &= \sum_{k_1, \dots, k_d} [C(\mathcal{T})]_{i_1 \dots i_d}^{k_1 \dots k_d} [\mu(t)]_{k_1 \dots k_d} \\ [C(\mathcal{T})]_{i_1 \dots i_d}^{k_1 \dots k_d} &= \mathcal{T}_{j_1 \dots j_d} \quad j_s = k_s - i_s \text{ mod}(n_s) \end{aligned} \quad (31)$$

The matrix $C(\mathcal{T})$ is the d -dimensional generalization of a circulant matrix with \mathcal{T} as the equivalent of its first row. For example, if $d = 1$ and $p = 2$ with $\mathcal{T} = (0, 1, 0, 0, 0, 1)$ this defines the additive rule $\sigma + \sigma^5$ (rule 90) and the matrix $C(\mathcal{T})$ is given in Eq. (10a).

Let S and \mathcal{T} be elements of $\mathcal{E}(\mathcal{A}, \mathcal{L}(\vec{n}))$ and define the binary operation $\psi : \mathcal{E}(\mathcal{A}, \mathcal{L}(\vec{n})) \times \mathcal{E}(\mathcal{A}, \mathcal{L}(\vec{n})) \mapsto \mathcal{E}(\mathcal{A}, \mathcal{L}(\vec{n}))$ by

$$[\psi(S, \mathcal{T})]_{i_1 \dots i_d} = \sum_{\substack{k_1, \dots, k_d \\ 0 \leq k_s < n_s}} S_{k_1 \dots k_d} \mathcal{T}_{i_1 - k_1 \dots i_d - k_d} \quad (32)$$

with all sums taken mod(p).

Lemma 6 ([129]) *Let S and \mathcal{T} be as above, with generalized circulant matrices $C(S), C(\mathcal{T})$. The product $C(S)C(\mathcal{T})$ defined by*

$$\begin{aligned} [C(S)C(\mathcal{T})]_{i_1 \dots i_d}^{j_1 \dots j_d} &= \sum_{k_1 \dots k_d} [C(S)]_{i_1 \dots i_d}^{k_1 \dots k_d} [C(\mathcal{T})]_{k_1 \dots k_d}^{j_1 \dots j_d} \text{ mod}(p) \end{aligned} \quad (33)$$

is also a generalized circulant and $C(S)C(\mathcal{T}) = C(\psi(S, \mathcal{T}))$.

An important tool for analysis of additive rules on multi-dimensional tori is the discrete baker transformation. This

is a discrete version of the baker transformation (Bernoulli shift) for continuous one dimensional dynamical systems [130]. The discrete baker transformation is the operation $B_p : \mathcal{E}(\mathcal{A}, \mathcal{L}(\vec{n})) \mapsto \mathcal{E}(\mathcal{A}, \mathcal{L}(\vec{n}))$ defined by

$$[B_p * \mathcal{T}]_{i_1 \dots i_d} = \sum_{i_s = p k_s \bmod(n_s)} \mathcal{T}_{k_1 \dots k_d} \bmod(p) \quad (34)$$

with the empty sum set to 0. In the one dimensional case with $p = 2$ and $\mathcal{T} = (a_0, \dots, a_{n-1})$ this becomes

$$[B_2 * \mathcal{T}]_i = \sum_{k: i=2k \bmod(n)} a_k \bmod(2) \quad (35)$$

Example 8 If $d = 1$, $p = 2$ and $n = 7$ then $B_2(a_0, a_1, a_2, a_3, a_4, a_5, a_6) = (a_0, a_4, a_1, a_5, a_2, a_6, a_3)$. On the other hand, if $n = 6$ then $B_2(a_0, a_1, a_2, a_3, a_4, a_5) = (a_0 + a_3, 0, a_1 + a_4, 0, a_2 + a_5, 0)$.

In one dimension, if \mathcal{X} is the rule defined by $\mathcal{T} = (a_0, \dots, a_{n-1})$ then $B_p \mathcal{X} = \mathcal{X}^2$.

Example 9 If $d = 2$ and $p = 3$ with $n_1 = 5$, $n_2 = 6$ and

$$\mathcal{T} = \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} & a_{04} \\ a_{10} & a_{11} & a_{12} & a_{13} & a_{14} \\ a_{20} & a_{21} & a_{22} & a_{23} & a_{24} \\ a_{30} & a_{31} & a_{32} & a_{33} & a_{34} \\ a_{40} & a_{41} & a_{42} & a_{43} & a_{44} \\ a_{50} & a_{51} & a_{52} & a_{53} & a_{54} \end{bmatrix}$$

then

$$B_3 * \mathcal{T} = \begin{bmatrix} a_{00} + a_{20} + a_{40} & a_{02} + a_{22} + a_{42} & a_{04} + a_{24} + a_{44} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ a_{30} + a_{10} + a_{50} & a_{32} + a_{12} + a_{52} & a_{34} + a_{14} + a_{54} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ & a_{01} + a_{21} + a_{41} & a_{03} + a_{23} + a_{43} \\ & 0 & 0 \\ & 0 & 0 \\ a_{31} + a_{11} + a_{51} & a_{33} + a_{13} + a_{53} \\ & 0 & 0 \\ & 0 & 0 \end{bmatrix}$$

with all sums mod(3).

The discrete baker transformation exponentially speeds up rule evolution.

Theorem 10 ([129])

If p is prime then $[C(\mathcal{T})]^{p^r} = C(B_p^r * \mathcal{T})$.

Thus, if the matrix \mathcal{T} defines a rule $\mathcal{X} : \mathcal{E}(\mathcal{A}, \mathcal{L}(\vec{n})) \mapsto \mathcal{E}(\mathcal{A}, \mathcal{L}(\vec{n}))$ then \mathcal{X}^{p^r} is obtained from the matrix $B_p^r * \mathcal{T}$.

The operator B_p is a permutation if every sum in Eq. (34) contains at most one non-zero summand. In Example 7, B_2 is a permutation for $n = 7$ but is not for $n = 6$.

Lemma 7 ([129]) Set $\iota = \max\{ex_p n_s | 1 \leq s \leq d\}$. That is, ι is the largest integer k such that for some s , p^k divides n_s . Then for all $r > \iota$ and any $\mathcal{T} \in \mathcal{E}(\mathcal{A}, \mathcal{L}(\vec{n}))$, B_p is a permutation on $B_p^r * \mathcal{T}$.

Theorem 11 ([129]) Let q be prime and let $\text{ord}_m q$ be the order of q in m when this is defined and 1 otherwise. Write $n_s = p^{k_s} m_s$ and set $c = \text{lcm}(\text{ord}_{m_1} p, \dots, \text{ord}_{m_d} p)$. Then, for any rule \mathcal{X} defined by a matrix $\mathcal{T} \in \mathcal{E}(\mathcal{A}, \mathcal{L}(\vec{n}))$ the following are true:

1. $B_p^{t+c} * \mathcal{T} = B_p^t * \mathcal{T}$
2. The period of any cycle in $\text{STD}(\mathcal{X})$ divides $p^t(p^c - 1)$
3. The height of trees in $\text{STD}(\mathcal{X})$ does not exceed p^t .

The baker transformation is a linear transformation on the space of d -dimensional matrices with entries from \mathcal{A} . Since each element of this space defines an additive cellular automata rule, the vertices of the state transition diagram for the baker transformation can be labeled by these rules, and this is exhaustive.

Definitions:

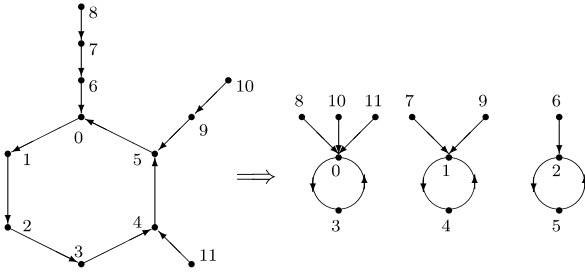
1. An oriented graph $G = (V, E)$ is a set V of vertices together with an edge set $E \subseteq V \times V$. If $(v, w) \in E$ then there is an edge directed from vertex v to vertex w .
2. An oriented graph $G_1 = (V_1, E_1)$ reduces to an oriented graph $G_2 = (V_2, E_2)$ modulo p ($G_1 <_p G_2$) if (a) $V_1 = V_2 = V$, and (b) $(v, w) \in E$ if and only if there is an oriented chain of length p from v to w in G_2 .

Example 10 Figure 2 shows the reduction modulo 3 of a connected graph with 12 vertices to a graph with three connected components.

The baker diagram for the space $\mathcal{E}(\mathcal{A}, \mathcal{L}(\vec{n}))$ has an edge directed from rule \mathcal{X} to rule \mathcal{Y} if and only if $\mathcal{Y} = B_p \mathcal{X}$.

Lemma 8 ([129]) If $(\mathcal{X}, \mathcal{Y})$ is an edge in $\text{STD}(B_p)$ then $\text{STD}(\mathcal{Y}) <_p \text{STD}(\mathcal{X})$.

Since $\mathcal{E}(\mathcal{A}, \mathcal{L}(\vec{n}))$ is finite, iteration of B_p on $\mathcal{E}(\mathcal{A}, \mathcal{L}(\vec{n}))$ must eventually reach a fixed point or cycle. If $\mathcal{T} \in \mathcal{E}(\mathcal{A}, \mathcal{L}(\vec{n}))$ set $\tau(\mathcal{T})$ equal to the sum mod(p) of all entries in \mathcal{T} . If \mathcal{X} is the rule defined by \mathcal{T} take $\eta(\mathcal{X})$ as the maximum height of trees in $\text{STD}(\mathcal{X})$, $\pi(\mathcal{X})$ as the number of vertices in $\text{STD}(\mathcal{X})$ lying on cycles, and $\theta_s(\mathcal{X})$ as the number of cycles of length s contained in $\text{STD}(\mathcal{X})$.



Additive Cellular Automata, Figure 2

Example of Graph Reduction Modulo 3 for a Graph with 12 Vertices (from [129])

Theorem 12 ([129])

1. If $STD(X) <_p STD(Y)$ then
 - a) $\eta(X) \leq \left\lceil \frac{\eta(Y)}{p} \right\rceil$
 - b) If j is the largest number such that $j\theta_{pj}(Y) > 0$ then $\theta_{pj}(X) = 0$
 - c) $\pi(X) = \pi(Y)$.
2. Let X, Y be two rules belonging to a cycle R of $STD(B_p)$ with period l_R and let $\{c_1, \dots, c_k\}$ be the set of all cycle periods in $STD(X)$. Then
 - a) $STD(X)$ and $STD(Y)$ are isomorphic as graphs
 - b) $\eta(X) \leq 1$ and $p \nmid c_s$ for all s
 - c) For all s , $c_s | (p^{l_R} - 1)$, $l_R = lcm(ord_{c_1} p, \dots, ord_{c_k} p)$, $l_R | c$
 - d) In addition, if X is defined by the matrix T and $\tau(T) = 0$ then the number of connected components of $STD(X)$ is divisible by p .

Future Directions

Much work remains on both the theoretical analysis and the applications of cellular automata. While much of this work will utilize non-additive automata, there are still many open questions on additive cellular automata as well. Several theoretical questions revolve around the issue of surjectivity. While all additive cellular automata are surjective (bracketing the spurious Garden-of-Eden states that exist for rules defined on finite state spaces) and for general rules surjectivity in dimensions higher than one is undecidable, it is desirable to have a simple general criteria for surjectivity in one dimension. It would also be useful to know if any surjective cellular automaton (additive or non-additive) is capable of universal computation. If there are surjective rules that are universal computers this would connect to work in mathematical logic on the computational limits of formal systems [133]. This connec-

tion arises because surjective rules only exhibit Garden-of-Eden states when acting on finite state spaces. The appearance of predecessor states when the state space is enlarged seems analogous to the increase in computational power of a formal system when it is enlarged by the addition of a new axiom.

Section “*d*-Dimensional Rules” introduced the discrete baker transformation of additive cellular automata rules. This transformation is a linear operator on the space of additive rules and further research into its properties can contribute to a deeper understanding of the structure of additive rules. In addition, there appears to be a connection between certain universal numbers that can be defined from this transformation and the well-known Mersenne and Fermat numbers. Elucidation of this connection would provide a significant link to number-theoretic aspects of cellular automata.

When applications are considered, [52] indicates many avenues of continued development for additive cellular automata. In addition, many of the references in the general bibliography point to directions of current cellular automata research in a number of areas. As indicated in the introduction, applications in physics (crystal growth, hydrodynamics, reaction-diffusion systems, astronomy), medicine and biology (pattern formation, genetic interaction networks, disease modeling, ecosystem modeling), pattern recognition and image processing, and computation (random number generation, language and pattern recognition, test pattern generation for VLSI chips, signature analysis, error correcting codes, cryptography) are only a small portion of the cellular automata applications that continue to be studied.

Bibliography

This article provides a brief survey of some of the significant theoretical results on additive cellular automata, together with references to applications of both cellular automata in general and additive cellular automata in particular. For historical information, references [1,2,3,5,7] are recommended. References [4,6,8,9] provide a general background in the use of cellular automata in modeling, as well as a number of examples. Specific exemplary cases of applications are found in [10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72]. [52], which deals extensively with the use of additive cellular automata in computing applications and VLSI chip design, is of particular value. [82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99,100,101] provide a good background in

the relation between cellular automata and fractal patterns. [81,102,103,104,105,106,107,108,109,110,111,112,113,114,115,116,117,118,119,120,121,122,123,124,125,126,127,128,129] deal with the theoretical analysis of additive cellular automata. A good survey of work in cellular automata up to the mid-1990s is [131]. An important computational survey of cellular automata dynamics is given in [132].

Primary Literature

1. von Neumann J (1963) The general and logical theory of automata. In: Taub A (ed) J. von Neumann Collected Works, vol 5. Pergamon, NY, pp 288–328
2. von Neumann J, Burk AW (ed) (1966) Theory of Self-Reproducing Automata. University of Illinois Press, Urbana
3. Arbib M (1966) Simple self-reproducing universal automata. *Inf Control* 9:177–189
4. Toffoli T, Margolis N (1987) Cellular Automata Machines: A New Environment for Modeling. MIT Press, Cambridge
5. Codd EF (1968) Cellular Automata. Academic Press, NY
6. Duff MJB, Preston K, Jr. (1984) Modern Cellular Automata: Theory and Applications. Plenum, NY
7. Sarkar P (2000) A brief history of cellular automata. *ACM Comput Syst* 32(1):80–107
8. Chopard B, Droz M (1998) Cellular Automata Modelling of Physical Systems. Cambridge University Press, Cambridge
9. Lindenmayer A, Rozenberg G (1976) Automata, Languages, Development. North Holland, Amsterdam
10. Mackay AL (1976) Crystal Symmetry. *Phys Bull* 27:495–497
11. Langer JS (1980) Instabilities and pattern formation in crystal growth. *Rev Mod Phys* 52:1
12. Lin F, Goldenfeld N (1990) Generic features of late-stage crystal growth. *Phys Rev A* 42:895–903
13. Greenberg JM, Hastings SP (1978) Spatial patterns for discrete models of diffusion in excitable media. *SIAM J Appl Math* 34(3):515–523
14. Greenberg JM, Hassard BD, Hastings SP (1978) Pattern formation and periodic structures in systems modeled by reaction-diffusion equations. *Bull Am Math Soc* 84:1296–1327
15. Madore BF, Freedman WL (1983) Computer simulations of the Belousov-Zhabotinsky reaction. *Science* 222:615–616
16. Adamatzky A, Costello BDL, Asai T (2005) Reaction-Diffusion Computers. Elsevier, London
17. Oono Y, Kohmoto M (1985) A discrete model for chemical turbulence. *Phys Rev Lett* 55:2927–2931
18. Falk H (1986) Comments on a simple cellular automata in spin representation. *Physica D* 20:447–449
19. Canning A, Droz M (1991) A comparison of spin exchange and cellular automata models for diffusion-controlled reactions. In: Gutowitz HA (ed) Cellular Automata: Theory and Experiment. MIT Press, Cambridge, pp 285–292
20. Vitanni P (1973) Sexually reproducing cellular automata. *Math Biosci* 18:23–54
21. Young D (1984) A local activator-inhibitor model of vertebrate skin patterns. *Math Biosci* 72:51–58
22. Dutching W, Vogelsaenger T (1985) Recent progress in modeling and simulation of three dimensional tumor growth and treatment. *Biosystems* 18(1):79–104
23. Moreira J, Deutsch A (2002) Cellular automata models of tumor development: A critical review. *Adv Complex Syst* 5(2&3):247–269
24. Sieburg HB, McCutchan JA, Clay OK, Cabalero L, Ostlund JJ (1991) Simulation on HIV infection in artificial immune systems. In: Gutowitz HA (ed) Cellular Automata: Theory and Experiment. MIT Press, Cambridge, pp 208–227
25. Santos RMZD, Continho S (2001) Dynamics of HIAV approach: A cellular automata approach. *Phys Rev Lett* 87(16):102–104
26. Beauchemin C, Samuel J, Tuszynski J (2005) A simple cellular automaton model for influenza A viral infection. *J Theor Biol* 232(2):223–234
27. Burks C, Farmer D (1984) Towards modeling DNA sequences as automata. *Physica D* 10:157–167
28. Moore JH, Hahn LW (2002) Cellular automata and genetic algorithms for parallel problem solving in human genetics. In: Merelo JJ, Panagiotis A, Beyer H-G (eds) Lecture Notes in Computer Science. Springer, Berlin, pp 821–830
29. Gerola H, Seiden P (1978) Stochastic star formation and spiral structure of galaxies. *Astrophys J* 223:129–135
30. Flache A, Hegselmann R (1998) Understanding complex social dynamics: A plea for cellular automata based modeling. *J Artif Soc Social Simul* 1(3):1
31. Chen K, Bak P, Jensen MH (1990) A deterministic critical forest-fire model. *Phys Lett A* 149:207–210
32. Drossel B, Schwabl F (1992) Self-organized critical forest-fire model. *Phys Rev Lett* 69:1629–1632
33. Smith III AR (1972) Real-time language recognition by one-dimensional cellular automata. *J Comput Syst Sci* 6:233–253
34. Sommerhalder R, van Westrhenen SC (1983) Parallel language recognition in constant time by cellular automata. *Acta Inform* 19:397–407
35. Ibarra OH, Palis MA, Kim SM (1985) Fast parallel language recognition by cellular automata. *Theor Comput Sci* 41:231–246
36. Morita K, Ueno S (1994) Parallel generation and parsing of array languages using reversible cellular automata. *Int J Pattern Recognit Artif Intell* 8:543–561
37. Jen E (1986) Invariant strings and pattern recognizing properties of 1D CA. *J Stat Phys* 43:243–265
38. Raghavan R (1993) Cellular automata in pattern recognition. *Inf Sci* 70:145–177
39. Chattopadhyay S, Adhikari S, Sengupta S, Pal M (2000) Highly regular, modular, and cascable design of cellular automata-based pattern classifier. *IEEE Trans VLSI Syst* 8(6):724–735
40. Rosenfeld A (1979) Picture Languages. Academic Press, NY
41. Sternberg SR (1980) Language and architecture for parallel image processing. In: Gelesma ES, Kanal LN (eds) Pattern Recognition in Practice. North-Holland, Amsterdam, p 35
42. Hopcroft JE, Ullman JD (1972) Introduction to Automata Theory, Language, and Computation. Addison-Wesley, Reading
43. Cole SN (1969) Real time computation by n-dimensional iterative arrays of finite state machines. *IEEE Trans Comput* C-18:349
44. Benjamin SC, Johnson NF (1997) A possible nanometer-scale computing device based on an additive cellular automata. *Appl Phys Lett* 70(17):2321–2323
45. Carter F (1984) The molecular device computer: point of departure for large scale cellular automata. *Physica D* 10:175–194

46. Hillis WD (1984) The connection machine: A computer architecture based on cellular automata. *Physica D* 10:213–228
47. Manning FB (1977) An approach to highly integrated computer-maintained cellular arrays. *IEEE Trans Comput* C-26:536
48. Atrubin AJ (1965) A one-dimensional real time iterative multiplier. *IEEE Trans Comput* EC-14:394
49. Nishio H (1981) Real time sorting of binary numbers by a 1-dimensional cellular automata. *Kyoto University Technical Report*
50. Fischer PC (1965) Generation of primes by a one-dimensional real time iterative array. *J Assoc Comput Machin* 12:388
51. Pries W, Thanailakis A, Card HC (1986) Group properties of cellular automata and VLSI applications. *IEEE Trans Comput* C-35:1013–1024
52. Chaudhuri PP, Chowdhury DR, Nandi S, Chattopadhyay S (1997) *Additive Cellular Automata: Theory and Applications*, vol 1. IEEE Computer Society Press, Los Alamitos
53. Bardell PH, McAnney WH (1986) Pseudo-random arrays for built-in tests. *IEEE Trans Comput* C-35(7):653–658
54. Hortensius PD, McLeod RD, Card HC (1989) Parallel random number generation for VLSI systems using cellular automata. *IEEE Trans Comput* 38(10):1466–1473
55. Tsalides P, York TA, Thanailakis A (1991) Pseudorandom number generation for VLSI systems using cellular automata. *IEEE Proc E: Comput Digit Technol* 138:241–249
56. Matsumoto M (1998) Simple cellular automata as pseudorandom m-sequence generators for built-in self-test. *ACM Trans Modeling Comput Simul (TOMACS)* 8(1):31–42
57. Tomassini M, Sipper M, Perrenoud M (2000) On the generation of high-quality random numbers by two-dimensional cellular automata. *IEEE Trans Comput* 49(10):1146–1151
58. Das AK, Chaudhuri PP (1989) An efficient on-chip deterministic test pattern generation scheme. *Euromicro J, Microprocess, Microprogramm* 26:195–204
59. Serra M (1990) Algebraic analysis and algorithms for linear cellular automata over GF(2) and their applications to digital circuit testing. *Congressus Numerantium* 75:127–139
60. Das AK, Chaudhuri PP (1993) Vector space theoretical analysis of additive cellular automata and its applications for pseudo-exhaustive test pattern generation. *IEEE Trans Comput* 42:340–352
61. Tziones P, Tsalides P, Thanailakis A (1994) A new cellular automaton-based nearest neighbor pattern classifier and its VLSI implementation. *IEEE Trans VLSI Implement* 2(3):343–353
62. Mrugalski G, Rajski J, Tyszer J (2000) Cellular automata-based test pattern generators with phase shifters. *IEEE Trans Comput-Aided Des* 19(8):878–893
63. Sikdar BK, Ganguly N, Chaudhuri PP (2002) Design of hierarchical cellular automata for on-chip test pattern generator. *IEEE Trans Comput Assist Des* 21(12):1530–1539
64. Hortensius PD, McLeod RD, Card HC (1990) Cellular automata based signature analysis for built-in self-test. *IEEE Trans Comput* C-39:1273–1283
65. Serra M, Slater T, Muzio JC, Miller DM (1990) Analysis of one-dimensional cellular automata and their aliasing probabilities. *IEEE Trans Comput-Aided Des* 9:767–778
66. Dasgupta P, Chattopadhyay S, Sengupta I (2001) Theory and application of nongroup cellular automata for message authentication. *J Syst Architecture* 47(7):383–404
67. Chowdhury DR, Basu S, Gupta IS, Chaudhuri PP (1994) Design of CAECC cellular automata based error correcting code. *IEEE Trans Comput* 43:759–764
68. Chowdhury DR, Gupta IS, Chaudhuri PP (1995) Cellular automata based byte error correcting code. *IEEE Trans Comput* 44:371–382
69. Chowdhury DR, Gupta IS, Chaudhuri PP (1995) A low-cost high-capacity associative memory design using cellular automata. *IEEE Trans Comput* 44:1260–1264
70. Nandi S, Kar BK, Chaudhuri PP (1994) Theory and application of cellular automata in cryptography. *IEEE Trans Comput* 43(12):1346–1357
71. Cattell K, Muzio JC (1996) Synthesis of one-dimensional linear hybrid cellular automata. *IEEE Trans Comput-Aided Des* 15:325–335
72. Cattell K, Zhang S, Serra M, Zmuzio JC (1999) 2-by-n hybrid cellular automata with regular configuration: Theory and applications. *IEEE Trans Comput* 48(3):285–295
73. Hedlund GA (1969) Endomorphisms and automorphisms of the shift dynamical system. *Math Syst Theory* 4:320–375
74. Wolfram S (1983) Statistical mechanics of cellular automata. *Rev Mod Phys* 55:601–644
75. Gardner M (1970) The fantastic combinations of John Conway's new solitaire game 'life'. *Sci Am* 223:120–123
76. Gardner M (1971) On cellular automata self-reproduction, the Garden of Eden and the game of 'life'. *Sci Am* 224:112–117
77. Jen E (1988) Cylindrical cellular automata. *Commun Math Phys* 118:569–590
78. Tadaki S, Matsufuji S (1993) Periodicity in one-dimensional finite linear cellular automata. *Prog Theor Phys* 89(2):325–331
79. Nandi S, Pal Chaudhuri P (1996) Analysis of periodic and intermediate boundary 90/150 cellular automata. *IEEE Trans Comput* 45(1):1–12
80. Chin W, Cortzen B, Goldman J (2001) Linear cellular automata with boundary conditions. *Linear Algebra Appl* 322:193–206
81. Voorhees BH (1996) *Computational Analysis of One Dimensional Cellular Automata*. World Scientific, Singapore
82. Willson S (1984) Cellular automata can generate fractals. *Discret Appl Math* 8:91–99
83. Willson S (1984) Growth rates and fractional dimension in cellular automata. *Physica D* 10:69–74
84. Peitgen H-O, Richter PH (1986) *The Beauty of Fractals: Images of Complex Dynamical Systems*. Springer, NY
85. Willson S (1987) The equality of fractional dimension for certain cellular automata. *Physica D* 24:179–189
86. Willson S (1987) Computing fractional dimension of additive cellular automata. *Physica D* 24:190–206
87. Culik II K, Dube S (1989) Fractals and recurrent behavior of cellular automata. *Complex Syst* 3:253–267
88. Willson S (1992) Calculating growth rates and moments for additive cellular automata. *Discret Appl Math* 35:47–65
89. Voorhees B (1988) Cellular automata, Pascal's triangle, and generation of order. *Physica D* 31:135–140
90. von Haeseler F, Peitgen H-O, Skordev G (1992) Pascal's triangle, dynamical systems and attractors. *Ergod Theory Dyn Syst* 12:479–486
91. Allouche JP, von Haeseler F, Peitgen H-O, Skordev G (1996) Linear cellular automata, finite automata and Pascal's triangle. *Discret Appl Math* 66:1–22
92. von Haeseler F, Peitgen H-O, Skordev G (1992) Linear cellular automata, substitutions, hierarchical iterated function sys-

- tems. In: Encarnacao JL, Peitgen H-O, Sakas G, Englert G (eds) *Fractal Geometry and Computer Graphics*. Springer, NY
93. von Haeseler F, Peitgen H-O, Skordev G (1993) Cellular automata, matrix substitution, and fractals. *Ann Math Artif Intell* 8(3,4):345–362
 94. von Haeseler F, Peitgen H-O, Skordev G (1995) Global analysis of self-similar features of cellular automata: selected examples. *Physica D* 86:64–80
 95. Barbé A, von Haeseler F, Peitgen H-O, Skordev G (1995) Course-graining invariant patterns of one-dimensional two-state linear cellular automata. *Int J Bifurc Chaos* 5:1611–1631
 96. von Haeseler F, Peitgen H-O, Skordev G (2001) Self-similar structures of rescaled evolution sets of cellular automata I. *Int J Bifurc Chaos* 11(4):913–926
 97. Nagler J, Clausen JC (2005) $1/f^\alpha$ spectra in elementary cellular automata and fractal signals. *Phys Rev E* 71:067103
 98. von Haeseler F, Peitgen H-O, Skordev G (2001) Self-similar structures of rescaled evolution sets of cellular automata II. *Int J Bifurc Chaos* 11(4):927–941
 99. Barbé A, von Haeseler F, Peitgen H-O, Skordev G (2003) Rescaled evolution sets of linear cellular automata on a cylinder. *Int J Bifurc Chaos* 13(4):815–842
 100. Takahashi S (1990) Cellular automata and multifractals: Dimension spectra of linear cellular automata. *Physica D* 45:36–48
 101. Takahashi S (1992) Self-similarity of linear cellular automata. *J Comput Syst Sci* 44(1):114–140
 102. Martin O, Odlyzko A, Wolfram S (1984) Algebraic properties of cellular automata. *Commun Math Phys* 93:219–258
 103. Jen E (1986) Global properties of cellular automata. *J Stat Phys* 43(1/2):219–242
 104. Jen E (1988) Linear cellular automata and recurring sequences in finite fields. *Commun Math Phys* 119:13–28
 105. Guan P, He Y (1986) Exact results for deterministic cellular automata with additive rules. *J Stat Phys* 43(3/4):463–478
 106. Das AK, Sanyal A, Chaudhuri PP (1992) On characterization of cellular automata with matrix algebra. *Inf Sci* 61:251–277
 107. Tadaki S (1994) Orbits in one-dimensional finite linear cellular automata. *Phys Rev E* 49(2):1168–1173
 108. Voorhees B (2008) Representations of rule 90 and related rules for periodic, null, and half-infinite boundary conditions. *J Cell Autom* 3(1):1–25
 109. Davis PJ (1979) *Circulant Matrices*. Wiley-Interscience, NY
 110. Culik II K (1987) On invertible cellular automata. *Complex Syst* 1:1035–1044
 111. Toffoli T, Margolis N (1990) Invertible cellular automata: A review. *Physica D* 45:229–253
 112. Kari J (1990) Reversibility of 2D cellular is undecidable. *Physica D* 45:379–385
 113. Sutner K (1991) De Bruijn graphs and linear cellular automata. *Complex Syst* 5(1):19–30
 114. Morita K (1994) Reversible cellular automata. *J Inf Process Soc Japan* 35:315–321
 115. Moraal H (2000) Graph-theoretical characterization of invertible cellular automata. *Physica D* 141:1–18
 116. Voorhees B (1994) A note on injectivity of additive cellular automata. *Complex Syst* 8(3):151–159
 117. Lind DA (1984) Applications of ergodic theory and sofic systems to cellular automata. *Physica D* 10:36–44
 118. Jen E (1989) Limit cycles in one-dimensional cellular automata. In: Stein DL (ed) *Lectures in the Sciences of Complexity*. Addison Wesley, Reading
 119. Elspas B (1959) The theory of autonomous linear sequential networks. *TRE Trans Circuits* CT-6:45–60
 120. Stevens JG, Rosensweig RE, Cerkanowicz AE (1993) Transient and cyclic behavior of cellular automata with null boundary conditions. *J Stat Phys* 73(1,2):159–174
 121. Stevens JG (1999) On the construction of state diagrams for cellular automata with additive rules. *Inf Sci* 115:43–59
 122. Thomas DM, Stevens JG, Letteiri S (2006) Characteristic and minimal polynomials of linear cellular automata. *Rocky Mountain J Math* 36(3):1077–1092
 123. Sutner K (1988) On σ -automata. *Complex Syst* 2(1):1–28
 124. Sutner K (1988) Additive automata on graphs. *Complex Syst* 2:649–661
 125. Sutner K (2000) Sigma-automata and Chebyshev polynomials. *Theor Comput Sci* 230:49–73
 126. Sutner K (2001) Decomposition of additive CA. *Complex Syst* 13(2):245–270
 127. Lidl R, Pilz G (1984) *Applied Abstract Algebra*. Springer, NY
 128. Voorhees B (1993) Predecessors of cellular automata states: I. Additive automata. *Physica D* 68:283–292
 129. Bulitko V, Voorhees B, Bulitko V (2006) Discrete baker transformation for linear cellular automata analysis. *J Cell Autom* 1:40–70
 130. Arnold V, Aviz A (1968) *Ergodic Problems of Classical Mechanics*. Benjamin, NY
 131. Illichinsky A (2001) *Cellular Automata: A Discrete Universe*. World Scientific, Singapore
 132. Wuensche A, Lesser M (1992) *The Global Dynamics of Cellular Automata*. Addison Wesley, Reading
 133. Chaitin G (2006) *Meta Math! Vintage*, NY

Books and Reviews

- Wolfram S (1994) *Cellular Automata and Complexity*. Addison Wesley, Reading
- Wolfram S (2002) *A New Kind of Science*. Wolfram Media, Champaign
- Legendi T (1987) *Parallel Processing by Cellular Automata and Arrays*. Kluwer, Dordrecht
- Rietman E (1989) *Exploring the Geometry of Nature: Computer Modeling of Chaos, Fractals, Cellular Automata, and Neural Networks*. McGraw-Hill, NY
- Manneville P, Boccara N, Vishniac GY, Bidaux R (1990) *Cellular Automata and Modeling of Complex Physical Systems*. Springer, NY
- Goles E, Martinez S (eds) (1992) *Statistical Physics, Automata Networks and Dynamical Systems*. Kluwer, Dordrecht
- Boccara N, Goles E, Martinez S (1993) *Cellular Automata and Cooperative Systems*. Kluwer, Dordrecht
- Goles E, Martinez S (1994) *Cellular Automata, Dynamical Systems and Neural Networks*. Kluwer, Dordrecht
- Adamatzky A (1994) *Identification of Cellular Automata*. Taylor & Francis, London
- Perdang JM, Lejeune A (eds) (1994) *Cellular Automata: Prospects in Astronomy and Astrophysics*. World Scientific, Singapore
- Goles E, Martinez S (1996) *Dynamics of Complex Interacting Systems*. Kluwer, Dordrecht
- Sipper M (1997) *Evolution of Parallel Cellular Machines: The Cellular Programming Approach*. Springer, NY

- Delorme M, Mazoyer J (eds) (1998) Cellular Automata: A Parallel Model. Kluwer, Dordrecht
- Goles E, Martinez S (eds) (1999) Cellular Automata and Complex Systems. Kluwer, Dordrecht
- Crutchfield JP, Hanson JE (1999) Computational Mechanics of Cellular Processes. University of California Press, Berkeley
- Wolf-Gladrow DA (2000) Lattice-Gas Cellular Automata and Lattice Boltzmann Models. Springer, NY
- Lafe O (2000) Cellular Automata Transformations: Theory and Applications in Multimedia Compression, Encryption and Modeling. Kluwer, Dordrecht
- Yang T (2001) Cellular Image Processing. Nova, NY
- Griffeath D, Moore C (eds) (2003) New Constructions in Cellular Automata. Oxford University Press, NY
- Deutsch A, Dormann S (2004) Cellular Automata Modeling of Biological Pattern Formation: Characterization, Applications, and Analysis. Springer, Berlin
- Amos M (2004) Cellular Computing. Oxford University Press, NY
- Rothman DH, Zaleski S (2004) Lattice-Gas Cellular Automata: Simple Models of Complex Hydrodynamics. Cambridge University Press, Cambridge
- Batty M (2005) Cities and Complexity: Understanding Cities with Cellular Automata, Agent-Based Models, and Fractals. MIT Press, Cambridge
- Kier LB, Seybold PG, Cheng C-K (2005) Cellular Automata Modeling of Chemical Systems: A Testbook and Laboratory Manual. Springer, NY
- Schiff JL (2007) Cellular Automata: A Discrete View. Wiley-Interscience, NY
- Gros C (2007) Complex and Adaptive Dynamical Systems: A Primer. Springer, NY
- Bandini S, Moroni L (eds) (1996) ACRI '96: Proceedings of the 2nd International Conference on Cellular Automata for Research and Industry. Springer, NY
- Bandini S, Serra R, Liverani FS (eds) (1998) Cellular Automata: Research Towards Industry, ACRI '98: Proceedings of the 3rd International Conference on Cellular Automata for Research and Industry. Springer, NY
- Bandini S, Worsch T (eds) (2000) Theory and Practical Issues on Cellular Automata: Proceedings of the 4th International Conference on Cellular Automata for Research and Industry. Springer, NY
- Tomassini M, Chopard B (eds) (2002) Cellular Automata: Proceedings of the 5th International Conference on Cellular Automata for Research and Industry. Springer, NY
- Sloot PMA, Chopard B, Hoekstra AG (eds) (2004) Cellular Automata: Proceedings of the 6th International Conference on Cellular Automata for Research and Industry. Springer, NY
- El Yacoubi S, Chopard B, Bandini S (eds) (2006) Cellular Automata: Proceedings of the 7th International Conference on Cellular Automata for Research and Industry. Springer, NY

Websites

- <http://cell-auto.com/links/> Gives many links to other sites on cellular automata
- <http://www.theory.org/complexity/cdpt/html/node4.html> Provides reviews of theoretical aspects of cellular automata
- <http://www.ddlab.com> An excellent site; it provides access to the Discrete Dynamics Lab program, a valuable asset in work on cellular automata and random Boolean networks

<http://cellular.ci.ulsu.mx> Provides access to a number of worthwhile unpublished papers and a number of useful references

Adomian Decomposition Method Applied to Non-linear Evolution Equations in Soliton Theory

ABDUL-MAJID WAZWAZ

Department of Mathematics, Saint Xavier University, Chicago, USA

Article Outline

Glossary
 Definition of the Subject
 Introduction
 Adomian Decomposition Method and Adomian Polynomials
 Modified Decomposition Method and Noise Terms Phenomenon
 Solitons, Peakons, Kinks, and Compactons
 Solitons of the KdV Equation
 Kinks of the Burgers Equation
 Peakons of the Camassa–Holm Equation
 Compactons of the $K(n,n)$ Equation
 Future Directions
 Bibliography

Glossary

Solitons Solitons appear as a result of a balance between a weakly nonlinear convection and a linear dispersion. The solitons are localized highly stable waves that retains its identity: shape and speed, upon interaction, and resemble particle like behavior. In the case of a collision, solitons undergo a phase shift.

Types of solitons *Solitary waves*, which are localized traveling waves, are asymptotically zero at large distances and appear in many structures such as solitons, kinks, peakons, cuspons, and compactons, among others. *Solitons* appear as a bell-shaped sech profile. *Kink waves* rise or descend from one asymptotic state to another. *Peakons* are peaked solitary-wave solutions. *Cuspons* exhibit cusps at their crests. In the peakon structure, the traveling-wave solutions are smooth except for a peak at a corner of its crest. Peakons are the points at which spatial derivative changes sign so that peakons have a finite jump in the first derivative of the solution $u(x, t)$. Unlike peakons, where the derivatives

at the peak differ only by a sign, the derivatives at the jump of a cusp diverge.

Compactons are solitons with compact spatial support such that each compacton is a soliton confined to a finite core or a soliton without exponential tails. Compactons are generated due to the delicate interaction between the effect of the genuine nonlinear convection and the genuinely nonlinear dispersion.

Adomian method The Adomian decomposition method approaches linear and nonlinear, and homogeneous and inhomogeneous differential and integral equations in a unified way. The method provides the solution in a rapidly convergent series with terms that are elegantly determined in a recursive manner. The method can be used to obtain closed-form solutions, if such solutions exist. A truncated number of the obtained series can be used for numerical purposes. The method was modified to accelerate the computational process. The noise terms phenomenon, which may appear for inhomogeneous cases, can give the exact solution in two iterations only.

Definition of the Subject

Nonlinear phenomena play a significant role in many branches of applied sciences such as applied mathematics, physics, biology, chemistry, astronomy, plasma, and fluid dynamics. Nonlinear dispersive equations that govern these phenomena have the genuine soliton property. Solitons are pulses that propagate without any change of its identity, i. e., shape and speed, during their travel through a nonlinear dispersive medium [1,5,34]. Solitons resemble properties of a particle, hence the suffix *on* is used [19,20].

Solitons exist in many scientific branches, such as optical fiber photonics, fiber lasers, plasmas, molecular systems, laser pulses propagating in solids, liquid crystals, nonlinear optics, cosmology, and condensed-matter physics. Based on its importance in many fields, a huge amount of research work has been conducted during the last four decades to make more progress in understanding the soliton phenomenon. A variety of very powerful algorithms has been used to achieve this goal. The Adomian decomposition method introduced in [2,6,21,22,23,24,25,26], which will be used in this work, is one of the reliable methods that has been used recently.

Introduction

The aim of this work is to apply the Adomian decomposition method to derive specific types of soliton solutions. Solitons were discovered *experimentally* by John

Scott Russell in 1844. Korteweg and de Vries in 1895 investigated *analytically* the soliton concept [10], where they derived the pioneer equation of solitons, well known as the KdV equation, that models the height of the surface of shallow water in the presence of solitary waves. Moreover, Zabusky and Kruskal [35] investigated this phenomenon *analytically* in 1965. Since 1965, a huge number of research works have been conducted on nonlinear dispersive and dissipative equations. The aim of these works has been to study thoroughly the characteristics of soliton solutions and to study various types of solitons that appear as a result of these equations. Several reliable methods were used in the literature to handle nonlinear dispersive equations. Hirota's bilinear method [8,9] has been used for single- and multiple-soliton solutions. The inverse scattering method [1] has been widely used. For single-soliton solutions, several methods, such as the tanh method [13,14,15], the pseudospectral method, and the truncated Painlevé expansion, are used.

However, in this work, the decomposition method, introduced by Adomian in 1984, will be applied to derive the desired types of soliton solutions. The method approaches all problems in a unified way and in a straightforward manner. The method computes the solution in a fast convergent series with components that are elegantly determined. Unlike other methods, the initial or boundary conditions are necessary for the use of the Adomian method.

Adomian Decomposition Method and Adomian Polynomials

The Adomian decomposition method, developed by George Adomian in 1984, has been receiving much attention in applied mathematics in general and in the area of initial value and boundary value problems in particular. Moreover, it is also used in the area of series solutions for numerical purposes. The method is powerful and effective and can be used in linear or nonlinear, ordinary or partial differential equations, and integral equations. The decomposition method demonstrates fast convergence of the solution and provides numerical approximation with a high level of accuracy. The method handles applied problems directly and in a straightforward manner without using linearization, perturbation, or any other restrictive assumption that might change the physical behavior of the physical model under investigation.

The method is effectively addressed and thoroughly used by many researchers in the literature [1,21,22,23,24,25,26]. It is important to indicate that well-known methods, such as Backlund transformation, inverse scattering method, Hirota's bilinear formalism, the tanh method, and

many others, can handle problems without using initial or boundary value conditions. For the Adomian decomposition method, the initial or boundary conditions are necessary to conduct the determination of the components recursively. However, some of the standard methods require huge calculations, whereas the Adomian method minimizes the volume of computational work.

The Adomian decomposition method consists in decomposing the unknown function $u(x, t)$ of any equation into a sum of an infinite number of components given by the decomposition series

$$u(x, t) = \sum_{n=0}^{\infty} u_n(x, t), \quad (1)$$

where the components $u_n(x, t)$, $n \geq 0$ are to be determined in a recursive manner. The decomposition method concerns itself with the determination of the components $u_0(x, t)$, $u_1(x, t)$, $u_2(x, t)$, \dots individually. The determination of these components can be obtained through a recursive relation that usually involves evaluation of simple integrals.

We now give a clear overview of Adomian decomposition method. Consider the linear differential equation written in an operator form by

$$Lu + Ru = f, \quad (2)$$

where L is the lower-order derivative, which is assumed to be invertible, R is a linear differential operator of order greater than L , and f is a source term. We next apply the inverse operator L^{-1} to both sides of Eq. (2) and use the given initial or boundary condition to get

$$u(x, t) = g - L^{-1}(Ru), \quad (3)$$

where the function g represents the terms arising from integrating the source term f and from using the given conditions; all are assumed to be prescribed. Substituting the infinite series of components

$$u(x, t) = \sum_{n=0}^{\infty} u_n(x, t) \quad (4)$$

into both sides of (3) yields

$$\sum_{n=0}^{\infty} u_n = g - L^{-1} \left(R \left(\sum_{n=0}^{\infty} u_n \right) \right). \quad (5)$$

The decomposition method suggests that the zeroth component u_0 is usually defined by all terms not included under the inverse operator L^{-1} , which arise from the initial

data and from integrating the inhomogeneous term. This in turn gives the formal recursive relation

$$\begin{aligned} u_0(x, t) &= g, \\ u_{k+1}(x, t) &= -L^{-1}(R(u_k)), \quad k \geq 0, \end{aligned} \quad (6)$$

or, equivalently,

$$\begin{aligned} u_0(x, t) &= g, \\ u_1(x, t) &= -L^{-1}(R(u_0)), \\ u_2(x, t) &= -L^{-1}(R(u_1)), \\ u_3(x, t) &= -L^{-1}(R(u_2)), \\ &\vdots \end{aligned} \quad (7)$$

The differential equation under consideration is now reduced to integrals that can be easily evaluated. Having determined these components $u_0(x, t)$, $u_1(x, t)$, $u_2(x, t)$, \dots , we then substitute the obtained components into (4) to obtain the solution in a series form. The determined series may converge very rapidly to a closed-form solution, if an exact solution exists. For concrete problems, where a closed-form solution is not obtainable, a truncated number of terms is usually used for numerical purposes. Few terms of the truncated series give an approximation with a high degree of accuracy. The convergence concept of the decomposition series is investigated thoroughly in the literature.

Several significant studies were conducted to compare the performance of the Adomian method with other methods, such as Picard's method, Taylor series method, finite differences method, perturbation techniques, and others. The conclusions emphasized the fact that the Adomian method has many advantages and requires less computational work compared to existing techniques.

The Adomian method and many others applied this method to many deterministic and stochastic problems. However, the Adomian method, like some other methods, suffers if the zeroth component $u_0(x, t) = 0$ and makes the integrand of the right side in (7) $u_1(x, t) = 0$. If the right side integrand in (7) does not vanish, if it is of a form such as e^{u_0} or $\ln(\alpha + u_0)$, $\alpha > 0$, then the method works effectively.

As stated before, the Adomian method decomposes the unknown function $u(x, t)$ into an infinite number of components. However, for nonlinear functions of $u(x, t)$ such as u^2 , u^3 , $\ln(1 + u)$, $\cos u$, e^u , uu_x , etc., a special representation for nonlinear terms was developed by Adomian and others. Adomian introduced a formal algorithm to establish the proper representation for all forms of nonlinear functions of $u(x, t)$. This representation of nonlin-

ear terms is necessary to apply the Adomian method properly. Several alternative algorithms have been introduced in the literature by researchers to calculate Adomian polynomials. However, the Adomian algorithm remains the most commonly applied because it is simple and practical; therefore, it will be used in this work.

Adomian assumed that the nonlinear term $F(u)$ can be expressed by an infinite series of the so-called Adomian polynomials A_n given in the form

$$F(u) = \sum_{n=0}^{\infty} A_n(u_0, u_1, u_2, \dots, u_n), \quad (8)$$

where A_n can be evaluated for all forms of nonlinearity.

Adomian polynomials A_n for the nonlinear term $F(u)$ can be evaluated using the following expression:

$$A_n = \frac{1}{n!} \frac{d^n}{d\lambda^n} \left[F \left(\sum_{i=0}^n \lambda^i u_i \right) \right]_{\lambda=0}, \quad n = 0, 1, 2, \dots \quad (9)$$

The general formula (9) can be simplified as follows. Assuming that the nonlinear function is $F(u)$, using (9), Adomian polynomials are given by

$$\begin{aligned} A_0 &= F(u_0), \\ A_1 &= u_1 F'(u_0), \\ A_2 &= u_2 F'(u_0) + \frac{1}{2!} u_1^2 F''(u_0), \\ A_3 &= u_3 F'(u_0) + u_1 u_2 F''(u_0) + \frac{1}{3!} u_1^3 F'''(u_0), \\ A_4 &= u_4 F'(u_0) + \left(\frac{1}{2!} u_2^2 + u_1 u_3 \right) F''(u_0) \\ &\quad + \frac{1}{2!} u_1^2 u_2 F'''(u_0) + \frac{1}{4!} u_1^4 F^{(iv)}(u_0), \\ A_5 &= u_5 F'(u_0) + (u_2 u_3 + u_1 u_4) F''(u_0) \\ &\quad + \left(\frac{1}{2!} u_1 u_2^2 + \frac{1}{2!} u_1^2 u_3 \right) F'''(u_0) \\ &\quad + \frac{1}{3!} u_1^3 u_2 F^{(iv)}(u_0) + \frac{1}{5!} u_1^5 F^{(v)}(u_0). \end{aligned} \quad (10)$$

Other polynomials can be generated in a similar manner. It is clear that A_0 depends only on u_0 , A_1 depends only on u_0 and u_1 , A_2 depends only on u_0 , u_1 , and u_2 , and so on.

For $F(u) = u^2$, we find

$$\begin{aligned} A_0 &= F(u_0) = u_0^2, \\ A_1 &= u_1 F'(u_0) = 2u_0 u_1, \\ A_2 &= u_2 F'(u_0) + \frac{1}{2!} u_1^2 F''(u_0) = 2u_0 u_2 + u_1^2, \\ A_3 &= u_3 F'(u_0) + u_1 u_2 F''(u_0) + \frac{1}{3!} u_1^3 F'''(u_0) \\ &= 2u_0 u_3 + 2u_1 u_2. \end{aligned} \quad (11)$$

Modified Decomposition Method and Noise Terms Phenomenon

A reliable modification of the Adomian decomposition method [22] was developed by Wazwaz in 1999. The modification will further accelerate the convergence of the series solution. As presented earlier, the standard decomposition method admits the use of the recursive relation

$$\begin{aligned} u_0 &= g, \\ u_{k+1} &= -L^{-1}(Ru_k), \quad k \geq 0. \end{aligned} \quad (12)$$

The modified decomposition method introduces a slight variation to the recursive relation (12) that will lead to the determination of the components of u in a faster and easier way. For specific cases, the function g can be set as the sum of two partial functions, g_1 and g_2 . In other words, we can set

$$g = g_1 + g_2. \quad (13)$$

This assumption gives a slight qualitative change in the formation of the recursive relation (12). To reduce the size of calculations, we identify the zeroth component u_0 by one part of g , that is, g_1 or g_2 . The other part of g can be added to the component u_1 among other terms. In other words, the modified recursive relation can be defined as

$$\begin{aligned} u_0 &= g_1, \\ u_1 &= g_2 - L^{-1}(Ru_0), \\ u_{k+1} &= -L^{-1}(Ru_k), \quad k \geq 1. \end{aligned} \quad (14)$$

The change occurred in the formation of the first two components u_0 and u_1 only [22,24,25]. Although this variation in the formation of u_0 and u_1 is slight, it plays a major role in accelerating the convergence of the solution and in minimizing the size of calculations. It is interesting to point out that by selecting the parts g_1 and g_2 properly, the exact solution $u(x, t)$ may be obtained by using very few iterations, and sometimes by evaluating only two components. Moreover, if g consists of one term only, the standard decomposition method should be employed in this case.

Another useful feature of the Adomian method is the noise terms phenomenon. The noise terms may appear for inhomogeneous problems only. This phenomenon was addressed by Adomian in 1994. In 1997, Wazwaz investigated the necessary conditions for the appearance of noise terms in the decomposition series. **Noise terms** are defined as identical terms with opposite signs [21] that arise in the components u_0 and u_1 particularly, and in other components as well. By canceling the noise terms between u_0 and u_1 , even though u_1 contains other terms, the remaining

noncanceled terms of u_0 may give the exact solution of the equation. Therefore, it is necessary to verify that the non-canceled terms of u_0 satisfy the equation. The noise terms, if they exist in the components u_0 and u_1 , will provide the solution in a closed form with only two successive iterations.

It was formally proved by Wazwaz in 1997 that a necessary condition for the appearance of the noise terms is required. The conclusion made is that the zeroth component u_0 must contain the exact solution u , among other terms [21]. Moreover, it was shown that the nonhomogeneity condition does not always guarantee the appearance of the noise terms.

Solitons, Peakons, Kinks, and Compactons

There are many types of *solitary waves*. *Solitons*, which are localized traveling waves, are asymptotically zero at large distances [1,5,7,21,22,23,24,25,26,27,28,29,30,31,32,33]. Solitons appear as a bell-shaped sech profile. Soliton solution $u(\xi)$ results as a balance between nonlinearity and dispersion, where $u(\xi)$, $u'(\xi)$, $u''(\xi)$, $\dots \rightarrow 0$ as $\xi \rightarrow \pm\infty$, where $\xi = x - ct$, and c is the speed of the wave propagation. The soliton solution either decays exponentially as in the KdV equation, or it converges to a constant at infinity such as the kinks of the sine-Gordon equation. This means that the soliton solutions appear as sech^α or $\arctan(e^{\alpha(x-ct)})$. Moreover, one soliton interacts with other solitons preserving its permanent form.

Another type of solitary wave is the *kink wave*, which rises or descends from one asymptotic state to another [18]. The Burgers equation and the sine-Gordon equation are examples of nonlinear wave equations that exhibit kink solutions. It is to be noted that the kink $u(\xi)$ converges to $\pm\alpha$, where α is a constant. However, $u'(\xi)$, $u''(\xi)$, $\dots \rightarrow 0$ as $\xi \rightarrow \pm\infty$.

Peakons are peaked solitary-wave solutions and are another type of solitary-wave solution. In this structure, the traveling wave solutions are smooth except for a peak at a corner of its crest. Peakons are the points at which a spatial derivative changes signs so that peakons have a finite jump in the first derivative of the solution $u(x, t)$ [3,4,11,12,32]. This means that the first derivative of $u(x, t)$ has identical values with opposite signs around the peak.

A significant type of soliton is the *compacton*, which is a soliton with compact spatial support such that each compacton is a soliton confined to a finite core or a soliton without exponential tails [16,17]. Compactons were formally derived by Rosenau and Hyman in 1993 where a special kind of the KdV equation was used to derive this

new discovery. Unlike a soliton that narrows as the amplitude increases, a compacton's width is independent of the amplitude [16,17,27,28,29,30]. Classical solitons are analytic solutions, whereas compactons are nonanalytic solutions [16,17]. As will be shown by a graph below, compactons are solitons that are free of exponential wings or tails. Compactons arise as a result of the delicate interaction between genuine nonlinear terms and the genuinely nonlinear dispersion, as is the case with the K(n,n) equation.

Solitons of the KdV Equation

In 1895, Korteweg, together with his Ph.D student de Vries, derived *analytically* a nonlinear partial differential equation, well known now by the KdV equation given in its simplest form by

$$u_t + 6uu_x + u_{xxx} = 0, u(x, 0) = \frac{2c^2 e^{cx}}{(1 + e^{cx})^2}. \quad (15)$$

The KdV equation is the simplest equation embodying both nonlinearity and dispersion [5,34]. This equation has served as a model for the development of solitary-wave theory. The KdV equation is a completely integrable bi-Hamiltonian equation. The KdV equation is used to model the height of the surface of shallow water in the presence of solitary waves. The nonlinearity represented by uu_x tends to localize the wave, while the linear dispersion represented by u_{xxx} spreads it out. The balance between the weak nonlinearity and the linear dispersion gives solitons that consist of single humped waves. The equilibrium between the nonlinearity uu_x and the dispersion u_{xxx} of the KdV equation is stable.

In 1965, Zabusky and Kruskal [35] investigated *numerically* the nonlinear interaction of a large solitary wave overtaking a smaller one and discovered that solitary waves underwent nonlinear interactions following the KdV equation. Further, the waves emerged from this interaction retaining their original shape and amplitude, and therefore conserved energy and mass. The only effect of the interaction was a phase shift.

To apply the Adomian decomposition method, we first write the KdV Eq. (15) in an operator form:

$$L_t u = -u_{xxx} - 6uu_x, \quad (16)$$

where the differential operator L_t is

$$L_t = \frac{\partial}{\partial t}, \quad (17)$$

assuming that the integral operator L_t^{-1} exists and may be regarded as a onefold definite integral defined by

$$L_t^{-1}(\cdot) = \int_0^t (\cdot) dt. \quad (18)$$

This means that

$$L_t^{-1} L_t u(x, t) = u(x, t) - u(x, 0). \quad (19)$$

Applying L_t^{-1} to both sides of (16) and using the initial condition we find

$$u(x, t) = \frac{2c^2 e^{cx}}{(1 + e^{cx})^2} + L_t^{-1}(-u_{xxx} - 6uu_x). \quad (20)$$

Notice that the right-hand side contains a linear term u_{xxx} and a nonlinear term uu_x . Accordingly, Adomian polynomials for this nonlinear term are given by

$$\begin{aligned} A_0 &= F(u_0) = u_0 u_{0x}, \\ A_1 &= \frac{1}{2} L_x(2u_0 u_1) = u_{0x} u_1 + u_0 u_{1x}, \\ A_2 &= \frac{1}{2} L_x(2u_0 u_2 + u_1^2) = u_{0x} u_2 + u_{1x} u_1 + u_{2x} u_0, \\ A_3 &= \frac{1}{2} L_x(2u_0 u_3 + 2u_1 u_2) \\ &= u_{0x} u_3 + u_{1x} u_2 + u_{2x} u_1 + u_{3x} u_0. \end{aligned} \quad (21)$$

Recall that the Adomian decomposition method suggests that the linear function u may be represented by a decomposition series

$$u = \sum_{n=0}^{\infty} u_n, \quad (22)$$

whereas nonlinear term $F(u)$ can be expressed by an infinite series of the so-called Adomian polynomials A_n given in the form

$$F(u) = \sum_{n=0}^{\infty} A_n(u_0, u_1, u_2, \dots, u_n). \quad (23)$$

Using the decomposition identification for the linear and nonlinear terms in (20) yields

$$\begin{aligned} \sum_{n=0}^{\infty} u_n(x, t) &= \frac{2c^2 e^{cx}}{(1 + e^{cx})^2} - L_t^{-1} \left(\sum_{n=0}^{\infty} u_n(x, t) \right)_{xxx} \\ &\quad - 6L_t^{-1} \left(\sum_{n=0}^{\infty} A_n \right). \end{aligned} \quad (24)$$

The Adomian method allows for the use of the recursive relation

$$\begin{aligned} u_0(x, t) &= \frac{2c^2 e^{cx}}{(1 + e^{cx})^2}, \\ u_{k+1}(x, t) &= -L_t^{-1}(u_{kxxx}) - 6L_t^{-1}(A_k), \quad k \geq 0. \end{aligned} \quad (25)$$

The components u_n , $n \geq 0$ can be elegantly calculated by

$$\begin{aligned} u_0(x, t) &= \frac{2c^2 e^{cx}}{(1 + e^{cx})^2}, \\ u_1(x, t) &= -L_t^{-1}(u_{0xxx}) - 6L_t^{-1}(A_0) \\ &= \frac{2c^5 e^{cx}(e^{cx} - 1)}{(1 + e^{cx})^3} t, \\ u_2(x, t) &= -L_t^{-1}(u_{1xxx}) - 6L_t^{-1}(A_1) \\ &= \frac{c^8 e^{cx}(e^{2cx} - 4e^{cx} + 1)}{(1 + e^{cx})^4} t^2, \\ u_3(x, t) &= -L_t^{-1}(u_{2xxx}) - 6L_t^{-1}(A_2) \\ &= \frac{c^{11} e^{cx}(e^{3cx} - 11e^{2cx} + 11e^{cx} - 1)}{3(1 + e^{cx})^5} t^3, \\ u_4(x, t) &= -L_t^{-1}(u_{3xxx}) - 6L_t^{-1}(A_3) \\ &= \frac{c^{14} e^{cx}(e^{4cx} - 26e^{3cx} + 66e^{2cx} - 26e^{cx} + 1)}{12(1 + e^{cx})^6} t^4, \end{aligned} \quad (26)$$

and so on. The series solution is thus given by

$$\begin{aligned} u(x, t) &= \frac{2c^2 e^{cx}}{(1 + e^{cx})^2} \\ &\times \left(1 + \frac{c^3(e^{cx} - 1)}{(1 + e^{cx})} t + \frac{c^6(e^{2cx} - 4e^{cx} + 1)}{2(1 + e^{cx})^2} t^2 \right. \\ &\quad \left. + \frac{c^9(e^{3cx} - 11e^{2cx} + 11e^{cx} - 1)}{6(1 + e^{cx})^3} t^3 + \dots \right), \end{aligned} \quad (27)$$

and in a closed form by

$$u(x, t) = \frac{2c^2 e^{c(x-c^2 t)}}{(1 + e^{c(x-c^2 t)})^2}. \quad (28)$$

The last equation emphasizes the fact that the dispersion relation is $\omega = c^3$. Moreover, the exact solution (28) can be rewritten as

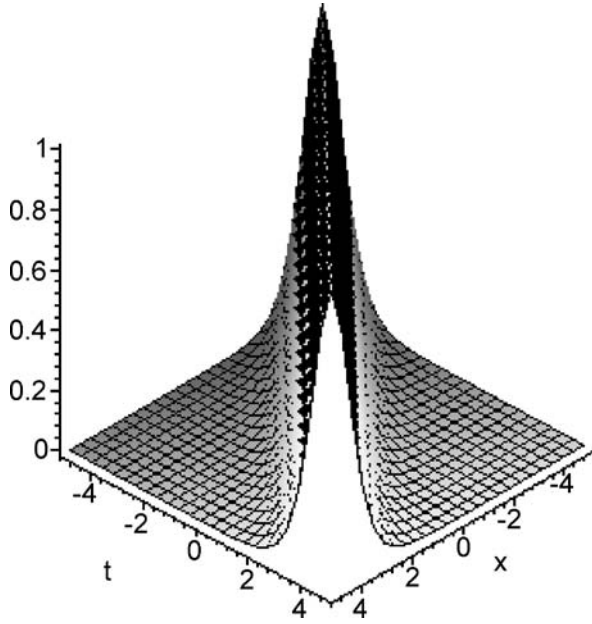
$$u(x, t) = \frac{c^2}{2} \operatorname{sech}^2 \left[\frac{c}{2} (x - c^2 t) \right]. \quad (29)$$

This in turn gives the bell-shaped soliton solution of the KdV equation. A typical graph of a bell-shaped soliton is given in Fig. 1.

The graph shows that solitons are characterized by exponential wings or tails. The graph also confirms that solitons become asymptotically zero at large distances.

Kinks of the Burgers Equation

Burgers (1895–1981) introduced one of the fundamental model equations in fluid mechanics [18] that demonstrates



Adomian Decomposition Method Applied to Non-linear Evolution Equations in Soliton Theory, Figure 1

Figure 1 shows the soliton graph $u(x, t) = \text{sech}^2(x - ct)$, $c = 1$, $-5 \leq x, t \leq 5$

coupling between nonlinear advection uu_x and linear diffusion u_{xx} . The Burgers equation appears in gas dynamics and traffic flow. Burgers introduced this equation to capture some of the features of turbulent fluid in a channel caused by the interaction of the opposite effects of convection and diffusion. The standard form of the Burgers equation is given by

$$u_t + uu_x = \nu u_{xx}, \quad (30)$$

$$u(x, 0) = \frac{2c}{(1 + e^{\frac{cx}{\nu}})}, \quad t > 0, \quad \nu \neq 0,$$

where $u(x, t)$ is the velocity and ν is a constant that defines the kinematic viscosity. If the viscosity $\nu = 0$, then the equation is called an *inviscid* Burgers equation. The inviscid Burgers equation will not be examined in this work.

It is the goal of this work to apply the Adomian decomposition method to the Burgers equation; therefore we write (30) in an operator form

$$L_t u = \nu u_{xx} - uu_x, \quad (31)$$

where the differential operator L_t is

$$L_t = \frac{\partial}{\partial t}, \quad (32)$$

and as a result

$$L_t^{-1}(\cdot) = \int_0^t (\cdot) dt. \quad (33)$$

This means that

$$L_t^{-1} L_t u(x, t) = u(x, t) - u(x, 0). \quad (34)$$

Applying L_t^{-1} to both sides of (31) and using the initial condition we find

$$u(x, t) = \frac{2c}{(1 + e^{\frac{cx}{\nu}})} + L_t^{-1}(\nu u_{xx} - uu_x). \quad (35)$$

Notice that the right-hand side contains a linear term and a nonlinear term uu_x . The Adomian polynomials for this nonlinear term uu_x are the same as in the KdV equation.

Using the decomposition identification for the linear and nonlinear terms in (35) yields

$$\sum_{n=0}^{\infty} u_n(x, t) = \frac{2c}{(1 + e^{\frac{cx}{\nu}})} + L_t^{-1} \left(\nu \sum_{n=0}^{\infty} u_n(x, t) \right)_{xx} - L_t^{-1} \left(\sum_{n=0}^{\infty} A_n \right). \quad (36)$$

The Adomian method allows for the use of the recursive relation

$$u_0(x, t) = \frac{2c}{(1 + e^{\frac{cx}{\nu}})}, \quad (37)$$

$$u_{k+1}(x, t) = L_t^{-1}(\nu u_{k,xx}) - L_t^{-1}(A_k), \quad k \geq 0.$$

The components u_n , $n \geq 0$ can be elegantly calculated by

$$u_0(x, t) = \frac{2c}{(1 + e^{\frac{cx}{\nu}})},$$

$$u_1(x, t) = L_t^{-1}(\nu u_{0,xx}) - L_t^{-1}(A_0)$$

$$= \frac{2c^3 e^{\frac{cx}{\nu}}}{\nu(1 + e^{\frac{cx}{\nu}})^2} t,$$

$$u_2(x, t) = L_t^{-1}(\nu u_{1,xx}) - L_t^{-1}(A_1)$$

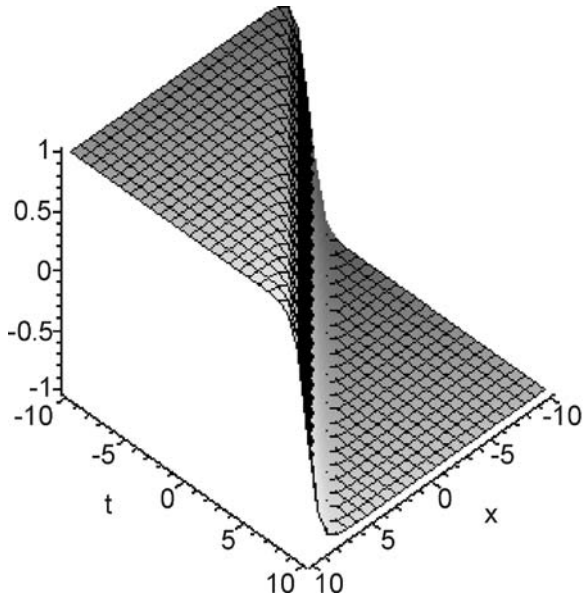
$$= \frac{c^5 e^{\frac{cx}{\nu}} (e^{\frac{cx}{\nu}} - 1)}{\nu^2 (1 + e^{\frac{cx}{\nu}})^3} t^2, \quad (38)$$

$$u_3(x, t) = L_t^{-1}(\nu u_{2,xx}) - L_t^{-1}(A_2)$$

$$= \frac{c^7 e^{\frac{cx}{\nu}} (e^{\frac{2cx}{\nu}} - 4e^{\frac{cx}{\nu}} + 1)}{3\nu^3 (1 + e^{\frac{cx}{\nu}})^4} t^3,$$

$$u_4(x, t) = L_t^{-1}(\nu u_{3,xx}) - L_t^{-1}(A_3)$$

$$= \frac{c^9 e^{\frac{cx}{\nu}} (e^{\frac{3cx}{\nu}} - 11e^{\frac{2cx}{\nu}} + 11e^{\frac{cx}{\nu}} - 1)}{12\nu^4 (1 + e^{\frac{cx}{\nu}})^5} t^4,$$



Adomian Decomposition Method Applied to Non-linear Evolution Equations in Soliton Theory, Figure 2

Figure 2 shows the kink graph $u(x, t) = \tanh(x - ct)$, $c = 1$, $-10 \leq x, t \leq 10$

and so on. The series solution is thus given by

$$u(x, t) = \frac{2ce^{\frac{cx}{v}}}{(1 + e^{\frac{cx}{v}})} \times \left(e^{-\frac{cx}{v}} + \frac{c^2}{v(1 + e^{\frac{cx}{v}})} t + \frac{c^4(e^{\frac{cx}{v}} - 1)}{2v^2(1 + e^{\frac{cx}{v}})^2} t^2 + \frac{c^6(e^{\frac{2cx}{v}} - 4e^{\frac{cx}{v}} + 1)}{6v^3(1 + e^{\frac{cx}{v}})^3} t^3 + \dots \right), \quad (39)$$

and in a closed form by

$$u(x, t) = \frac{2c}{\left(1 + e^{\frac{c}{v}(x-ct)}\right)}, \quad v \neq 0, \quad (40)$$

or equivalently

$$u(x, t) = c \left(1 - \tanh \left[\frac{c}{2v}(x - ct) \right] \right). \quad (41)$$

Figure 2 shows a kink graph.

The graph shows that the kink converges to ± 1 as $\xi \rightarrow \pm\infty$.

Peakons of the Camassa–Holm Equation

Camassa and Holm [4] derived in 1993 a completely integrable wave equation

$$u_t + 2ku_x - u_{xxt} + 3uu_x = 2u_x u_{xx} + uu_{xxx} \quad (42)$$

by retaining two terms that are usually neglected in the small-amplitude shallow-water limit [3]. The constant k is related to the critical shallow-water wave speed. This equation models the unidirectional propagation of water waves in shallow water. The CH Eq. (42) differs from the well-known regularized long wave (RLW) equation only through the two terms on the right-hand side of (42). Moreover, this equation has an integrable bi-Hamiltonian structure and arises in the context of differential geometry, where it can be seen as a reexpression for geodesic flow on an infinite-dimensional Lie group. The CH equation admits a second-order isospectral problem and allows for peaked solitary-wave solutions, called peakons.

In this work, we will apply the Adomian method to the CH equation

$$u_t - u_{xxt} + 3uu_x = 2u_x u_{xx} + uu_{xxx}, \quad u(x, 0) = ce^{-|x|}, \quad (43)$$

or equivalently

$$u_t = u_{xxt} - 3uu_x + 2u_x u_{xx} + uu_{xxx}, \quad u(x, 0) = ce^{-|x|}, \quad (44)$$

where we set $k = 0$.

Proceeding as before, the CH Eq. (44) in an operator form is as follows:

$$L_t u = u_{xxt} - 3uu_x + 2u_x u_{xx} + uu_{xxx}, \quad u(x, 0) = ce^{-|x|}, \quad (45)$$

where the differential operator L_t is as defined above. It is important to point out that the CH equation includes three nonlinear terms; therefore, we derive the following three sets of Adomian polynomials for the terms uu_x , $u_x u_{xx}$, and uu_{xxx} :

$$\begin{aligned} A_0 &= u_0 u_{0x}, \\ A_1 &= u_{0x} u_1 + u_0 u_{1x}, \\ A_2 &= u_{0x} u_2 + u_{1x} u_1 + u_{2x} u_0, \end{aligned} \quad (46)$$

$$\begin{aligned} B_0 &= u_{0x} u_{0xx}, \\ B_1 &= u_{0xx} u_{1x} + u_{0x} u_{1xx}, \\ B_2 &= u_{0xx} u_{2x} + u_{1x} u_{1xx} + u_{0x} u_{2xx}, \end{aligned} \quad (47)$$

and

$$\begin{aligned} C_0 &= u_0 u_{0xxx}, \\ C_1 &= u_{0xxx} u_1 + u_0 u_{1xxx}, \\ C_2 &= u_{0xxx} u_2 + u_{1xxx} u_1 + u_0 u_{2xxx}. \end{aligned} \quad (48)$$

Applying the inverse operator L_t^{-1} to both sides of (45) and using the initial condition we find

$$u(x, t) = ce^{-|x|} + L_t^{-1}(u_{xxt} - 3uu_x + 2u_x u_{xx} + uu_{xxx}). \quad (49)$$

Case 1. For $x > 0$, the initial condition will be $u(x, 0) = e^{-x}$. Using the decomposition series (1) and Adomian polynomials in (49) yields

$$\begin{aligned} \sum_{n=0}^{\infty} u_n(x, t) &= ce^{-x} + L_t^{-1} \left(\sum_{n=0}^{\infty} u_n(x, t) \right)_{xxt} \\ &+ L_t^{-1} \left(-3 \sum_{n=0}^{\infty} A_n + 2 \sum_{n=0}^{\infty} B_n + \sum_{n=0}^{\infty} C_n \right). \end{aligned} \quad (50)$$

The Adomian scheme allows the use of the recursive relation

$$\begin{aligned} u_0(x, t) &= ce^{-x}, \\ u_{k+1}(x, t) &= L_t^{-1}(u_k(x, t))_{xxt} \\ &+ L_t^{-1}(-3A_k + 2B_k + C_k), \quad k \geq 0. \end{aligned} \quad (51)$$

The components u_n , $n \geq 0$ can be elegantly computed by

$$\begin{aligned} u_0(x, t) &= ce^{-x}, \\ u_1(x, t) &= L_t^{-1}(u_0(x, t))_{xxt} + L_t^{-1}(-3A_0 + 2B_0 + C_0) \\ &= c^2 e^{-x} t, \\ u_2(x, t) &= L_t^{-1}(u_1(x, t))_{xxt} + L_t^{-1}(-3A_1 + 2B_1 + C_1) \\ &= \frac{1}{2!} c^3 e^{-x} t^2, \\ u_3(x, t) &= L_t^{-1}(u_2(x, t))_{xxt} + L_t^{-1}(-3A_2 + 2B_2 + C_2) \\ &= \frac{1}{3!} c^4 e^{-x} t^3, \\ u_4(x, t) &= L_t^{-1}(u_3(x, t))_{xxt} + L_t^{-1}(-3A_3 + 2B_3 + C_3) \\ &= \frac{1}{4!} c^4 e^{-x} t^4, \end{aligned} \quad (52)$$

and so on. The series solution for $x > 0$ is given by

$$\begin{aligned} u(x, t) &= ce^{-x} \\ &\times \left(1 + ct + \frac{1}{2!}(ct)^2 + \frac{1}{3!}(ct)^3 + \frac{1}{4!}(ct)^4 + \dots \right), \end{aligned} \quad (53)$$

and in a closed form by

$$u(x, t) = ce^{-(x-ct)}. \quad (54)$$

Case 2. For $x < 0$, the initial condition will be $u(x, 0) = e^x$. Proceeding as before, we obtain

$$\begin{aligned} u_0(x, t) &= ce^x, \\ u_1(x, t) &= L_t^{-1}(u_0(x, t))_{xxt} + L_t^{-1}(-3A_0 + 2B_0 + C_0) \\ &= -c^2 e^x t, \\ u_2(x, t) &= L_t^{-1}(u_1(x, t))_{xxt} + L_t^{-1}(-3A_1 + 2B_1 + C_1) \\ &= \frac{1}{2!} c^3 e^x t^2, \\ u_3(x, t) &= L_t^{-1}(u_2(x, t))_{xxt} + L_t^{-1}(-3A_2 + 2B_2 + C_2) \\ &= -\frac{1}{3!} c^4 e^x t^3, \\ u_4(x, t) &= L_t^{-1}(u_3(x, t))_{xxt} + L_t^{-1}(-3A_3 + 2B_3 + C_3) \\ &= \frac{1}{4!} c^4 e^x t^4, \end{aligned} \quad (55)$$

and so on. The series solution for $x < 0$ is given by

$$\begin{aligned} u(x, t) &= ce^x \\ &\times \left(1 - ct + \frac{1}{2!}(ct)^2 - \frac{1}{3!}(ct)^3 + \frac{1}{4!}(ct)^4 + \dots \right), \end{aligned} \quad (56)$$

and in a closed form by

$$u(x, t) = ce^{(x-ct)}. \quad (57)$$

Combining the results for both cases gives the peakon solution

$$u(x, t) = ce^{-|x-ct|}. \quad (58)$$

Figure 3 shows a peakon graph.

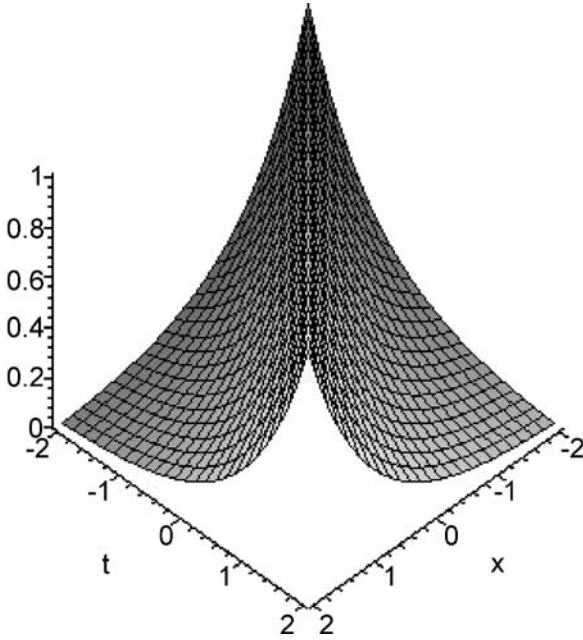
The graph shows that a peakon with a peak at a corner is generated with equal first derivatives at both sides of the peak, but with opposite signs.

Compactons of the K(n,n) Equation

The K(n,n) equation [16,17] was introduced by Rosenau and Hyman in 1993. This equation was investigated experimentally and analytically. The K(m,n) equation is a genuinely nonlinear dispersive equation, a special type of the KdV equation, named K(m,n), of the form

$$u_t + (u^m)_x + (u^n)_{xxx} = 0, \quad n > 1. \quad (59)$$

Compactons, which are solitons with compact support or strict localization of solitary waves, have been investigated thoroughly in the literature. The delicate interaction between the effect of the genuine nonlinear convection $(u^n)_x$ and the genuinely nonlinear dispersion of $(u^n)_{xxx}$ generates solitary waves with exact compact support that are called *compactons*. It was also discovered that solitary



Adomian Decomposition Method Applied to Non-linear Evolution Equations in Soliton Theory, Figure 3

Figure 3 shows the peakon graph $u(x, t) = e^{-|x-ct|}$, $c = 1$, $-2 \leq x$, $t \leq 2$

waves may compactify under the influence of nonlinear dispersion, which is capable of causing deep qualitative changes in the nature of genuinely nonlinear phenomena [16,17]. Unlike solitons that narrow as the amplitude increases, a compacton's width is independent of the amplitude. Compactons such as drops do not possess infinite wings; hence they interact among themselves only across short distances.

Compactons are nonanalytic solutions, whereas classical solitons are analytic solutions. The points of nonanalyticity at the edge of a compacton correspond to points of genuine nonlinearity for the differential equation and introduce singularities in the associated dynamical system for the traveling waves [16,17,27,28,29,30]. Compactons were proved to collide elastically and vanish outside a finite core region. This discovery was studied thoroughly by many researchers who were involved with identical nonlinear dispersive equations. It is to be noted that solutions were obtained only for cases where $m = n$. However, for $m \neq n$, solutions have not yet been determined.

Without loss of generality, we will examine the special K(3,3) initial value problem

$$u_t + (u^3)_x + (u^3)_{xxx} = 0, \quad u(x, 0) = \frac{\sqrt{6c}}{2} \cos\left(\frac{1}{3}x\right). \quad (60)$$

We first write the K(3,3) Eq. (60) in an operator form:

$$L_t u = -(u^3)_x - (u^3)_{xxx}, \quad u(x, 0) = \frac{\sqrt{6c}}{2} \cos\left(\frac{1}{3}x\right), \quad (61)$$

where the differential operator L_t is as defined above. The differential operator L_t is defined by

$$L_t = \frac{\partial}{\partial t}. \quad (62)$$

Applying L_t^{-1} to both sides of (61) and using the initial condition we find

$$u(x, t) = \frac{\sqrt{6c}}{2} \cos\left(\frac{1}{3}x\right) - L_t^{-1}((u^3)_x + (u^3)_{xxx}). \quad (63)$$

Notice that the right-hand side contains two nonlinear terms $(u^3)_x$ and $(u^3)_{xxx}$. Accordingly, the Adomian polynomials for these terms are given by

$$\begin{aligned} A_0 &= (u_0^3)_x, \\ A_1 &= (3u_0^2 u_1)_x, \\ A_2 &= (3u_0^2 u_2 + 3u_0 u_1^2)_x, \\ A_3 &= (3u_0^2 u_3 + 6u_0 u_1 u_2 + u_1^3)_x, \end{aligned} \quad (64)$$

and

$$\begin{aligned} B_0 &= (u_0^3)_{xxx}, \\ B_1 &= (3u_0^2 u_1)_{xxx}, \\ B_2 &= (3u_0^2 u_2 + 3u_0 u_1^2)_{xxx}, \\ B_3 &= (3u_0^2 u_3 + 6u_0 u_1 u_2 + u_1^3)_{xxx}, \end{aligned} \quad (65)$$

respectively.

Proceeding as before, Eq. (63) becomes

$$\sum_{n=0}^{\infty} u_n(x, t) = \frac{\sqrt{6c}}{2} \cos\left(\frac{1}{3}x\right) - L_t^{-1} \left(\sum_{n=0}^{\infty} A_n + B_n \right). \quad (66)$$

This gives the recursive relation

$$\begin{aligned} u_0(x, t) &= \frac{\sqrt{6c}}{2} \cos\left(\frac{1}{3}x\right), \\ u_{k+1}(x, t) &= -L_t^{-1}(L_t^{-1}(A_k + B_k)), \quad k \geq 0. \end{aligned} \quad (67)$$

The components u_n , $n \geq 0$ can be recursively determined as

$$\begin{aligned}
 u_0(x, t) &= \frac{\sqrt{6c}}{2} \cos\left(\frac{1}{3}x\right), \\
 u_1(x, t) &= \frac{c\sqrt{6c}}{6} \sin\left(\frac{1}{3}x\right) t, \\
 u_2(x, t) &= -\frac{c^2\sqrt{6c}}{36} \cos\left(\frac{1}{3}x\right) t^2, \\
 u_3(x, t) &= -\frac{c^3\sqrt{6c}}{324} \sin\left(\frac{1}{3}x\right) t^3, \\
 u_4(x, t) &= \frac{c^4\sqrt{6c}}{3888} \cos\left(\frac{1}{3}x\right) t^4, \\
 u_5(x, t) &= \frac{c^5\sqrt{6c}}{58320} \sin\left(\frac{1}{3}x\right) t^5,
 \end{aligned}
 \tag{68}$$

and so on. The series solution is thus given by

$$\begin{aligned}
 u(x, t) &= \frac{\sqrt{6c}}{2} \cos\left(\frac{1}{3}x\right) \left(1 - \frac{1}{2!} \left(\frac{ct}{3}\right)^2 + \frac{1}{4!} \left(\frac{ct}{3}\right)^4 + \dots\right) \\
 &\quad + \frac{\sqrt{6c}}{2} \sin\left(\frac{1}{3}x\right) \left(\frac{ct}{3} - \frac{1}{3!} \left(\frac{ct}{3}\right)^3 + \frac{1}{5!} \left(\frac{ct}{5}\right)^5 + \dots\right).
 \end{aligned}
 \tag{69}$$

This is equivalent to

$$u(x, t) = \frac{\sqrt{6c}}{2} \left(\cos\left(\frac{1}{3}x\right) \cos\left(\frac{1}{3}ct\right) + \sin\left(\frac{1}{3}x\right) \sin\left(\frac{1}{3}ct\right) \right),
 \tag{70}$$

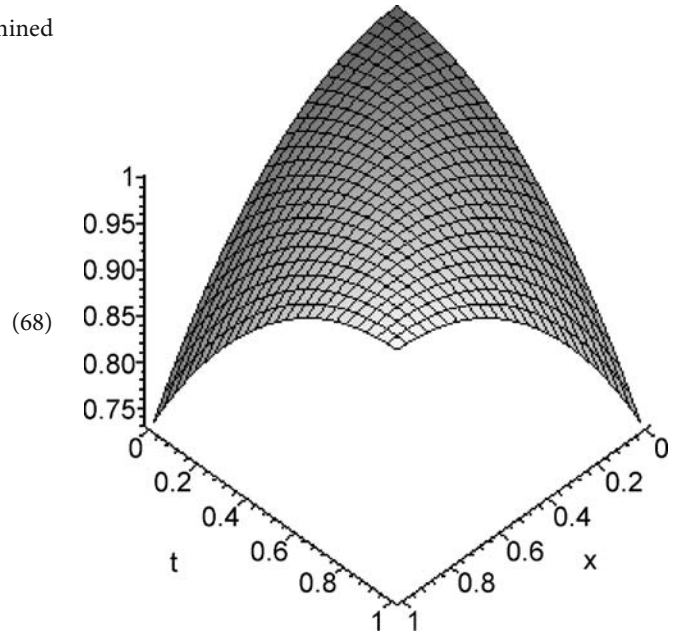
or equivalently

$$u(x, t) = \begin{cases} \left\{ \frac{\sqrt{6c}}{2} \cos\left[\frac{1}{3}(x - ct)\right] \right\}, & |x - ct| \leq \frac{3\pi}{2}, \\ 0 & \text{otherwise.} \end{cases}
 \tag{71}$$

This in turn gives the compacton solution of the K(3,3) equation. Figure 4 shows a compacton confined to a finite core without exponential wings.

The graph shows a compacton: a soliton free of exponential wings.

It is interesting to point out that the generalized solution of the K(n,n) equation, $n > 1$, is given by the



Adomian Decomposition Method Applied to Non-linear Evolution Equations in Soliton Theory, Figure 4

Figure 4 shows the compacton graph $u(x, t) = \cos^{\frac{1}{2}}(x - ct)$, $c = 1$, $0 \leq x, t \leq 1$

compactons

$$u(x, t) = \begin{cases} \left\{ \frac{2cn}{(n+1)} \cos^2\left[\frac{n-1}{2n}(x - ct)\right] \right\}^{\frac{1}{n-1}}, & |x - ct| \leq \frac{\pi}{2\mu}, \\ 0 & \text{otherwise,} \end{cases}
 \tag{72}$$

and

$$u_3(x, t) = \begin{cases} \left\{ \frac{2cn}{(n+1)} \sin^2\left[\frac{n-1}{2n}(x - ct)\right] \right\}^{\frac{1}{n-1}}, & |x - ct| \leq \frac{\pi}{\mu}, \\ 0 & \text{otherwise.} \end{cases}
 \tag{73}$$

Future Directions

The most significant advantage of the Adomian method is that it attacks any problem without any need for a transformation formula or any restrictive assumption that may change the physical behavior of the solution. For singular problems, it was possible to overcome the singularity phenomenon and to attain practically a series solution in a standard way. Moreover, for problems on unbounded domains, combining the obtained series solution

by the Adomian method with Padé approximants provides a promising tool to handle boundary value problems. The Padé approximants, which often show superior performance over series approximations, provide a promising tool for use in applied fields.

As stated above, unlike other methods such as Hirota and the inverse scattering method, where solutions can be obtained without using prescribed conditions, the Adomian method requires such conditions. Moreover, these conditions must be of a form that does not provide zero for the integrand of the first component $u_1(x, t)$. Such a case should be addressed to enhance the performance of the method. Another aspect that should be addressed is Adomian polynomials. The existing techniques require tedious work to evaluate it. Most importantly, the Adomian method guarantees only one solution for nonlinear problems. This is an issue that should be investigated to improve the performance of the Adomian method compared to other methods.

On the other hand, it is important to further examine the N-soliton solutions by simplified forms, such as the form given by Hereman and Nuseir in [7]. The bilinear form of Hirota is not that easy to use and is not always attainable for nonlinear models.

Bibliography

Primary Literature

1. Ablowitz MJ, Clarkson PA (1991) Solitons, nonlinear evolution equations and inverse scattering. Cambridge University Press, Cambridge
2. Adomian G (1984) A new approach to nonlinear partial differential equations. *J Math Anal Appl* 102:420–434
3. Boyd PJ (1997) Peakons and cashoidal waves: travelling wave solutions of the Camassa–Holm equation. *Appl Math Comput* 81(2–3):173–187
4. Camassa R, Holm D (1993) An integrable shallow water equation with peaked solitons. *Phys Rev Lett* 71(11):1661–1664
5. Drazin PG, Johnson RS (1996) Solitons: an introduction. Cambridge University Press, Cambridge
6. Helal MA, Mehanna MS (2006) A comparison between two different methods for solving KdV-Burgers equation. *Chaos Solitons Fractals* 28:320–326
7. Hereman W, Nuseir A (1997) Symbolic methods to construct exact solutions of nonlinear partial differential equations. *Math Comp Simul* 43:13–27
8. Hirota R (1971) Exact solutions of the Korteweg–de Vries equation for multiple collisions of solitons. *Phys Rev Lett* 27(18):1192–1194
9. Hirota R (1972) Exact solutions of the modified Korteweg–de Vries equation for multiple collisions of solitons. *J Phys Soc Jpn* 33(5):1456–1458
10. Korteweg DJ, de Vries G (1895) On the change of form of long waves advancing in a rectangular canal and on a new type of long stationary waves. *Philos Mag* 5th Ser 36:422–443
11. Lenells J (2005) Travelling wave solutions of the Camassa–Holm equation. *J Differ Equ* 217:393–430
12. Liu Z, Wang R, Jing Z (2004) Peaked wave solutions of Camassa–Holm equation. *Chaos Solitons Fractals* 19:77–92
13. Malfliet W (1992) Solitary wave solutions of nonlinear wave equations. *Am J Phys* 60(7):650–654
14. Malfliet W, Hereman W (1996) The tanh method: I. Exact solutions of nonlinear evolution and wave equations. *Phys Scr* 54:563–568
15. Malfliet W, Hereman W (1996) The tanh method: II. Perturbation technique for conservative systems. *Phys Scr* 54:569–575
16. Rosenau P (1998) On a class of nonlinear dispersive-dissipative interactions. *Phys D* 230(5/6):535–546
17. Rosenau P, Hyman J (1993) Compactons: solitons with finite wavelengths. *Phys Rev Lett* 70(5):564–567
18. Veksler A, Zarmi Y (2005) Wave interactions and the analysis of the perturbed Burgers equation. *Phys D* 211:57–73
19. Wadati M (1972) The exact solution of the modified Korteweg–de Vries equation. *J Phys Soc Jpn* 32:1681–1687
20. Wadati M (2001) Introduction to solitons. *Pramana J Phys* 57(5/6):841–847
21. Wazwaz AM (1997) Necessary conditions for the appearance of noise terms in decomposition solution series. *Appl Math Comput* 81:265–274
22. Wazwaz AM (1998) A reliable modification of Adomian's decomposition method. *Appl Math Comput* 92:1–7
23. Wazwaz AM (1999) A comparison between the Adomian decomposition method and the Taylor series method in the series solutions. *Appl Math Comput* 102:77–86
24. Wazwaz AM (2000) A new algorithm for calculating Adomian polynomials for nonlinear operators. *Appl Math Comput* 111(1):33–51
25. Wazwaz AM (2001) Exact specific solutions with solitary patterns for the nonlinear dispersive K(m,n) equations. *Chaos, Solitons and Fractals* 13(1):161–170
26. Wazwaz AM (2002) General solutions with solitary patterns for the defocusing branch of the nonlinear dispersive K(n,n) equations in higher dimensional spaces. *Appl Math Comput* 133(2/3):229–244
27. Wazwaz AM (2003) An analytic study of compactons structures in a class of nonlinear dispersive equations. *Math Comput Simul* 63(1):35–44
28. Wazwaz AM (2003) Compactons in a class of nonlinear dispersive equations. *Math Comput Model* 37(3/4):333–341
29. Wazwaz AM (2004) The tanh method for travelling wave solutions of nonlinear equations. *Appl Math Comput* 154(3):713–723
30. Wazwaz AM (2005) Compact and noncompact structures for variants of the KdV equation. *Int J Appl Math* 18(2):213–221
31. Wazwaz AM (2006) New kinds of solitons and periodic solutions to the generalized KdV equation. *Numer Methods Partial Differ Equ* 23(2):247–255
32. Wazwaz AM (2006) Peakons, kinks, compactons and solitary patterns solutions for a family of Camassa–Holm equations by using new hyperbolic schemes. *Appl Math Comput* 182(1):412–424
33. Wazwaz AM (2007) The extended tanh method for new solitons solutions for many forms of the fifth-order KdV equations. *Appl Math Comput* 184(2):1002–1014
34. Whitham GB (1999) Linear and nonlinear waves. Wiley, New York

35. Zabusky NJ, Kruskal MD (1965) Interaction of solitons in a collisionless plasma and the recurrence of initial states. *Phys Rev Lett* 15:240–243

Books and Reviews

- Adomian G (1994) *Solving Frontier Problems of Physics: The Decomposition Method*. Kluwer, Boston
- Burgers JM (1974) *The nonlinear diffusion equation*. Reidel, Dordrecht
- Conte R, Magri F, Musette M, Satsuma J, Winternitz P (2003) *Lecture Notes in Physics: Direct and Inverse methods in Nonlinear Evolution Equations*. Springer, Berlin
- Hirota R (2004) *The Direct Method in Soliton Theory*. Cambridge University Press, Cambridge
- Johnson RS (1997) *A Modern Introduction to the Mathematical Theory of Water Waves*. Cambridge University Press, Cambridge
- Kosmann-Schwarzbach Y, Grammaticos B, Tamizhmani KM (2004) *Lecture Notes in Physics: Integrability of Nonlinear Systems*. Springer, Berlin
- Wazwaz AM (1997) *A First Course in Integral Equations*. World Scientific, Singapore
- Wazwaz AM (2002) *Partial Differential Equations: Methods and Applications*. Balkema, The Netherlands

Agent Based Computational Economics

MOSHE LEVY

The Hebrew University, Jerusalem, Israel

Article Outline

[Glossary](#)

[Definition of the Subject](#)

[Introduction](#)

[Some of the Pioneering Studies](#)

[Illustration with the LLS Model](#)

[Summary and Future Directions](#)

[Bibliography](#)

Glossary

Agent-based simulation A simulation of a system of multiple interacting agents (sometimes also known as “microscopic simulation”). The “micro” rules governing the actions of the agents are known, and so are their rules of interaction. Starting with some initial conditions, the dynamics of the system are investigated by simulating the state of the system through discrete time steps. This approach can be employed to study general properties of the system, which are not sensitive to the initial conditions, or the dynamics of a specific system with fairly well-known initial conditions,

e.g. the impact of the baby boomers’ retirement on the US stock market.

Bounded-rationality Most economic models describe agents as being *fully* rational – given the information at their disposal they act in the optimal way which maximizes their objective (or utility) function. This optimization may be technically very complicated, requiring economic, mathematical and statistical sophistication. In contrast, *bounded* rational agents are limited in their ability to optimize. This limitation may be due to limited computational power, errors, or various psychological biases which have been experimentally documented.

Market anomalies Empirically documented phenomena that are difficult to explain within the standard rational representative agent economic framework. Some of these phenomena are the over-reaction and under-reaction of prices to news, the auto-correlation of stock returns, various calendar and day-of-the-week effects, and the excess volatility of stock returns.

Representative agent A standard modeling technique in economics, by which an entire class of agents (e.g. investors) are modeled by a single “representative” agent. If agents are completely homogeneous, it is obvious that the representative agent method is perfectly legitimate. However, when agents are heterogeneous, the representative agent approach can lead to a multitude of problems (see [16]).

Definition of the Subject

Mainstream economic models typically make the assumption that an entire group of agents, e.g. “investors”, can be modeled with a single “rational representative agent”. While this assumption has proven extremely useful in advancing the science of economics by yielding analytically tractable models, it is clear that the assumption is not realistic: people are different one from the other in their tastes, beliefs, and sophistication, and as many psychological studies have shown, they often deviate from rationality in systematic ways.

Agent Based Computational Economics is a framework allowing economics to expand beyond the realm of the “rational representative agent”. By modeling and simulating the behavior of each agent and the interaction among agents, agent based simulation allows us to investigate the dynamics of complex economic systems with many heterogeneous and not necessarily fully rational agents.

The agent based simulation approach allows economists to investigate systems that can not be studied with

the conventional methods. Thus, the following key questions can be addressed: How do heterogeneity and systematic deviations from rationality affect markets? Can these elements explain empirically observed phenomena which are considered “anomalies” in the standard economics literature? How robust are the results obtained with the analytical models? By addressing these questions the agent based simulation approach complements the traditional analytical analysis, and is gradually becoming a standard tool in economic analysis.

Introduction

For solving the dynamics of two bodies (e. g. stars) with some initial locations and velocities and some law of attraction (e. g. gravitation) there is a well-known analytical solution. However, for a similar system with three bodies there is no known analytical solution. Of course, this does not mean that physicists can’t investigate and predict the behavior of such systems. Knowing the state of the system (i. e. the location, velocity, and acceleration of each body) at time t , allows us to calculate the state of the system an instant later, at time $t + \Delta t$. Thus, starting with the initial conditions we can predict the dynamics of the system by simply simulating the “behavior” of each element in the system over time.

This powerful and fruitful approach, sometimes called “Microscopic Simulation”, has been adopted by many other branches of science. Its application in economics is best known as “Agent Based Simulation” or “Agent Based Computation”. The advantages of this approach are clear – they allow the researcher to go where no analytical models can go. Yet, despite of the advantages, perhaps surprisingly, the agent based approach was not adopted very quickly by economists. Perhaps the main reason for this is that a particular simulation only describes the dynamics of a system with a particular set of parameters and initial conditions. With other parameters and initial conditions the dynamics may be different. So economists may ask: what is the value of conducting simulations if we get very different results with different parameter values? While in physics the parameters (like the gravitational constant) may be known with great accuracy, in economics the parameters (like the risk-aversion coefficient, or for that matter the entire decision-making rule) are typically estimated with substantial error. This is a strong point. Indeed, we would argue that the “art” of agent based simulations is the ability to understand the general dynamics of the system and to draw general conclusions from a finite number of simulations. Of course, one simulation is sufficient as a counterexample to show that a certain result does not hold, but

many more simulations are required in order to convince of an alternative general regularity.

This manuscript is intended as an introduction to agent-based computational economics. An introduction to this field has two goals: (i) to explain and to demonstrate the agent-based methodology in economics, stressing the advantages and disadvantages of this approach relative to the alternative purely analytical methodology, and (ii) to review studies published in this area. The emphasis in this paper will be on the first goal. While Sect. “[Some of the Pioneering Studies](#)” does provide a brief review of some of the cornerstone studies in this area, more comprehensive reviews can be found in [19,24,32,39,40], on which part of Sect. “[Some of the Pioneering Studies](#)” is based. A comprehensive review of the many papers employing agent based computational models in economics will go far beyond the scope of this article. To achieve goal (i) above, in Sect. “[Illustration with the LLS Model](#)” we will focus on one particular model of the stock market in some detail. Section “[Summary and Future Directions](#)” concludes with some thoughts about the future of the field.

Some of the Pioneering Studies

Schelling’s Segregation Model

Schelling’s [36] classical segregation model is one of the earliest models of population dynamics. Schelling’s model is not intended as a realistic tool for studying the actual dynamics of specific communities as it ignores economic, real-estate and cultural factors. Rather, the aim of this very simplified model is to explain the emergence of macroscopic single-race neighborhoods even when individuals are not racists. More precisely, Schelling found that the collective effect of neighborhood racial segregation results even from individual behavior that presents only a very mild preference for same-color neighbors. For instance, even the minimal requirement by each individual of having (at least) one neighbor belonging to ones’ own race leads to the segregation effect.

The agent based simulation starts with a square mesh, or lattice, (representing a town) which is composed of cells (representing houses). On these cells reside agents which are either “blue” or “green” (the different races). The crucial parameter is the minimal percentage of same-color neighbors that each agent requires. Each agent, in his turn, examines the color of all his neighbors. If the percentage of neighbors belonging to his own group is above the “minimal percentage”, the agent does nothing. If the percentage of neighbors of his own color is less then the minimal percentage, the agent moves to the closest unoccupied cell. The agent then examines the color of the neighbors of the

new location and acts accordingly (moves if the number of neighbors of his own color is below the minimal percentage and stays there otherwise). This goes on until the agent is finally located at a cite in which the minimal percentage condition holds. After a while, however, it might happen that following the moves of the other agents, the minimal percentage condition ceases to be fulfilled and then the agent starts moving again until he finds an appropriate cell. As mentioned above, the main result is that even for very mild individual preferences for same-color neighbors, after some time the entire system displays a very high level of segregation.

A more modern, developed and sophisticated reincarnation of these ideas is the Sugarscape environment described by Epstein and Axtell [6]. The model considers a population of moving, feeding, pairing, procreating, trading, warring agents and displays various qualitative collective events which their populations incur. By employing agent based simulation one can study the macroscopic results induced by the agents' individual behavior.

The Kim and Markowitz Portfolio Insurers Model

Harry Markowitz is very well known for being one of the founders of modern portfolio theory, a contribution for which he has received the Nobel Prize in economics. It is less well known, however, that Markowitz is also one of the pioneers in employing agent based simulations in economics.

During the October 1987 crash markets all over the globe plummeted by more than 20% within a few days. The surprising fact about this crash is that it appeared to be spontaneous – it was not triggered by any obvious event. Following the 1987 crash researchers started to look for endogenous market features, rather than external forces, as sources of price variation. The Kim-Markowitz [15] model explains the 1987 crash as resulting from investors' "Constant Proportion Portfolio Insurance" (CPPI) policy. Kim and Markowitz proposed that market instabilities arise as a consequence of the individual insurers' efforts to cut their losses by selling once the stock prices are going down.

The Kim Markowitz agent based model involves two groups of individual investors: rebalancers and insurers (CPPI investors). The rebalancers are aiming to keep a constant composition of their portfolio, while the insurers make the appropriate operations to insure that their eventual losses will not exceed a certain fraction of the investment per time period.

The rebalancers act to keep a portfolio structure with (for instance) half of their wealth in cash and half in stocks.

If the stock price rises, then the stocks weight in the portfolio will increase and the rebalancers will sell shares until the shares again constitute 50% of the portfolio. If the stock price decreases, then the value of the shares in the portfolio decreases, and the rebalancers will buy shares until the stock again constitutes 50% of the portfolio. Thus, the rebalancers have a stabilizing influence on the market by selling when the market rises and buying when the market falls.

A typical CPPI investor has as his/her main objective not to lose more than (for instance) 25% of his initial wealth during a quarter, which consists of 65 trading days. Thus, he aims to insure that at each cycle 75% of the initial wealth is out of reasonable risk. To this effect, he assumes that the current value of the stock will not fall in one day by more than a factor of 2. The result is that he always keeps in stock twice the difference between the present wealth and 75% of the initial wealth (which he had at the beginning of the 65 days investing period). This determines the amount the CPPI agent is bidding or offering at each stage. Obviously, after a price fall, the amount he wants to keep in stocks will fall and the CPPI investor will sell and further destabilize the market. After an increase in the prices (and personal wealth) the amount the CPPI agent wants to keep in shares will increase: he will buy, and may support a price bubble.

The simulations reveal that even a relatively small fraction of CPPI investors (i. e. less than 50%) is enough to destabilize the market, and crashes and booms are observed. Hence, the claim of Kim and Markowitz that the CPPI policy may be responsible for the 1987 crash is supported by the agent based simulations. Various variants of this model were studied intensively by Egenter, Lux and Stauffer [5] who find that the price time evolution becomes unrealistically periodic for a large number of investors (the periodicity seems related with the fixed 65 days quarter and is significantly diminished if the 65 day period begins on a different date for each investor).

The Arthur, Holland, Lebaron, Palmer and Tayler Stock Market Model

Palmer, Arthur, Holland, Lebaron and Tayler [30] and Arthur, Holland, Lebaron, Palmer and Tayler [3] (AHLPT) construct an agent based simulation model that is focused on the concept of co-evolution. Each investor adapts his/her investment strategy such as to maximally exploit the market dynamics generated by the investment strategies of all others investors. This leads to an ever-evolving market, driven endogenously by the ever-changing strategies of the investors.

The main objective of AHLPT is to prove that market fluctuations may be induced by this endogenous co-evolution, rather than by exogenous events. Moreover, AHLPT study the various regimes of the system: the regime in which rational fundamentalist strategies are dominating vs. the regime in which investors start developing strategies based on technical trading. In the technical trading regime, if some of the investors follow fundamentalist strategies, they may be punished rather than rewarded by the market. AHLPT also study the relation between the various strategies (fundamentals vs. technical) and the volatility properties of the market (clustering, excess volatility, volume-volatility correlations, etc.).

In the first paper quoted above, the authors simulated a single stock and further limited the bid/offer decision to a ternary choice of: i) bid to buy one share, ii) offer to sell one share, or: iii) do nothing. Each agent had a collection of rules which described how he should behave (i, ii or iii) in various market conditions. If the current market conditions were not covered by any of the rules, the default was to do nothing. If more than one rule applied in a certain market condition, the rule to act upon was chosen probabilistically according to the “strengths” of the applicable rules. The “strength” of each rule was determined according to the rule’s past performance: rules that “worked” became “stronger”. Thus, if a certain rule performed well, it became more likely to be used again.

The price is updated proportionally to the relative excess of offers over demands. In [3], the rules were used to predict future prices. The price prediction was then transformed into a buy/sell order through the use of a Constant Absolute Risk Aversion (CARA) utility function. The use of CARA utility leads to demands which do not depend on the investor’s wealth.

The heart of the AHLPT dynamics are the trading rules. In particular, the authors differentiate between “fundamental” rules and “technical” rules, and study their relative strength in various market regimes. For instance, a “fundamental” rule may require a market conditions of the type:

$$\text{dividend/current price} > 0.04$$

in order to be applied. A “technical” rule may be triggered if the market fulfills a condition of the type:

$$\text{current price} > 10\text{-period moving-average of past prices.}$$

The rules undergo genetic dynamics: the weakest rules are substituted periodically by copies of the strongest rules and all the rules undergo random mutations (or even versions of “sexual” crossovers: new rules are formed by combining parts from 2 different rules). The genetic dynamics

of the trading rules represent investors’ learning: new rules represent new trading strategies. Investors examine new strategies, and adopt those which tend to work best. The main results of this model are:

For a Few Agents, a Small Number of Rules, and Small Dividend Changes

- The price converges towards an equilibrium price which is close to the fundamental value.
- Trading volume is low.
- There are no bubbles, crashes or anomalies.
- Agents follow homogeneous simple fundamentalist rules.

For a Large Number of Agents and a Large Number of Rules

- There is no convergence to an equilibrium price, and the dynamics are complex.
- The price displays occasional large deviations from the fundamental value (bubbles and crashes).
- Some of these deviations are triggered by the emergence of collectively self-fulfilling agent price-prediction rules.
- The agents become heterogeneous (adopt very different rules).
- Trading volumes fluctuate (large volumes correspond to bubbles and crashes).
- The rules evolve over time to more and more complex patterns, organized in hierarchies (rules, exceptions to rules, exceptions to exceptions, and so on ...).
- The successful rules are time dependent: a rule which is successful at a given time may perform poorly if reintroduced after many cycles of market co-evolution.

The Lux and Lux and Marchesi Model

Lux [27] and Lux and Marchesi [28] propose a model to endogenously explain the heavy tail distribution of returns and the clustering of volatility. Both of these phenomena emerge in the Lux model as soon as one assumes that in addition to the fundamentalists there are also chartists in the model. Lux and Marchesi [28] further divide the chartists into optimists (buyers) and pessimists (sellers). The market fluctuations are driven and amplified by the fluctuations in the various populations: chartists converting into fundamentalists, pessimists into optimists, etc.

In the Lux and Marchesi model the stock’s fundamental value is exogenously determined. The fluctuations of the fundamental value are inputted exogenously as a white

noise process in the logarithm of the value. The market price is determined by investors' demands and by the market clearance condition.

Lux and Marchesi consider three types of traders:

- **Fundamentalists** observe the fundamental value of the stock. They anticipate that the price will eventually converge to the fundamental value, and their demand for shares is proportional to the difference between the market price and the fundamental value.
- **Chartists** look more at the present trends in the market price rather than at fundamental economic values; the chartists are divided into
- **Optimists** (they buy a fixed amount of shares per unit time)
- **Pessimists** (they sell shares).

Transitions between these three groups (optimists, pessimists, fundamentalists) happen with probabilities depending on the market dynamics and on the present numbers of traders in each of the three classes:

- **The transition probabilities of chartists** depend on the majority opinion (through an "opinion index" measuring the relative number of optimists minus the relative number of pessimists) and on the actual price trend (the current time derivative of the current market price), which determines the relative profit of the various strategies.
- **The fundamentalists decide to turn into chartists** if the profits of the later become significantly larger than their own, and vice versa (the detailed formulae used by Lux and Marchesi are inspired from the exponential transition probabilities governing statistical mechanics physical systems).

The main results of the model are:

- No long-term deviations between the current market price and the fundamental price are observed.
- The deviations from the fundamental price, which do occur, are unsystematic.
- In spite of the fact that the variations of the fundamental price are normally distributed, the variations of the market price (the market returns) are not. In particular the returns exhibit a frequency of extreme events which is higher than expected for a normal distribution. The authors emphasize the amplification role of the market that transforms the input normal distribution of the fundamental value variations into a leptokurtotic (heavy tailed) distribution of price variation, which is encountered in the actual financial data.
- clustering of volatility.

The authors explain the volatility clustering (and as a consequence, the leptokurticity) by the following mechanism. In periods of high volatility, the fundamental information is not very useful to insure profits, and a large fraction of the agents become chartists. The opposite is true in quiet periods when the actual price is very close to the fundamental value. The two regimes are separated by a threshold in the number of chartist agents. Once this threshold is approached (from below) large fluctuations take place which further increase the number of chartists. This destabilization is eventually dampened by the energetic intervention of the fundamentalists when the price deviates too much from the fundamental value. The authors compare this temporal instability with the on-off intermittence encountered in certain physical systems. According to Egenter et al. [5], the fraction of chartists in the Lux Marchesi model goes to zero as the total number of traders goes to infinity, when the rest of the parameters are kept constant.

Illustration with the LLS Model

The purpose of this section is to give a more detailed "hands on" example of the agent based approach, and to discuss some of the practical dilemmas arising when implementing this approach, by focusing on one specific model. We will focus on the so called LLS Model of the stock market (for more detail, and various versions of the model, see [11,17,22,23,24,25]). This section is based on the presentation of the LLS Model in Chap. 7 of [24]).

Background

Real life investors differ in their investment behavior from the investment behavior of the idealized representative rational investor assumed in most economic and financial models. Investors differ one from the other in their preferences, their investment horizon, the information at their disposal, and their interpretation of this information. No financial economist seriously doubts these observations. However, modeling the empirically and experimentally documented investor behavior and the heterogeneity of investors is very difficult and in most cases practically impossible to do within an analytic framework. For instance, the empirical and experimental evidence suggests that most investors are characterized by Constant Relative Risk Aversion (CRRA), which implies a power (myopic) utility function (see Eq. (2) below). However, for a general distribution of returns it is impossible to obtain an analytic solution for the portfolio optimization problem of investors with these preferences. Extrapolation of future returns from past returns, biased probability weighting, and partial deviations from rationality are also all ex-

perimentally documented but difficult to incorporate in an analytical setting. One is then usually forced to make the assumptions of rationality and homogeneity (at least in some dimension) and to make unrealistic assumptions regarding investors' preferences, in order to obtain a model with a tractable solution. The hope in these circumstances is that the model will capture the essence of the system under investigation, and will serve as a useful benchmark, even though some of the underlying assumptions are admittedly false.

Most homogeneous rational agent models lead to the following predictions: no trading volume, zero autocorrelation of returns, and price volatility which is equal to or lower than the volatility of the "fundamental value" of the stock (defined as the present value of all future dividends, see [37]). However, the empirical evidence is very different:

- Trading volume can be extremely heavy [1,14].
- Stock returns exhibit short-run momentum (positive autocorrelation) and long-run mean reversion (negative autocorrelation) [7,13,21,31].
- Stock returns are excessively volatile relative to the dividends [37].

As most standard rational-representative-agent models cannot explain these empirical findings, these phenomena are known as "anomalies" or "puzzles". Can these "anomalies" be due to elements of investors' behavior which are unmodeled in the standard rational-representative-agent models, such as the experimentally documented deviations of investors' behavior from rationality and/or the heterogeneity of investors? The agent based simulation approach offers us a tool to investigate this question. The strength of the agent based simulation approach is that since it is not restricted to the scope of analytical methods, one is able to investigate virtually any imaginable investor behavior and market structure. Thus, one can study models which incorporate the experimental findings regarding the behavior of investors, and evaluate the effects of various behavioral elements on market dynamics and asset pricing.

The LLS model incorporates some of the main empirical findings regarding investor behavior, and we employ this model in order to study the effect of each element of investor behavior on asset pricing and market dynamics. We start out with a benchmark model in which all of the investors are rational, informed and identical, and then, one by one, we add elements of heterogeneity and deviations from rationality to the model in order to study their effects on the market dynamics.

In the benchmark model all investors are Rational, Informed and Identical (RII investors). This is, in effect, a "representative agent" model. The RII investors are informed about the dividend process, and they rationally act to maximize their expected utility. The RII investors make investment decisions based on the present value of future cash flows. They are essentially fundamentalists who evaluate the stock's fundamental value and try to find bargains in the market. The benchmark model in which all investors are RII yields results which are typical of most rational-representative-agent models: in this model prices follow a random walk, there is no excess volatility of the prices relative to the volatility of the dividend process, and since all agents are identical, there is no trading volume.

After describing the properties of the benchmark model, we investigate the effects of introducing various elements of investor behavior which are found in laboratory experiments but are absent in most standard models. We do so by adding to the model a minority of investors who do not operate like the RII investors. These investors are Efficient Market Believers (EMB from now on). The EMBs are investors who believe that the price of the stock reflects all of the currently available information about the stock. As a consequence, they do not try to time the market or to buy bargain stocks. Rather, their investment decision is reduced to the optimal diversification problem. For this portfolio optimization, the *ex-ante* return distribution is required. However, since the *ex-ante* distribution is unknown, the EMB investors use the *ex-post* distribution in order to estimate the *ex-ante* distribution. It has been documented that in fact, many investors form their expectations regarding the future return distribution based on the distribution of past returns.

There are various ways to incorporate the investment decisions of the EMBs. This stems from the fact that there are different ways to estimate the *ex-ante* distribution from the *ex-post* distribution. How far back should one look at the historical returns? Should more emphasis be given to more recent returns? Should some "outlier" observations be filtered out? etc. Of course, there are no clear answers to these questions, and different investors may have different ways of forming their estimation of the *ex-ante* return distribution (even though they are looking at the same series of historical returns). Moreover, some investors may use the objective *ex-post* probabilities when constructing their estimation of the *ex-ante* distribution, whereas others may use biased subjective probability weights. In order to build the analysis step-by-step we start by analyzing the case in which the EMB population is homogeneous, and then introduce various forms of heterogeneity into this population.

An important issue in market modeling is that of the degree of investors' rationality. Most models in economics and finance assume that people are fully rational. This assumption usually manifests itself as the maximization of an expected utility function by the individual. However, numerous experimental studies have shown that people deviate from rational decision-making [41,42,43,44,45]. Some studies model deviations from the behavior of the rational agent by introducing a sub-group of liquidity, or "noise", traders. These are traders that buy and sell stocks for reasons that are not directly related to the future payoffs of the financial asset - their motivation to trade arises from outside of the market (for example, a "noise trader's" daughter unexpectedly announces her plans to marry, and the trader sells stocks because of this unexpected need for cash). The exogenous reasons for trading are assumed random, and thus lead to random or "noise" trading (see [10]). The LLS model takes a different approach to the modeling of noise trading. Rather than dividing investors into the extreme categories of "fully rational" and "noise traders", the LLS model assumes that most investors try to act as rationally as they can, but are influenced by a multitude of factors causing them to deviate to some extent from the behavior that would have been optimal from their point of view. Namely, all investors are characterized by a utility function and act to maximize their expected utility; however, some investors may deviate to some extent from the optimal choice which maximizes their expected utility. These deviations from the optimal choice may be due to irrationality, inefficiency, liquidity constraints, or a combination of all of the above.

In the framework of the LLS model we examine the effects of the EMBs' deviations from rationality and their heterogeneity, relative to the benchmark model in which investors are informed, rational and homogeneous. We find that the behavioral elements which are empirically documented, namely, extrapolation from past returns, deviation from rationality, and heterogeneity among investors, lead to all of the following empirically documented "puzzles":

- Excess volatility
- Short-term momentum
- Longer-term return mean-reversion
- Heavy trading volume
- Positive correlation between volume and contemporaneous absolute returns
- Positive correlation between volume and lagged absolute returns

The fact that all these anomalies or "puzzles", which are hard to explain with standard rational-representative-

agent models, are generated naturally by a simple model which incorporates the experimental findings regarding investor behavior and the heterogeneity of investors, leads one to suspect that these behavioral elements and the diversity of investors are a crucial part of the workings of the market, and as such they cannot be "assumed away". As the experimentally documented bounded-rational behavior and heterogeneity are in many cases impossible to analyze analytically, agent based simulation presents a very promising tool for investigating market models incorporating these elements.

The LLS Model

The stock market consists of two investment alternatives: a stock (or index of stocks) and a bond. The bond is assumed to be a riskless asset, and the stock is a risky asset. The stock serves as a proxy for the market portfolio (e.g., the Standard & Poors 500 index). The extension from one risky asset to many risky assets is possible; however, one stock (the index) is sufficient for our present analysis because we restrict ourselves to global market phenomena and do not wish to deal with asset allocation across several risky assets. Investors are allowed to revise their portfolio at given time points, i.e. we discuss a discrete time model.

The bond is assumed to be a riskless investment yielding a constant return at the end of each time period. The bond is in infinite supply and investors can buy from it as much as they wish at a given rate of r_f . The stock is in finite supply. There are N outstanding shares of the stock. The return on the stock is composed of two elements:

- a) Capital Gain: If an investor holds a stock, any rise (fall) in the price of the stock contributes to an increase (decrease) in the investor's wealth.
- b) Dividends: The company earns income and distributes dividends at the end of each time period. We denote the dividend per share paid at time t by D_t . We assume that the dividend is a stochastic variable following a multiplicative random walk, i.e., $\tilde{D}_t = D_{t-1}(1 + \tilde{z})$, where \tilde{z} is a random variable with some probability density function $f(z)$ in the range $[z_1, z_2]$. (In order to allow for a dividend cut as well as a dividend increase we typically choose: $z_1 < 0, z_2 > 0$).

The total return on the stock in period t , which we denote by R_t is given by:

$$\tilde{R}_t = \frac{\tilde{P}_t + \tilde{D}_t}{P_{t-1}}, \quad (1)$$

where \tilde{P}_t is the stock price at time t .

All investors in the model are characterized by a von Neuman-Morgenstern utility function. We assume that all

investors have a power utility function of the form:

$$U(W) = \frac{W^{1-\alpha}}{1-\alpha}, \quad (2)$$

where α is the risk aversion parameter. This form of utility function implies Constant Relative Risk Aversion (CRRA). We employ the power utility function (Eq. (2)) because the empirical evidence suggests that relative risk aversion is approximately constant (see, for example [8,9,18,20]), and the power utility function is the unique utility function which satisfies the CRRA condition. Another implication of CRRA is that the optimal investment choice is independent of the investment horizon [33,34]. In other words, regardless of investors' actual investment horizon, they choose their optimal portfolio as though they are investing for a single period. The myopia property of the power utility function simplifies our analysis, as it allows us to assume that investors maximize their one-period-ahead expected utility.

We model two different types of investors: Rational, Informed, Identical (RII) investors, and Efficient Market Believers (EMB). These two investor types are described below.

Rational Informed Identical (RII) Investors RII investors evaluate the “fundamental value” of the stock as the discounted stream of all future dividends, and thus can also be thought of as “fundamentalists”. They believe that the stock price may deviate from the fundamental value in the short run, but if it does, it will eventually converge to the fundamental value. The RII investors act according to the assumption of asymptotic convergence: if the stock price is low relative to the fundamental value they buy in anticipation that the underpricing will be corrected, and vice versa. We make the simplifying assumption that the RII investors believe that the convergence of the price to the fundamental value will occur in the next period, however, our results hold for the more general case where the convergence is assumed to occur some T periods ahead, with $T > 1$.

In order to estimate next period's return distribution, the RII investors need to estimate the distribution of next period's price, \tilde{P}_{t+1} , and of next period's dividend, \tilde{D}_{t+1} . Since they know the dividend process, the RII investors know that $\tilde{D}_{t+1} = D_t(1 + \tilde{z})$ where \tilde{z} is distributed according to $f(z)$ in the range $[z_1, z_2]$. The RII investors employ Gordon's dividend stream model in order to calculate the fundamental value of the stock:

$$P_{t+1}^f = \frac{E_{t+1}[\tilde{D}_{t+2}]}{k - g}, \quad (3)$$

where the superscript f stands for the *fundamental* value, $E_{t+1}[\tilde{D}_{t+2}]$ is the dividend corresponding to time $t + 2$ as expected at time $t + 1$, k is the discount factor or the expected rate of return demanded by the market for the stock, and g is the *expected* growth rate of the dividend, i. e., $g = E(\tilde{z}) = \int_{z_1}^{z_2} f(z)zdz$.

The RII investors believe that the stock price may temporarily deviate from the fundamental value; however, they also believe that the price will eventually converge to the fundamental value. For simplification we assume that the RII investors believe that the convergence to the fundamental value will take place next period. Thus, the RII investors estimate P_{t+1} as:

$$P_{t+1} = P_{t+1}^f.$$

The expectation at time $t + 1$ of \tilde{D}_{t+2} depends on the realized dividend observed at $t + 1$:

$$E_{t+1}[\tilde{D}_{t+2}] = D_{t+1}(1 + g).$$

Thus, the RII investors believe that the price at $t + 1$ will be given by:

$$P_{t+1} = P_{t+1}^f = \frac{D_{t+1}(1 + g)}{k - g}.$$

At time t , D_t is known, but D_{t+1} is not; therefore P_{t+1}^f is also not known with certainty at time t . However, given D_t , the RII investors know the distribution of \tilde{D}_{t+1} :

$$\tilde{D}_{t+1} = D_t(1 + \tilde{z}),$$

where \tilde{z} is distributed according to the known $f(z)$. The realization of \tilde{D}_{t+1} determines P_{t+1}^f . Thus, at time t , RII investors believe that P_{t+1} is a random variable given by:

$$\tilde{P}_{t+1} = \tilde{P}_{t+1}^f = \frac{D_t(1 + \tilde{z})(1 + g)}{k - g}.$$

Notice that the RII investors face uncertainty regarding next period's price. In our model we assume that the RII investors are certain about the dividend growth rate g , the discount factor k , and the fact that the price will converge to the fundamental value next period. In this framework the only source of uncertainty regarding next period's price stems from the uncertainty regarding next period's dividend realization. More generally, the RII investors' uncertainty can result from uncertainty regarding any one of the above factors, or a combination of several of these factors. Any mix of these uncertainties is possible to investigate in the agent based simulation framework, but very hard, if not impossible, to incorporate in an analytic

framework. As a consequence of the uncertainty regarding next period's price and of their risk aversion, the RII investors do not buy an infinite number of shares even if they perceive the stock as underpriced. Rather, they estimate the stock's next period's return distribution, and find the optimal mix of the stock and the bond which maximizes their expected utility. The RII investors estimate next period's return on the stock as:

$$\tilde{R}_{t+1} = \frac{\tilde{P}_{t+1} + \tilde{D}_{t+1}}{P_t} = \frac{\frac{D_t(1+\tilde{z})(1+g)}{k-g} + D_t(1+\tilde{z})}{P_t}, \quad (4)$$

where \tilde{z} , the next year growth in the dividend, is the source of uncertainty. The demands of the RII investors for the stock depend on the price of the stock. For any *hypothetical* price P_h investors calculate the proportion of their wealth x they should invest in the stock in order to maximize their expected utility. The RII investor i believes that if she invests a proportion x of her wealth in the stock at time t , then at time $t+1$ her wealth will be:

$$\tilde{W}_{t+1}^i = W_h^i[(1-x)(1+r_f) + x\tilde{R}_{t+1}], \quad (5)$$

where \tilde{R}_{t+1} is the return on the stock, as given by Eq. (4), and W_h^i is the wealth of investor i at time t given that the stock price at time t is P_h .

If the price in period t is the hypothetical price P_h , the $t+1$ expected utility of investor i is the following function of her investment proportion in the stock, x :

$$EU(\tilde{W}_{t+1}^i) = EU\left(W_h^i[(1-x)(1+r_f) + x\tilde{R}_{t+1}]\right). \quad (6)$$

Substituting \tilde{R}_{t+1} from Eq. (4), using the power utility function (Eq. (2)), and substituting the hypothetical price P_h for P_t , the expected utility becomes the following function of x :

$$EU(\tilde{W}_{t+1}^i) = \frac{(W_h^i)^{1-\alpha}}{1-\alpha} \int_{z_1}^{z_2} \left[(1-x)(1+r_f) + x \left(\frac{\frac{D_t(1+z)(1+g)}{k-g} + D_t(1+z)}{P_h} \right) \right]^{1-\alpha} f(z) dz, \quad (7)$$

where the integration is over all possible values of z . In the agent based simulation framework, this expression for the expected utility, and the optimal investment proportion x , can be solved numerically for any general choice of distribution $f(z)$. For the sake of simplicity we restrict the present analysis to the case where \tilde{z} is distributed uniformly in the range $[z_1, z_2]$. This simplification leads to the

following expression for the expected utility:

$$\begin{aligned} EU(\tilde{W}_{t+1}^i) &= \frac{(W_h^i)^{1-\alpha}}{(1-\alpha)(2-\alpha)} \frac{1}{(z_2 - z_1)} \left(\frac{k-g}{k+1} \right) \frac{P_h}{x D_t} \\ &\left\{ \left[(1-x)(1+r_f) + \frac{x}{P_h} \left(\frac{k+1}{k-g} \right) D_t(1+z_2) \right]^{(2-\alpha)} \right. \\ &\quad \left. - \left[(1-x)(1+r_f) + \frac{x}{P_h} \left(\frac{k+1}{k-g} \right) D_t(1+z_1) \right]^{(2-\alpha)} \right\} \end{aligned} \quad (8)$$

For any hypothetical price P_h , each investor (numerically) finds the optimal proportion x_h which maximizes his/her expected utility given by Eq. (8). Notice that the optimal proportion, x_h , is independent of the wealth, W_h^i . Thus, if all RII investors have the same degree of risk aversion, α , they will have the same optimal investment proportion in the stock, regardless of their wealth. The number of shares demanded by investor i at the hypothetical price P_h is given by:

$$N_h^i(P_h) = \frac{x_h^i(P_h) W_h^i(P_h)}{P_h}. \quad (9)$$

Efficient Market Believers (EMB) The second type of investors in the LLS model are EMBs. The EMBs believe in market efficiency - they believe that the stock price accurately reflects the stock's fundamental value. Thus, they do not try to time the market or to look for "bargain" stocks. Rather, their investment decision is reduced to the optimal diversification between the stock and the bond. This diversification decision requires the *ex-ante* return distribution for the stock, but as the *ex-ante* distribution is not available, the EMBs assume that the process generating the returns is fairly stable, and they employ the *ex-post* distribution of stock returns in order to estimate the *ex-ante* return distribution.

Different EMB investors may disagree on the optimal number of *ex-post* return observations that should be employed in order to estimate the *ex-ante* return distribution. There is a trade-off between using more observations for better statistical inference, and using a smaller number of only more recent observations, which are probably more representative of the *ex-ante* distribution. As in reality, there is no "recipe" for the optimal number of observations to use. EMB investor i believes that the m^i most recent returns on the stock are the best estimate of the *ex-ante* distribution. Investors create an estimation of the *ex-ante* return distribution by assigning an equal probability

to each of the m^i most recent return observations:

$$\text{Prob}^i(\tilde{R}_{t+1} = R_{t-j}) = \frac{1}{m^i} \quad \text{for } j = 1, \dots, m^i \quad (10)$$

The expected utility of EMB investor i is given by:

$$\begin{aligned} EU(W_{t+1}^i) &= \frac{(W_h^i)^{1-\alpha}}{(1-\alpha)} \frac{1}{m^i} \sum_{j=1}^{m^i} [(1-x)(1+r_f) + xR_{t-j}]^{1-\alpha}, \end{aligned} \quad (11)$$

where the summation is over the set of m^i most recent *ex post* returns, x is the proportion of wealth invested in the stock, and as before W_h^i is the wealth of investor i at time t given that the stock price at time t is P_h . Notice that W_h^i does not change the optimal diversification policy, i. e., x . Given a set of m^i past returns, the optimal portfolio for the EMB investor i is an investment of a proportion x^{*i} in the stock and $(1-x^{*i})$ in the bond, where x^{*i} is the proportion which maximizes the above expected utility (Eq. (11)) for investor i . Notice that x^{*i} generally cannot be solved for analytically. However, in the agent based simulation framework this does not constitute a problem, as one can find x^{*i} numerically.

Deviations from Rationality Investors who are efficient market believers, and are rational, choose the investment proportion x^* which maximizes their expected utility. However, many empirical studies have shown that the behavior of investors is driven not only by rational expected utility maximization but by a multitude of other factors (see, for example, [34,41,42,43,44]). Deviations from the optimal rational investment proportion can be due to the cost of resources which are required for the portfolio optimization: time, access to information, computational power, etc., or due to exogenous events (for example, an investor plans to revise his portfolio, but gets distracted because his car breaks down). We assume that the different factors causing the investor to deviate from the optimal investment proportion x^* are random and uncorrelated with each other. By the central limit theorem, the aggregate effect of a large number of random uncorrelated influences is a normally distributed random influence, or “noise”. Hence, we model the effect of all the factors causing the investor to deviate from his optimal portfolio by adding a normally distributed random variable to the optimal investment proportion. To be more specific, we assume:

$$x^i = x^{*i} + \tilde{\varepsilon}^i, \quad (12)$$

where $\tilde{\varepsilon}^i$ is a random variable drawn from a truncated normal distribution with mean zero and standard deviation σ . Notice that noise is investor-specific, thus, $\tilde{\varepsilon}^i$ is drawn separately and independently for each investor.

The noise can be added to the decision-making of the RII investors, the EMB investors, or to both. The results are not much different with these various approaches. Since the RII investors are taken as the benchmark of rationality, in this chapter we add the noise only to the decision-making of the EMB investors.

Market Clearance The number of shares demanded by each investor is a monotonically decreasing function of the hypothetical price P_h (see [24]). As the total number of outstanding shares is N , the price of the stock at time t is given by the market clearance condition: P_t is the unique price at which the total demand for shares is equal to the total supply, N :

$$\sum_i N_h^i(P_t) = \sum_i \frac{x_h(P_t) W_h^i(P_t)}{P_t} = N, \quad (13)$$

where the summation is over all the investors in the market, RII investors as well as EMB investors.

Agent Based Simulation The market dynamics begin with a set of initial conditions which consist of an initial stock price P_0 , an initial dividend D_0 , the wealth and number of shares held by each investor at time $t = 0$, and an initial “history” of stock returns. As will become evident, the general results do not depend on the initial conditions. At the first period ($t = 1$), interest is paid on the bond, and the time 1 dividend $\tilde{D}_1 = D_0(1 + \tilde{z})$ is realized and paid out. Then investors submit their demand orders, $N_h^i(P_h)$, and the market clearing price P_1 is determined. After the clearing price is set, the new wealth and number of shares held by each investor are calculated. This completes one time period. This process is repeated over and over, as the market dynamics develop.

We would like to stress that even the simplified benchmark model, with only RII investors, is impossible to solve analytically. The reason for this is that the optimal investment proportion, $x_h(P_h)$, cannot be calculated analytically. This problem is very general and it is encountered with almost any choice of utility function and distribution of returns. One important exception is the case of a negative exponential utility function and normally distributed returns. Indeed, many models make these two assumptions for the sake of tractability. The problem with the assumption of negative exponential utility is that it implies Constant Absolute Risk Aversion (CARA), which is

very unrealistic, as it implies that investors choose to invest the same dollar amount in a risky prospect *independent of their wealth*. This is not only in sharp contradiction to the empirical evidence, but also excludes the investigation of the two-way interaction between wealth and price dynamics, which is crucial to the understanding of the market.

Thus, one contribution of the agent based simulation approach is that it allows investigation of models with realistic assumptions regarding investors' preferences. However, the main contribution of this method is that it permits us to investigate models which are much more complex (and realistic) than the benchmark model, in which all investors are RII. With the agent based simulation approach one can study models incorporating the empirically and experimentally documented investors' behavior, and the heterogeneity of investors.

Results of the LLS Model

We begin by describing the benchmark case where all investors are rational and identical. Then we introduce to the market EMB investors and investigate their affects on the market dynamics.

Benchmark Case: Fully Rational and Identical Agents

In this benchmark model all investors are RII: rational, informed and identical. Thus, it is not surprising that the benchmark model generates market dynamics which are typical of homogeneous rational agent models:

No Volume All investors in the model are identical; they therefore always agree on the optimal proportion to invest in the stock. As a consequence, all the investors always achieve the same return on their portfolio. This means that at any time period the ratio between the wealth of any two investors is equal to the ratio of their initial wealths, i. e.:

$$\frac{W_t^i}{W_t^j} = \frac{W_0^i}{W_0^j}. \quad (14)$$

As the wealth of investors is always in the same proportion, and as they always invest the same fraction of their wealth in the stock, the number of shares held by different investors is also always in the same proportion:

$$\frac{N_t^i}{N_t^j} = \frac{\frac{x_t W_t^i}{P_t}}{\frac{x_t W_t^j}{P_t}} = \frac{W_t^i}{W_t^j} = \frac{W_0^i}{W_0^j}. \quad (15)$$

Since the total supply of shares is constant, this implies that each investor always holds the same number of shares, and there is *no trading volume* (the number of shares held may

vary from one investor to the other as a consequence of different initial endowments).

Log-Prices Follow a Random Walk In the benchmark model all investors believe that next period's price will converge to the fundamental value given by the discounted dividend model (Eq. (3)). Therefore, the actual stock price is always close to the fundamental value. The fluctuations in the stock price are driven by fluctuations in the fundamental value, which in turn are driven by the fluctuating dividend realizations. As the dividend fluctuations are (by assumption) uncorrelated over time, one would expect that the price fluctuations will also be uncorrelated. To verify this intuitive result, we examine the return autocorrelations in simulations of the benchmark model.

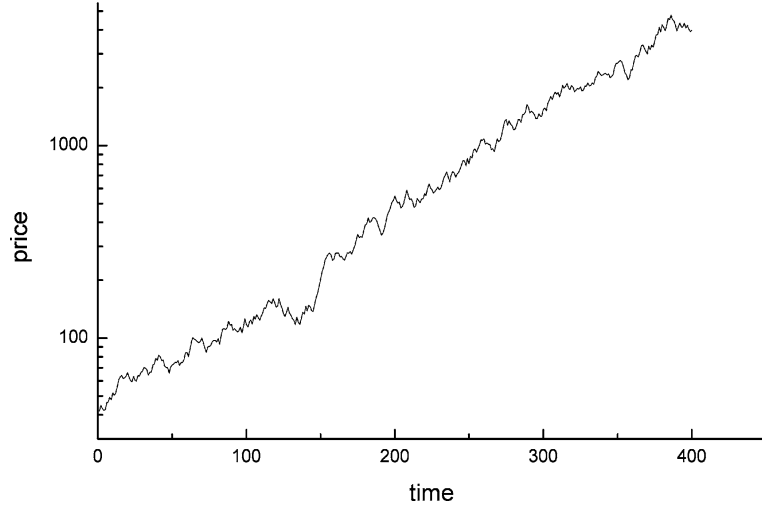
Let us turn to the simulation of the model. We first describe the parameters and initial conditions used in the simulation, and then report the results. We simulate the benchmark model with the following parameters:

- Number of investors = 1000
- Risk aversion parameter $\alpha = 1.5$. This value roughly conforms with the estimate of the risk aversion parameter found empirically and experimentally.
- Number of shares = 10,000.
- We take the time period to be a quarter, and accordingly we choose:
- Riskless interest rate $r_f = 0.01$.
- Required rate of return on stock $k = 0.04$.
- Maximal one-period dividend decrease $z_1 = -0.07$.
- Maximal one-period dividend growth $z_2 = 0.10$.
- \bar{z} is uniformly distributed between these values. Thus, the average dividend growth rate is $g = (z_1 + z_2)/2 = 0.015$.

Initial Conditions: Each investor is endowed at time $t = 0$ with a total wealth of \$1000, which is composed of 10 shares worth an initial price of \$50 per share, and \$500 in cash. The initial quarterly dividend is set at \$0.5 (for an annual dividend yield of about 4%). As will soon become evident, the dynamics are not sensitive to the particular choice of initial conditions.

Figure 1 shows the price dynamics in a typical simulation with these parameters (simulations with the same parameters differ one from the other because of the different random dividend realizations). Notice that the vertical axis in this figure is logarithmic. Thus, the roughly constant slope implies an approximately exponential price growth, or an approximately constant *average* return.

The prices in this simulation seem to fluctuate randomly around the trend. However, Fig. 1 shows only one simulation. In order to have a more rigorous analysis we



Agent Based Computational Economics, Figure 1
Price Dynamics in the Benchmark Model

perform many independent simulations, and employ statistical tools. Namely, for each simulation we calculate the autocorrelation of returns. We perform a univariate regression of the return in time t on the return on time $t - j$:

$$R_t = \alpha_j + \beta_j R_{t-j} + \varepsilon_t,$$

where R_t is the return in period t , and j is the lag. The autocorrelation of returns for lag j is defined as:

$$\rho_j = \frac{\text{cov}(R_t, R_{t-j})}{\hat{\sigma}^2(R)},$$

and it is estimated by $\hat{\beta}$. We calculate the autocorrelation for different lags, $j = 1, \dots, 40$. Figure 2 shows the average autocorrelation as a function of the lag, calculated over 100 independent simulations. It is evident both from the figure that the returns are uncorrelated in the benchmark model, conforming with the random-walk hypothesis.

No Excess Volatility Since the RII investors believe that the stock price will converge to the fundamental value next period, in the benchmark model prices are always close to the fundamental value given by the discounted dividend stream. Thus, we do not expect prices to be more volatile than the value of the discounted dividend stream. For a formal test of excess volatility we follow the technique in [37]. For each time period we calculate the actual price P_t , and the fundamental value of discounted dividend stream, P_t^f , as in Eq. (3). Since prices follow an upward trend, in order to have a meaningful measure of the volatility, we must detrend these price series. Following

Shiller, we run the regression:

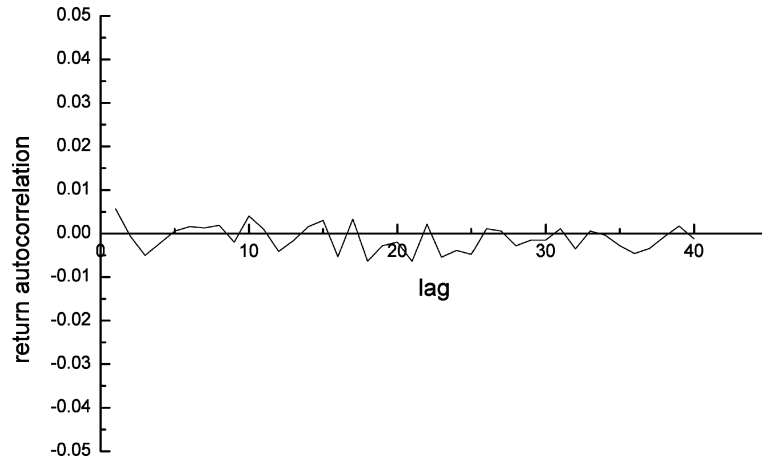
$$\ln P_t = bt + c + \varepsilon_t, \quad (16)$$

in order to find the average exponential price growth rate (where b and c are constants). Then, we define the detrended price as: $p_t = P_t / e^{bt}$. Similarly, we define the detrended value of the discounted dividend stream p_t^f , and compare $\sigma(p_t)$ with $\sigma(p_t^f)$. For 100 1000-period simulations we find an average $\sigma(p_t)$ of 22.4, and an average $\sigma(p_t^f)$ of 22.9. As expected, the actual price and the fundamental value have almost the same volatility.

To summarize the results obtained for the benchmark model, we find that when all investors are assumed to be rational, informed and identical, we obtain results which are typical of rational-representative-agent models: no volume, no return autocorrelations, and no excess volatility. We next turn to examine the effect of introducing into the market EMB investors, which model empirically and experimentally documented elements of investors' behavior.

The Introduction of a Small Minority of EMB Investors

In this section we will show that the introduction of a small minority of heterogeneous EMB investors generates many of the empirically observed market "anomalies" which are absent in the benchmark model, and indeed, in most other rational-representative-agent models. We take this as strong evidence that the "non-rational" elements of investor behavior which are documented in experimental studies, and the heterogeneity of investors, both of which



Agent Based Computational Economics, Figure 2
Return Autocorrelation in Benchmark Model

are incorporated in the LLS model, are crucial to understanding the dynamics of the market.

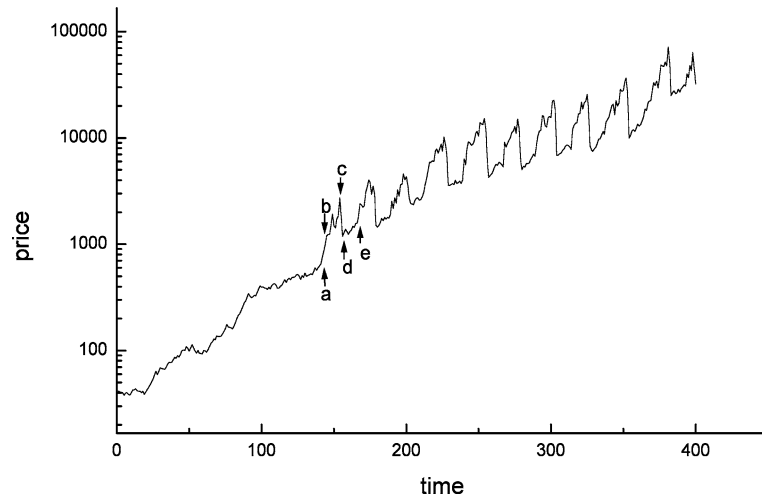
In presenting the results of the LLS model with EMB investors we take an incremental approach. We begin by describing the results of a model with a small sub-population of *homogeneous* EMB believers. This model produces the above mentioned market “anomalies”; however, it produces unrealistic cyclic market dynamics. Thus, this model is presented both for analyzing the source of the “anomalies” in a simplified setting, and as a reference point with which to compare the dynamics of the model with a *heterogeneous* EMB believer population.

We investigate the effects of investors’ heterogeneity by first analyzing the case in which there are two types of EMBs. The two types differ in the method they use to estimate the *ex-ante* return distribution. Namely, the first type looks at the set of the last m_1 *ex-post* returns, whereas the second type looks at the set of the last m_2 *ex-post* returns. It turns out that the dynamics in this case are much more complicated than a simple “average” between the case where all EMB investors have m_1 and the case where all EMB investors have m_2 . Rather, there is a complex non-linear interaction between the two EMB sub-populations. This implies that the heterogeneity of investors is a very important element determining the market dynamics, an element which is completely absent in representative-agent models.

Finally, we present the case where there is an entire spectrum of EMB investors differing in the number of *ex-post* observations they take into account when estimating the *ex-ante* distribution. This general case generates very realistic-looking market dynamics with all of the above mentioned market anomalies.

Homogeneous Sub-Population of EMBs When a very small sub-population of EMB investors is introduced to the benchmark LLS model, the market dynamics change dramatically. Figure 3 depicts a typical price path in a simulation of a market with 95% RII investors and 5% EMB investors. The EMB investors have $m = 10$ (i. e., they estimate the *ex-ante* return distribution by observing the set of the last 10 *ex-post* returns). σ , the standard deviation of the random noise affecting the EMBs’ decision making is taken as 0.2. All investors, RII and EMB alike, have the same risk aversion parameter $\alpha = 1.5$ (as before). In the first 150 trading periods the price dynamics look very similar to the typical dynamics of the benchmark model. However, after the first 150 or so periods the price dynamics change. From this point onwards the market is characterized by periodic booms and crashes. Of course, Fig. 3 describes only one simulation. However, as will become evident shortly, different simulations with the same parameters may differ in detail, but the pattern is general: at some stage (not necessarily after 150 periods) the EMB investors induce cyclic price behavior. It is quite astonishing that such a small minority of only 5% of the investors can have such a dramatic impact on the market.

In order to understand the periodic booms and crashes let us focus on the behavior of the EMB investors. After every trade, the EMB investors revise their estimation of the *ex-ante* return distribution, because the set of *ex-post* returns they employ to estimate the *ex-ante* distribution changes. Namely, investors add the latest return generated by the stock to this set and delete the oldest return from this set. As a result of this update in the estimation of the *ex-ante* distribution, the optimal investment proportion x^* changes, and EMB investors revise their portfolios at



Agent Based Computational Economics, Figure 3

5% of Investors are Efficient Market Believers, 95% Rational Informed Investors

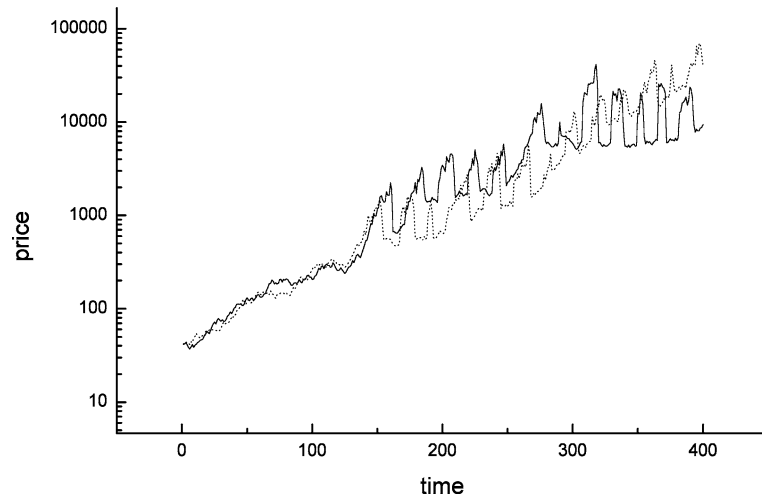
next period's trade. During the first 150 or so periods, the informed investors control the dynamics and the returns fluctuate randomly (as in the benchmark model). As a consequence, the investment proportion of the EMB investors also fluctuates irregularly. Thus, during the first 150 periods the EMB investors do not effect the dynamics much. However, at point **a** the dynamics change qualitatively (see Fig. 3). At this point, a relatively high dividend is realized, and as a consequence, a relatively high return is generated. This high return leads the EMB investors to increase their investment proportion in the stock at the next trading period. This increased demand of the EMB investors is large enough to effect next period's price, and thus a second high return is generated. Now the EMB investors look at a set of *ex-post* returns with two high returns, and they increase their investment proportion even further. Thus, a positive feedback loop is created.

Notice that as the price goes up, the informed investors realize that the stock is overvalued relative to the fundamental value P^f and they decrease their holdings in the stock. However, this effect does not stop the price increase and break the feedback loop because the EMB investors continue to buy shares aggressively. The positive feedback loop pushes the stock price further and further up to point **b**, at which the EMBs are invested 100% in the stock. At point **b** the positive feedback loop "runs out of gas". However, the stock price remains at the high level because the EMB investors remain fully invested in the stock (the set of past $m=10$ returns includes at this stage the very high returns generated during the "boom" – segment **a–b** in Fig. 3).

When the price is at the high level (segment **b–c**), the dividend yield is low, and as a consequence, the returns are generally low. As time goes by and we move from point **b** towards point **c**, the set of $m = 10$ last returns gets filled with low returns. Despite this fact, the extremely high returns generated in the boom are also still in this set, and they are high enough to keep the EMB investors fully invested. However, 10 periods after the boom, these extremely high returns are pushed out of the set of relevant *ex-post* returns. When this occurs, at point **c**, the EMB investors face a set of low returns, and they cut their investment proportion in the stock sharply. This causes a dramatic crash (segment **c–d**). Once the stock price goes back down to the "fundamental" value, the informed investors come back into the picture. They buy back the stock and stop the crash.

The EMB investors stay away from the stock as long as the *ex-post* return set includes the terrible return of the crash. At this stage the informed investors regain control of the dynamics and the stock price remains close to its fundamental value. 10 periods after the crash the extremely negative return of the crash is excluded from the *ex-post* return set, and the EMB investors start increasing their investment proportion in the stock (point **e**). This drives the stock price up, and a new boom-crash cycle is initiated. This cycle repeats itself over and over almost periodically.

Figure 3 depicts the price dynamics of a single simulation. One may therefore wonder how general the results discussed above are. Figure 4 shows two more simulations with the same parameters but different dividend realiza-



Agent Based Computational Economics, Figure 4
Two More Simulations – same Parameters as Fig. 3, Different Dividend Realizations

tions. It is evident from this figure that although the simulations vary in detail (because of the different dividend realizations), the overall price pattern with periodic boom-crash cycles is robust.

Although these dynamics are very unrealistic in terms of the periodicity, and therefore the predictability of the price, they do shed light on the mechanism generating many of the empirically observed market phenomena. In the next section, when we relax the assumption that the EMB population is homogeneous with respect to m , the price is no longer cyclic or predictable, yet the mechanisms generating the market phenomena are the same as in this homogeneous EMB population case. The homogeneous EMB population case generates the following market phenomena:

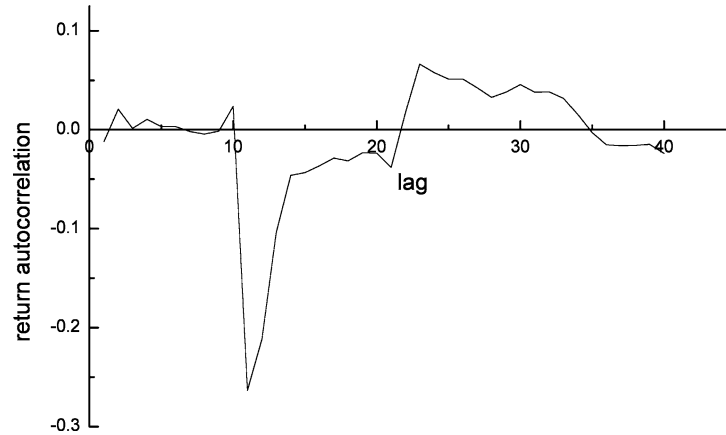
Heavy Trading Volume As explained above, shares change hands continuously between the RII investors and the EMB investors. When a “boom” starts the RII investors observe higher *ex-post* returns and become more optimistic, while the RII investor view the stock as becoming overpriced and become more pessimistic. Thus, at this stage the EMBs buy most of the shares from the RIIs. When the stock crashes, the opposite is true: the EMBs are very pessimistic, but the RII investors buy the stock once it falls back to the fundamental value. Thus, there is substantial trading volume in this market. The average trading volume in a typical simulation is about 1000 shares per period, which are 10% of the total outstanding shares.

Autocorrelation of Returns The cyclic behavior of the price yields a very definite return autocorrelation pat-

tern. The autocorrelation pattern is depicted graphically in Fig. 5. The autocorrelation pattern is directly linked to the length of the price cycle, which in turn are determined by m . Since the moving window of *ex-post* returns used to estimate the *ex-ante* distribution is $m = 10$ periods long, the price cycles are typically a little longer than 20 periods long: a cycle consists of the positive feedback loop (segment **a–b** in Fig. 3) which is about 2–3 periods long, the upper plateau (segment **b–c** in Fig. 3) which is about 10 periods long, the crash that occurs during one or two periods, and the lower plateau (segment **d–e** in Fig. 3) which is again about 10 periods long, for a total of about 23–25 periods. Thus, we expect positive autocorrelation for lags of about 23–25 periods, because this is the lag between one point and the corresponding point in the next (or previous) cycle. We also expect negative autocorrelation for lags of about 10–12 periods, because this is the lag between a boom and the following (or previous) crash, and vice versa. This is precisely the pattern we observe in Fig. 5.

Excess Volatility The EMB investors induce large deviations of the price from the fundamental value. Thus, price fluctuations are caused not only by dividend fluctuations (as the standard theory suggests) but also by the endogenous market dynamics driven by the EMB investors. This “extra” source of fluctuations causes the price to be more volatile than the fundamental value P^f .

Indeed, for 100 1000-period independent simulations with 5% EMB investors we find an average $\sigma(p_t)$ of 46.4, and an average $\sigma(p_t^f)$ of 30.6; i. e., we have excess volatility of about 50%.



Agent Based Computational Economics, Figure 5

Return Autocorrelation 5%, Efficient Market Believers, $m = 10$

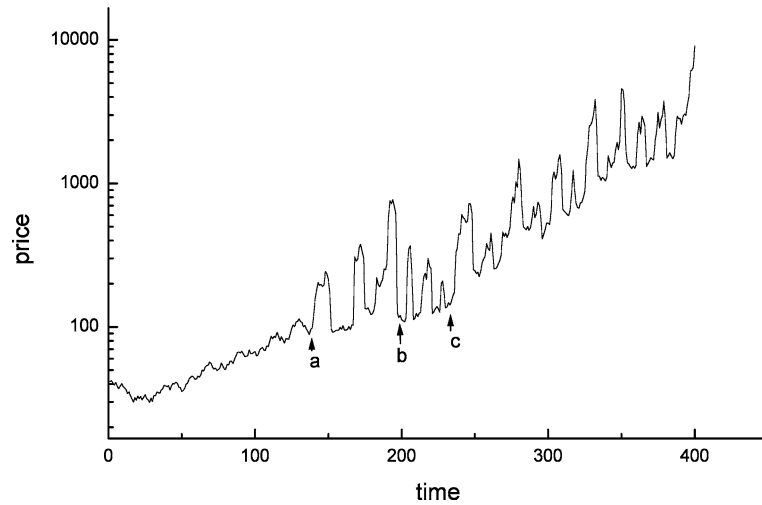
As a first step in analyzing the effects of heterogeneity of the EMB population, in the next section we examine the case of two types of EMB investors. We later analyze a model in which there is a full spectrum of EMB investors.

Two Types of EMBs One justification for using a representative agent in economic modeling is that although investors are heterogeneous in reality, one can model their collective behavior with one representative or “average” investor. In this section we show that this is generally not true. Many aspects of the dynamics result from the non-linear interaction between different investor types. To illustrate this point, in this section we analyze a very simple case in which there are only two types of EMB investors: one with $m = 5$ and the other with $m = 15$. Each of these two types consists of 2% of the investor population, and the remaining 96% are informed investors. The representative agent logic may tempt us to think that the resulting market dynamics would be similar to that of one “average” investor, i. e. an investor with $m = 10$. Figure 6 shows that this is clearly not the case. Rather than seeing periodic cycles of about 23–25 periods (which correspond to the average m of 10, as in Fig. 3), we see an irregular pattern. As before, the dynamics are first dictated by the informed investors. Then, at point **a**, the EMB investors with $m = 15$ induce cycles which are about 30 periods long. At point **b** there is a transition to shorter cycles induced by the $m = 5$ population, and at point **c** there is another transition back to longer cycles. What is going on?

These complex dynamics result from the non-linear interaction between the different sub-populations. The transitions from one price pattern to another can be partly understood by looking at the wealth of each sub-

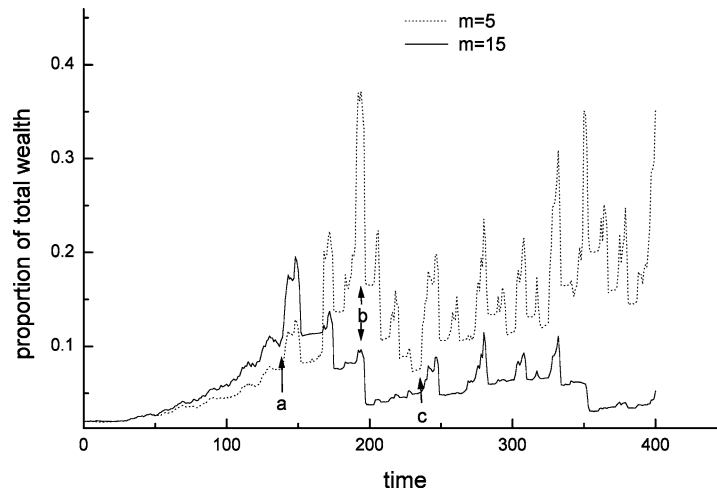
population. Figure 7 shows the proportion of the total wealth held by each of the two EMB populations (the remaining proportion is held by the informed investors). As seen in Fig. 7, the cycles which start at point **a** are dictated by the $m = 15$ rather than the $m = 5$ population, because at this stage the $m = 15$ population controls more of the wealth than the $m = 5$ population. However, after 3 cycles (at point **b**) the picture is reversed. At this point the $m = 5$ population is more powerful than the $m = 15$ population, and there is a transition to shorter boom-crash cycles. At point **c** the wealth of the two sub-populations is again almost equal, and there is another transition to longer cycles. Thus, the complex price dynamics can be partly understood from the wealth dynamics. But how are the wealth dynamics determined? Why does the $m = 5$ population become wealthier at point **b**, and why does it lose most of this advantage at point **c**? It is obvious that the wealth dynamics are influenced by the price dynamics, thus there is a complicated two-way interaction between the two. Although this interaction is generally very complex, some principle ideas about the mutual influence between the wealth and price patterns can be formulated. For example, a population that becomes dominant and dictates the price dynamics, typically starts under-performing, because it affects the price with its actions. This means pushing the price up when buying, and therefore buying high, and pushing the price down when selling. However, a more detailed analysis must consider the specific investment strategy employed by each population. For a more comprehensive analysis of the interaction between heterogeneous EMB populations see [25].

The two EMB population model generates the same market phenomena as did the homogeneous population



Agent Based Computational Economics, Figure 6

2% EMB $m = 5$, 2% EMB $m = 15$, 96% RII



Agent Based Computational Economics, Figure 7

Proportion of the total wealth held by the two EMB populations

case: heavy trading volume, return autocorrelations, and excess volatility. Although the price pattern is much less regular in the two-EMB-population case, there still seems to be a great deal of predictability about the prices. Moreover, the booms and crashes generated by this model are unrealistically dramatic and frequent. In the next section we analyze a model with a continuous spectrum of EMB investors. We show that this fuller heterogeneity of investors leads to very realistic price and volume patterns.

Full Spectrum of EMB Investors Up to this point we have analyzed markets with at most three different sub-

populations (one RII population and two EMB populations). The market dynamics we found displayed the empirically observed market anomalies, but they were unrealistic in the magnitude, frequency, and semi-predictability of booms and crashes. In reality, we would expect not only two or three investor types, but rather an entire spectrum of investors. In this section we consider a model with a full spectrum of different EMB investors. It turns out that *more is different*. When there is an entire range of investors, the price dynamics become realistic: booms and crashes are not periodic or predictable, and they are also less frequent and dramatic. At the same time, we still obtain all of the market anomalies described before.

In this model each investor has a different number of *ex-post* observations which he utilizes to estimate the *ex-ante* distribution. Namely, investor i looks at the set of the m^i most recent returns on the stock, and we assume that m^i is distributed in the population according to a truncated normal distribution with average \bar{m} and standard deviation σ_m (as $m \leq 0$ is meaningless, the distribution is truncated at $m = 0$).

Figure 8 shows the price pattern of a typical simulation of this model. In this simulation 90% of the investors are RII, and the remaining 10% are heterogeneous EMB investors with $\bar{m} = 40$, and $\sigma_m = 10$. The price pattern seems very realistic with “smoother” and more irregular cycles. Crashes are dramatic, but infrequent and unpredictable.

The heterogeneous EMB population model generates the following empirically observed market phenomena:

Return Autocorrelation: Momentum and Mean-Reversion

In the heterogeneous EMB population model trends are generated by the same positive feedback mechanism that generated cycles in the homogeneous case: high (low) returns tend to make the EMB investors more (less) aggressive, this generates more high (low) returns, etc. The difference between the two cases is that in the heterogeneous case there is a very complicated interaction between all the different investor sub-populations and as a result there are no distinct regular cycles, but rather, smoother and more irregular trends. There is no single cycle length – the dynamics are a combination of many different cycles. This makes the autocorrelation pattern also smoother and more continuous. The return autocorrelations in the heterogeneous model are shown in Fig. 9. This autocorrelation pattern conforms with the empirical findings. In the short-run (lags 1–4) the autocorrelation is positive – this is the empirically documented phenomena known as momentum: in the short-run, high returns tend to be followed by more high returns, and low returns tend to be followed by more low returns. In the longer-run (lags 5–13) the autocorrelation is negative, which is known as mean-reversion. For even longer lags the autocorrelation eventually tends to zero. The short-run momentum, longer-run mean-reversion, and eventual diminishing autocorrelation creates the general “U-shape” which is found in empirical studies [7,13,31] and which is seen in Fig. 9.

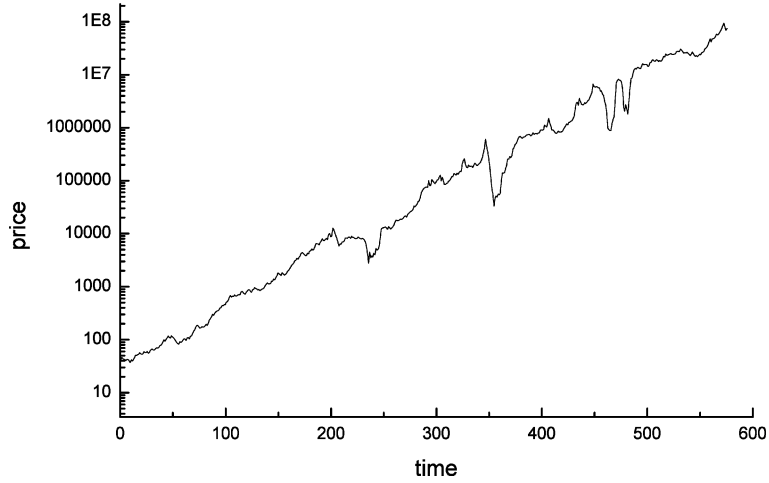
Excess Volatility The price level is generally determined by the fundamental value of the stock. However, as in the homogeneous EMB population case, the EMB investors occasionally induce temporary departures of the price away from the fundamental value. These temporary de-

partures from the fundamental value make the price more volatile than the fundamental value. Following Shiller’s methodology we define the detrended price, p , and fundamental value, p^f . Averaging over 100 independent simulations we find $\sigma(p) = 27.1$ and $\sigma(p^f) = 19.2$, which is an excess volatility of 41%.

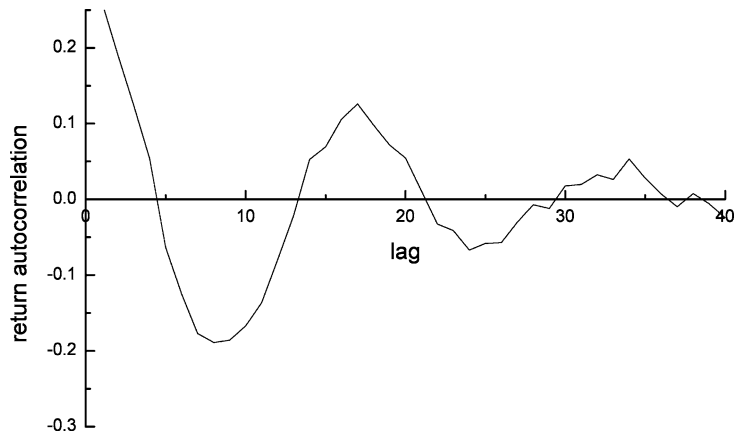
Heavy Volume As investors in our model have different information (the informed investors know the dividend process, while the EMB investors do not), and different ways of interpreting the information (EMB investors with different memory spans have different estimations regarding the *ex-ante* return distribution), there is a high level of trading volume in this model. The average trading volume in this model is about 1700 shares per period (17% of the total outstanding shares). As explained below, the volume is positively correlated with contemporaneous and lagged absolute returns.

Volume is Positively Correlated with Contemporaneous and Lagged Absolute Returns

Investors revise their portfolios as a result of changes in their beliefs regarding the future return distribution. The changes in the beliefs can be due to a change in the current price, to a new dividend realization (in the case of the informed investors), or to a new observation of an *ex-post* return (in the case of the EMB investors). If all investors change their beliefs in the same direction (for example, if everybody becomes more optimistic), the stock price can change substantially with almost no volume – everybody would like to increase the proportion of the stock in his portfolio, this will push the price up, but a very small number of shares will change hands. This scenario would lead to zero or perhaps even negative correlation between the magnitude of the price change (or return) and the volume. However, the typical scenario in the LLS model is different. Typically, when a positive feedback trend is induced by the EMB investors, the opinions of the informed investors and the EMB investors change in opposite directions. The EMB investors see a trend of rising prices as a positive indication about the *ex-ante* return distribution, while the informed investors believe that the higher the price level is above the fundamental value, the more overpriced the stock is, and the harder it will eventually fall. The exact opposite holds for a trend of falling prices. Thus, price trends are typically interpreted differently by the two investor types, and therefore induce heavy trading volume. The more pronounced the trend, the more likely it is to lead to heavy volume, and at the same time, to large price changes which are due to the positive feedback trading on behalf of the EMB investors.



Agent Based Computational Economics, Figure 8
Spectrum of Heterogeneous EMB Investors (10% EMB Investors, 90% RII Investors)



Agent Based Computational Economics, Figure 9
Return Autocorrelation – Heterogeneous EMB Population

This explains not only the positive correlation between volume and contemporaneous absolute rates of return, but also the positive correlation between volume and lagged absolute rates of return. The reason is that the behavior of the EMB investors induces short-term positive return autocorrelation, or momentum (see above). That is, a large absolute return this period is associated not only with high volume this period, but also with a large absolute return next period, and therefore with high volume next period. In other words, when there is a substantial price increase (decrease), EMB investors become more (less) aggressive and the opposite happens to the informed traders. As we have seen before, when a positive feedback loop is started, the EMB investors are more dominant in determining the

price, and therefore another large price increase (decrease) is expected next period. This large price change is likely to be associated with heavy trading volume as the opinions of the two populations diverge. Furthermore, this large increase (decrease) is expected to make the EMB investors even more optimistic (pessimistic) leading to another large price increase (decrease) and heavy volume next period.

In order to verify this relationship quantitatively, we regress volume on contemporaneous and lagged absolute rates of return for 100 independent simulations. We run the regressions:

$$\begin{aligned} V_t &= \alpha + \beta_C |R_t - 1| + \varepsilon_t, \quad \text{and} \\ V_t &= \alpha + \beta_L |R_{t-1} - 1| + \varepsilon_t, \end{aligned} \quad (17)$$

where V_t is the volume at time t and R_t is the total return on the stock at time t , and the subscripts C and L stand for contemporaneous and lagged. We find an average value of 870 for $\hat{\beta}_C$ with an average t -value of 5.0 and an average value of 886 for $\hat{\beta}_L$ with an average t -value of 5.1.

Discussion of the LLS Results The LLS model is an Agent Based Simulation model of the stock market which incorporates some of the fundamental experimental findings regarding the behavior of investors. The main non-standard assumption of the model is that there is a small minority of investors in the market who are uninformed about the dividend process and who believe in market efficiency. The investment decision of these investors is reduced to the optimal diversification between the stock and the bond.

The LLS model generates many of the empirically documented market phenomena which are hard to explain in the analytical rational-representative-agent framework. These phenomena are:

- Short term momentum;
- Longer term mean reversion;
- Excess volatility;
- Heavy trading volume;
- Positive correlation between volume and contemporaneous absolute returns;
- Positive correlation between volume and lagged absolute returns;
- Endogenous market crashes.

The fact that so many “puzzles” are explained with a simple model built on a small number of empirically documented behavioral elements leads us to suspect that these behavioral elements are very important in understanding the workings of the market. This is especially true in light of the observations that a very small minority of the non-standard bounded-rational investors can have a dramatic influence on the market, and that these investors are not wiped out by the majority of rational investors.

Summary and Future Directions

Standard economic models typically describe a world of homogeneous rational agents. This approach is the foundation of most of our present day knowledge in economic theory. With the Agent Based Simulation approach we can investigate a much more complex and “messy” world with different agent types, who employ different strategies to try to survive and prosper in a market with structural uncertainty. Agents can learn over time, from their own experience and from their observation about the performance of

other agents. They co-evolve over time and as they do so, the market dynamics change continuously. This is a world view closer to biology, than it is to the “clean” realm of physical laws which classical economics has aspired to.

The Agent Based approach should not and can not replace the standard analytical economic approach. Rather, these two methodologies support and complement each other: When an analytical model is developed, it should become standard practice to examine the robustness of the model’s results with agent based simulations. Similarly, when results emerge from agent based simulation, one should try to understand their origin and their generality, not only by running many simulations, but also by trying to capture the essence of the results in a simplified analytical setting (if possible).

Although the first steps in economic agent based simulations were made decades ago, economics has been slow and cautious to adopt this new methodology. Only in recent years has this field begun to bloom. It is my belief and hope that the agent based approach will prove as fruitful in economics as it has been in so many other branches of science.

Bibliography

Primary Literature

1. Admati A, Pfleiderer P (1988) A theory of intraday patterns: Volume and price variability. *Rev Financ Stud* 1:3–40
2. Arthur WB (1994) Inductive reasoning and bounded rationality (The El Farol problem). *Am Econ Rev* 84:406–411
3. Arthur WB, Holland JH, Lebaron B, Palmer RG, Tayler P (1997) Asset pricing under endogenous expectations in an artificial stock market. In: Arthur WB, Durlauf S, Lane D (eds) *The economy as an evolving complex system II*. Addison-Wesley, Redwood City
4. Brock WA, Hommes CA (1998) Heterogeneous beliefs and routes to chaos in a simple asset pricing model. *J Econ Dyn Control* 22:1235–1274
5. Egenter E, Lux T, Stauffer D (1999) Finite size effects in Monte Carlo Simulations of two stock market models. *Physica A* 268:250–256
6. Epstein JM, Axtell RL (1996) Complex adaptive systems. In: *Growing artificial societies: Social science from the bottom up*. MIT Press, Washington DC
7. Fama E, French K (1988) Permanent and temporary components of stock prices. *J Political Econ* 96:246–273
8. Friend I, Blume ME (1975) The demand for risky assets. *Am Econ Rev* 65:900–922
9. Gordon J, Paradis GE, Rorke CH (1972) Experimental evidence on alternative portfolio decision rules. *Am Econ Rev* 62(1):107–118
10. Grossman S, Stiglitz J (1980) On the impossibility of informationally efficient markets. *Am Econ Rev* 70:393–408
11. Hellthaler T (1995) The influence of investor number on a microscopical market. *Int J Mod Phys C* 6:845–852

12. Hommes CH (2002) Modeling the stylized facts in finance through simple nonlinear adaptive systems. *PNAS* 99:7221–7228
13. Jegadeesh N, Titman S (1993) Returns to buying winners and selling losers: Implications for stock market efficiency. *J Finance* 48:65–91
14. Karpoff J (1987) The relationship between price changes and trading volume: A survey. *J Financ Quantitative Anal* 22:109–126
15. Kim GW, Markowitz HM (1989) Investment rules, margin, and market volatility. *J Portf Manag* 16:45–52
16. Kirman AP (1992) Whom or what does the representative agent represent? *J Econ Perspectiv* 6:117–136
17. Kohl R (1997) The influence of the number of different stocks on the Levy, Levy Solomon model. *Int J Mod Phys C* 8:1309–1316
18. Kroll Y, Levy H, Rapoport A (1988) Experimental tests of the separation theorem and the capital asset pricing model. *Am Econ Rev* 78:500–519
19. LeBaron B (2000) Agent-based computational finance: Suggested readings and early research. *J Econ Dyn Control* 24:679–702
20. Levy H (1994) Absolute and relative risk aversion: An experimental study. *J Risk Uncertain* 8:289–307
21. Levy H, Lim KC (1998) The economic significance of the cross-sectional autoregressive model: Further analysis. *Rev Quant Finance Acc* 11:37–51
22. Levy M, Levy H (1996) The danger of assuming homogeneous expectations. *Financ Analyst J* 52:65–70
23. Levy M, Levy H, Solomon S (1994) A microscopic model of the stock market: Cycles, booms, and crashes. *Econ Lett* 45:103–111
24. Levy M, Levy H, Solomon S (2000) *Microscopic simulation of financial markets*. Academic Press, San Diego
25. Levy M, Persky N, Solomon S (1996) The complex dyn of a simple stock market model. *Int J High Speed Comput* 8:93–113
26. Lux T (1995) Herd behaviour, bubbles and crashes. *Econ J* 105:881
27. Lux T (1998) The socio-economic dynamics of speculative bubbles: Interacting agents, chaos, and the fat tails of returns distributions. *J Econ Behav Organ* 33:143–165
28. Lux T, Marchesi M (1999) Volatility clustering in financial markets: A micro-simulation of interacting agents. *Nature* 397:498
29. Orcutt GH, Caldwell SB, Wertheimer R (1976) *Policy exploration through microanalytic simulation*. The Urban Institute, Washington DC
30. Palmer RG, Arthur WB, Holland JH, LeBaron B, Tayler P (1994) Artificial economic life: A simple model of a stock market. *Physica D* 75:264–274
31. Poterba JM, Summers LH (1988) Mean reversion in stock returns: Evidence and implications. *J Financial Econ* 22:27–59
32. Samanidou E, Zschischang E, Stauffer D, Lux T (2007) Agent-based models of financial markets. *Rep Prog Phys* 70:409–450
33. Samuelson PA (1989) The judgement of economic science on rational portfolio management: Timing and long horizon effects. *J Portfolio Manag* 16:4–12
34. Samuelson PA (1994) The long term case for equities and how it can be oversold. *J Portf Management* 21:15–24
35. Sargent T (1993) *Bounded rationality and macroeconomics*. Oxford University Press, Oxford
36. Schelling TC (1978) *Micro motives and macro behavior*. Norton & Company, New York
37. Shiller RJ (1981) Do stock prices move too much to be justified by subsequent changes in dividends? *Am Econ Rev* 71:421–436
38. Stauffer D, de Oliveira PMC, Bernardes AT (1999) Monte Carlo Simulation of volatility correlation in microscopic market model. *Int J Theor Appl Finance* 2:83–94
39. Tesfatsion L (2002) Agent-based computational economics: Growing economies from the bottom up. *Artif Life* 8:55–82
40. Tesfatsion L (2001) Special issue on agent-based computational economics. *J Econ Dyn Control* 25:281–293
41. Thaler R (ed) (1993) *Advances in behavioral finance*. Russell Sage Foundation, New York
42. Thaler R (1994) *Quasi rational economics*. Russell Sage Foundation, New York
43. Tversky A, Kahneman D (1981) The framing of decisions and the psychology of choice. *Science* 211:453–480
44. Tversky A, Kahneman D (1986) Rational choice and the framing of decision. *J Bus* 59(4):251–278
45. Tversky A, Kahneman D (1992) Advances in prospect theory: Cumulative representation of uncertainty. *J Risk Uncertain* 5:297–323

Books and Reviews

- Anderson PW, Arrow J, Pines D (eds) (1988) *The economy as an evolving complex system*. Addison-Wesley, Redwood City
- Axelrod R (1997) *The complexity of cooperation: Agent-based models of conflict and cooperation*. Princeton University Press, Princeton
- Moss de Oliveira S, de Oliveira H, Stauffer D (1999) *Evolution, money, war and computers*. BG Teubner, Stuttgart-Leipzig
- Solomon S (1995) The microscopic representation of complex macroscopic phenomena. In: Stauffer D (ed) *Annual Rev Comput Phys II*. World Scientific, Singapore

Agent Based Modeling and Artificial Life

CHARLES M. MACAL

Center for Complex Adaptive Agent Systems Simulation (CAS2), Decision and Information Sciences Division, Argonne National Laboratory, Argonne, USA

Article Outline

[Glossary](#)

[Definition of the Subject](#)

[Introduction](#)

[Artificial Life](#)

[Alife in Agent-Based Modeling](#)

[Future Directions](#)

[Bibliography](#)

Glossary

- Adaptation** The process by which organisms (agents) change their behavior or by which populations of agents change their collective behaviors with respect to their environment.
- Agent-based modeling (ABM)** A modeling and simulation approach applied to a complex system or complex adaptive system, in which the model is comprised of a large number of interacting elements (agents).
- Ant colony optimization (ACO)** A heuristic optimization technique motivated by collective decision processes followed by ants in foraging for food.
- Artificial chemistry** Chemistry based on the information content and transformation possibilities of molecules.
- Artificial life (ALife)** A field that investigates life's essential qualities primarily from an information content perspective.
- Artificial neural network (ANN)** A heuristic optimization and simulated learning technique motivated by the neuronal structure of the brain.
- Autocatalytic set** A closed set of chemical reactions that is self-sustaining.
- Autonomous** The characteristic of being capable of making independent decisions over a range of situations.
- Avida** An advanced artificial life computer program developed by Adami [1] and others that models populations of artificial organisms and the essential features of life such as interaction and replication.
- Biologically-inspired computational algorithm**
Any kind of algorithm that is based on biological metaphors or analogies.
- Cellular automata (CA)** A mathematical construct and technique that models a system in discrete time and discrete space in which the state of a cell depends on transition rules and the states of neighboring cells.
- Coevolution** A process by which many entities adapt and evolve their behaviors as a result of mutually effective interactions.
- Complex system** A system comprised of a large number of strongly interacting components (agents).
- Complex adaptive system (CAS)** A system comprised of a large number of strongly interacting components (agents) that adapt at the individual (agent) level or collectively at the population level.
- Decentralized control** A feature of a system in which the control mechanisms are distributed over multiple parts of the system.
- Digital organism** An entity that is represented by its essential information-theoretic elements (genomes) and implemented as a computational algorithm or model.
- Downward causation** The process by which a higher-order emergent structure takes on its own emergent behaviors and these behaviors exert influence on the constituent agents of the emergent structure.
- Dynamic network analysis** Network modeling and analysis in which the structure of the network, i. e., nodes (agents) and links (agent interactions) is endogenous to the model.
- Echo** An artificial life computer program developed by Holland [36] that models populations of complex adaptive systems and the essential features of adaptation in nature.
- Emergence** The process by which order is produced in nature.
- Entropy** A measure of order, related to the information needed to specify a system and its state.
- Evolution (artificial)** The process by which a set of instructions is transmitted and changes over successive generations.
- Evolutionary game** A repeated game in which agents adapt their strategies in recognition of the fact that they will face their opponents in the future.
- Evolution strategies** A heuristic optimization technique motivated by the genetic operation of selection and mutation.
- Evolutionary algorithm** Any algorithm motivated by the genetic operations including selection, mutation, crossover, etc.
- Evolutionary computing** A field of computing based on the use of evolutionary algorithms.
- Finite state machine** A mathematical model consisting of entities with a finite (usually small) number of possible discrete states.
- Game of life, life** A cellular automaton developed by Conway [8] that illustrates a maximally complex system based on simple rules.
- Generative social science** Social science investigation with the goal of understanding how social processes emerge out of social interaction.
- Genetic algorithm (GA)** A specific kind of evolutionary algorithm motivated by the genetic operations of selection, mutation, and crossover.
- Genetic programming (GP)** A specific kind of evolutionary algorithm that manipulates symbols according to prescribed rules, motivated by the genetic operations of selection, mutation, and crossover.
- Genotype** A set of instructions for a developmental process that creates a complex structure, as in a genotype for transmitting genetic information and seeding a developmental process leading to a phenotype.
- Hypercycle** A closed set of functional relations that is

self-sustaining, as in an autocatalytic chemical reaction network.

Individual-based model An approach originating in ecology to model populations of agents that emphasizes the need to represent diversity among individuals.

Langton's ant An example of a very simple computational program that computes patterns of arbitrary complexity after an initial series of simple structures.

Langton's loop An example of a very simple computational program that computes replicas of its structures according to simple rules applied locally as in cellular automata.

Learning classifier system (LCS) A specific algorithmic framework for implementing an adaptive system by varying the weights applied to behavioral rules specified for individual agents.

Lindenmeyer system (L-system)

A formal grammar, which is a set of rules for rewriting strings of symbols.

Machine learning A field of inquiry consisting of algorithms for recognizing patterns in data (e.g., data mining) through various computerized learning techniques.

Mind-body problem A field of inquiry that addresses how human consciousness arises out of material processes, and whether consciousness is the result of a logical-deductive or algorithmic process.

Meme A term coined by Dawkins [17] to refer to the minimal encoding of cultural information, similar to the genome's role in transmitting genetic information.

Particle swarm optimization An optimization technique similar to ant colony optimization, based on independent particles (agents) that search a landscape to optimize a single objective or goal.

Phenotype The result of an instance of a genotype interacting with its environment through a developmental process.

Reaction-diffusion system A system that includes mechanisms for both attraction and transformation (e.g., of agents) as well as repulsion and diffusion.

Recursively generated object An object that is generated by the repeated application of simple rules.

Self-organization A process by which structure and organization arise from within the endogenous instructions and processes inherent in an entity.

Self-replication The process by which an agent (e.g., organism, machine, etc.) creates a copy of itself that contains instructions for both the agent's operation and its replication.

Social agent-based modeling Agent-based modeling applied to social systems, generally applied to people and human populations, but also animals.

Social network analysis (SNA) A collection of techniques and approaches for analyzing networks of social relationships.

Stigmergy The practice of agents using the environment as a tool for communication with other agents to supplement direct agent-to-agent communication.

Swarm An early agent-based modeling toolkit designed to model artificial life applications.

Swarm intelligence Collective intelligence based on the actions of a set of interacting agents behaving according to a set of prescribed simple rules.

Symbolic processing A computational technique that consists of processing symbols rather than strictly numerical data.

Sugarscape An abstract agent-based model of artificial societies developed by Epstein and Axtell [26] to investigate the emergence of social processes.

Tierra An early artificial life computer program developed by Ray [59] that models populations of artificial organisms and the essential features of life such as interaction and replication.

Universal Turing machine (UTM) An abstract representation of the capabilities of any computable system.

Update rule A rule or transformation directive for changing or updating the state of an entity or agent, as for example updating the state of an agent in an agent-based model or updating the state of an L-system.

Definition of the Subject

Agent-based modeling began as the computational arm of artificial life some 20 years ago. Artificial life is concerned with the emergence of order in nature. How do systems self-organize themselves and spontaneously achieve a higher-ordered state? Agent-based modeling then, is concerned with exploring and understanding the processes that lead to the emergence of order through computational means. The essential features of artificial life models are translated into computational algorithms through agent-based modeling. With its historical roots in artificial life, agent-based modeling has become a distinctive form of modeling and simulation. Agent-based modeling is a bottom-up approach to modeling complex systems by explicitly representing the behaviors of large numbers of agents and the processes by which they interact. These essential features are all that is needed to produce at least rudimentary forms of emergent behavior at the systems level. To understand the current state of agent-based mod-

eling and where the field aspires to be in the future, it is necessary to understand the origins of agent-based modeling in artificial life.

Introduction

The field of Artificial Life, or “ALife,” is intimately connected to Agent-Based Modeling, or “ABM.” Although one can easily enumerate some of life’s distinctive properties, such as reproduction, respiration, adaptation, emergence, etc., a precise definition of life remains elusive.

Artificial Life had its inception as a coherent and sustainable field of investigation at a workshop in the late 1980s [43]. This workshop drew together specialists from diverse fields who had been working on related problems in different guises, using different vocabularies suited to their fields.

At about the same time, the introduction of the personal computer suddenly made computing accessible, convenient, inexpensive and compelling as an experimental tool. The future seemed to have almost unlimited possibilities for the development of ALife computer programs to explore life and its possibilities. Thus several ALife software programs emerged that sought to encapsulate the essential elements of life through incorporation of ALife-related algorithms into easily usable software packages that could be widely distributed. Computational programs for modeling populations of digital organisms, such as Tierra, Avida, and Echo, were developed along with more general purpose agent-based simulators such as Swarm.

Yet, the purpose of ALife was never restricted to understanding or re-creating life as it exists today. According to Langton:

Artificial systems which exhibit lifelike behaviors are worthy of investigation on their own right, whether or not we think that the processes that they mimic have played a role in the development or mechanics of life as we know it to be. Such systems can help us expand our understanding of life as it could be. (p. xvi in [43])

The field of ALife addresses life-like properties of systems at an abstract level by focusing on the information content of such systems independent of the medium in which they exist, whether it be biological, chemical, physical or *in silico*. This means that computation, modeling, and simulation play a central role in ALife investigations.

The relationship between ALife and ABM is complex. A case can be made that the emergence of ALife as a field was essential to the creation of agent-based modeling. Computational tools were both required and became pos-

sible in the 1980s for developing sophisticated models of digital organisms and general purpose artificial life simulators. Likewise, a case can be made that the possibility for creating agent-based models was essential to making ALife a promising and productive endeavor. ABM made it possible to understand the logical outcomes and implications of ALife models and life-like processes. Traditional analytical means, although valuable in establishing baseline information, were limited in their capabilities to include essential features of ALife. Many threads of ALife are still intertwined with developments in ABM and vice versa. Agent-based models demonstrate the emergence of life-like features using ALife frameworks; ALife algorithms are widely used in agent-based models to represent agent behaviors. These threads are explored in this article. In ALife terminology, one could say that ALife and ABM have co-evolved to their present states. In all likelihood they will continue to do so.

This chapter covers in a necessarily brief and, perhaps superficial, but broad way, these relationships between ABM and ALife and extrapolates to future possibilities. This chapter is organized as follows. Section “[Artificial Life](#)” introduces Artificial Life, its essential elements and its relationship to computing and agent-based modeling. Section “[ALife in Agent-Based Modeling](#)” describes several examples of ABM applications spanning many scales. Section “[Future Directions](#)” concludes with future directions for ABM and ALife. A bibliography is included for further reading.

Artificial Life

Artificial Life was initially motivated by the need to model biological systems and brought with it the need for computation. The field of ALife has always been multi-disciplinary and continues to encompass a broad research agenda covering a variety of topics from a number of disciplines, including:

- Essential elements of life and artificial life,
- Origins of life and self-organization,
- Evolutionary dynamics,
- Replication and development processes,
- Learning and evolution,
- Emergence,
- Computation of living systems,
- Simulation systems for studying ALife, and
- Many others.

Each of these topics has threads leading into agent-based modeling.

The Essence of ALife

The essence of Artificial Life is summed up by Langton (p. xxii in [43]) with a list of essential characteristics:

- Lifelike behavior on the part of man-made systems
- Semi-autonomous entities whose local interactions with one another are governed by a set of simple rules
- Populations, rather than individuals
- Simple rather than complex specifications
- Local rather than global control
- Bottom-up rather than top-down modeling
- Emergent rather than pre-specified behaviors.

Langton observes that complex high-level dynamics and structures often emerge (in living and artificial systems), developing over time out of the local interactions among low-level primitives. Agent-based modeling has grown up around the need to model the essentials of ALife.

Self-Replication and Cellular Automata Artificial Life traces its beginnings to the work of John von Neumann in the 1940s and investigations into the theoretical possibilities for developing a *self-replicating* machine [64]. Such a self-replicating machine not only carries instructions for its operations, but also for its replication. The issue concerned how to replicate such a machine that contained the instructions for its operation along with the instructions for its replication. Did a machine to replicate such a machine need to contain both the instructions for the machine's operation and replication, as well as instructions for replicating the instructions on how to replicate the original machine? (see Fig. 1). Von Neumann used the abstract mathematical construct of *cellular automata*, originally conceived in discussions with Stanislaw Ulam, to prove that such a machine could be designed, at least in theory. Von Neumann was never able to build such a machine due to the lack of sophisticated computers that existed at the time.

Cellular automata (CA) have been central to the development of computing Artificial Life models. Virtually all of the early agent-based models that required agents to be spatially located were in the form of von Neumann's original cellular automata. A cellular automata is a *finite-state machine* in which time and space are treated as discrete rather than continuous, as would be the case, for example in differential equation models. A typical CA is a two-dimensional grid or lattice consisting of cells. Each cell assumes one of a finite number of states at any time. A cell's neighborhood is the set of cells surrounding a cell, typically, a five-cell neighborhood (von Neumann neigh-

borhood) or a nine-cell neighborhood (Moore neighborhood), as in Fig. 2.

A set of simple state transition rules determines the value of each cell based on the cell's state and the states of neighboring cells. Every cell is updated at each time according to the transition rules. Each cell is identical in terms of its update rules. Cells differ only in their initial states. A CA is deterministic in the sense that the same state for a cell and its set of neighbors always results in the same updated state for the cell. Typically, CAs are set up with periodic boundary conditions, meaning that the set of cells on one edge of the grid boundary are the neighbor cells to the cells on the opposite edge of the grid boundary. The space of the CA grid forms a surface on a toroid, or donut-shape, so there is no boundary per se. It is straightforward to extend the notion of cellular automata to two, three, or more dimensions.

Von Neumann solved the self-replication problem by developing a cellular automaton in which each cell had 29 possible states and five neighbors (including the updated cell itself). In the von Neumann neighborhood, neighbor cells are in the north, south, east and west directions from the updated cell.

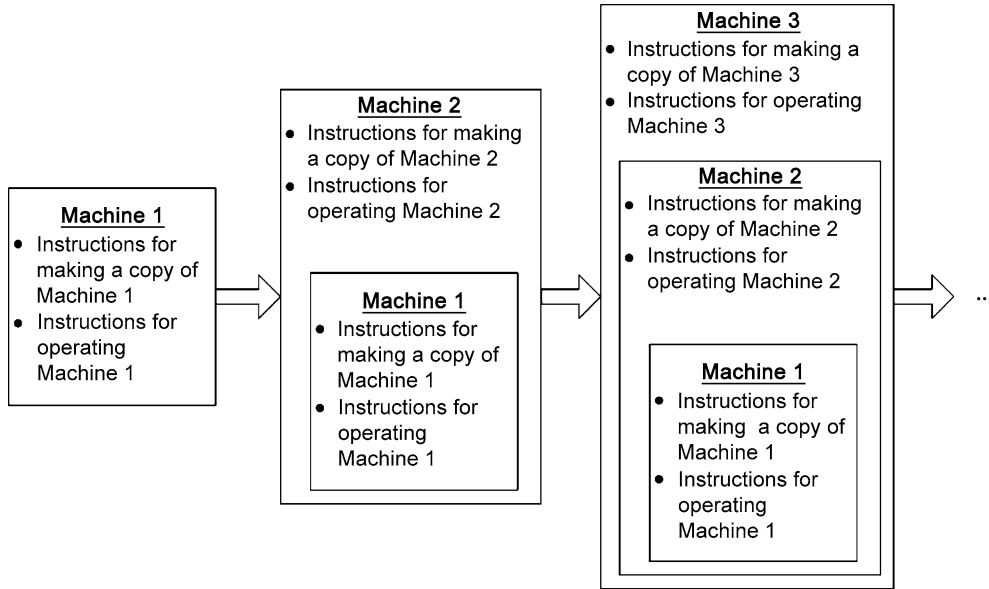
The Game Of Life Conway's Game of Life, or Life, developed in the 1970s, is an important example of a CA [8, 31, 57]. The simplest way to illustrate some of the basic ideas of agent-based modeling is through a CA. The Game of Life is a two-state, nine-neighbor cellular automata with three rules that determine the state (either On, i. e., shaded, or Off, i. e., white) of each cell:

1. A cell will be On in the next generation if exactly three of its eight neighboring cells are currently On.
2. A cell will retain its current state if exactly two of its neighbors are On.
3. A cell will be Off otherwise.

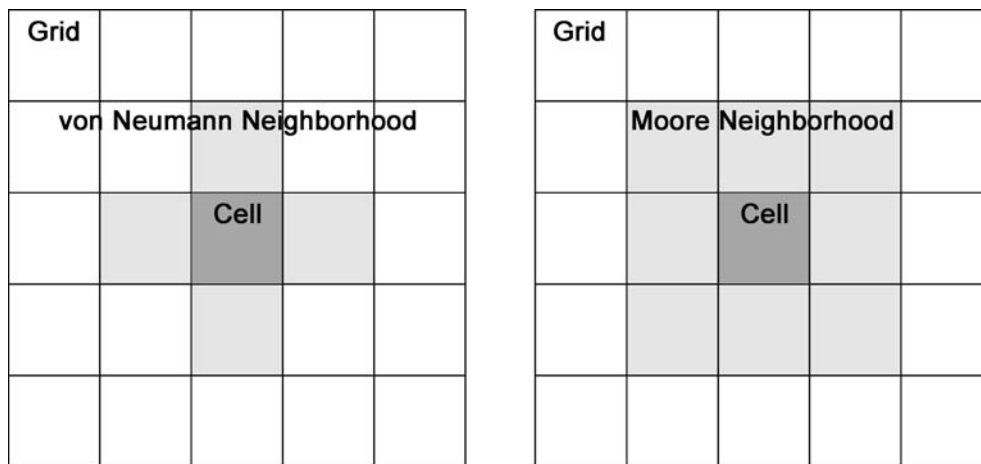
Initially, a small set of On cells is randomly distributed over the grid. The three rules are then applied repeatedly to all cells in the grid.

After several updates of all cells on the grid, distinctive patterns emerge, and in some cases these patterns can sustain themselves indefinitely throughout the simulation (Fig. 3). The state of each cell is based only on the current state of the cell and the cells touching it in its immediate neighborhood. The nine-neighbor per neighborhood assumption built into Life determines the scope of the locally available information for each cell to update its state.

Conway showed that, at least in theory, the structures and patterns that can result during a Life computation are complex enough to be the basis for a fully functional



Agent Based Modeling and Artificial Life, Figure 1
Von Neumann's Self-Replication Problem



Agent Based Modeling and Artificial Life, Figure 2
Cellular Automata Neighborhoods

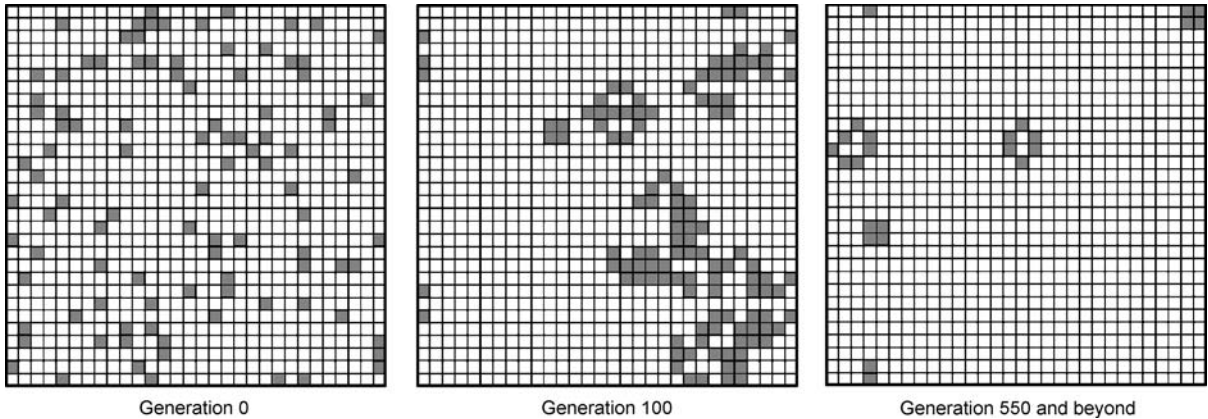
computer that is complex enough to spontaneously generate self-replicating structures (see the section below on universal computation). Two observations are important about the Life rules:

- As simple as the state transition rules are, by using only local information, structures of arbitrarily high complexity can emerge in a CA.
- The specific patterns that emerge are extremely sensitive to the specific rules used. For example, changing

Rule 1 above to “A cell will be On in the next generation if exactly four of its eight neighboring cells are currently On” results in the development of completely different patterns.

- The Game of Life provides insights into the role of information in fundamental life processes.

Cellular Automata Classes Wolfram investigated the possibilities for complexity in cellular automata across the



Agent Based Modeling and Artificial Life, Figure 3
Game of Life

full range of transition rules and initial states, using one-dimensional cellular automata [70]. He categorized four distinct classes for the resulting patterns produced by a CA as it is solved repeatedly over time. These are:

- Class I: homogeneous state,
- Class II: simple stable or periodic structure,
- Class III: chaotic (non-repeating) pattern, and
- Class IV: complex patterns of localized structures.

The most interesting of these is Class IV cellular automata, in which very complex patterns of non-repeating localized structures emerge that are often long-lived. Wolfram showed that these Class IV structures were also complex enough to support universal computation [72]. Langton [45] coined the term “life at the edge of chaos” to describe the idea that Class IV systems are situated in a thin region between Class II and Class III systems. Agent-based models often yield Class I, II, and III behaviors.

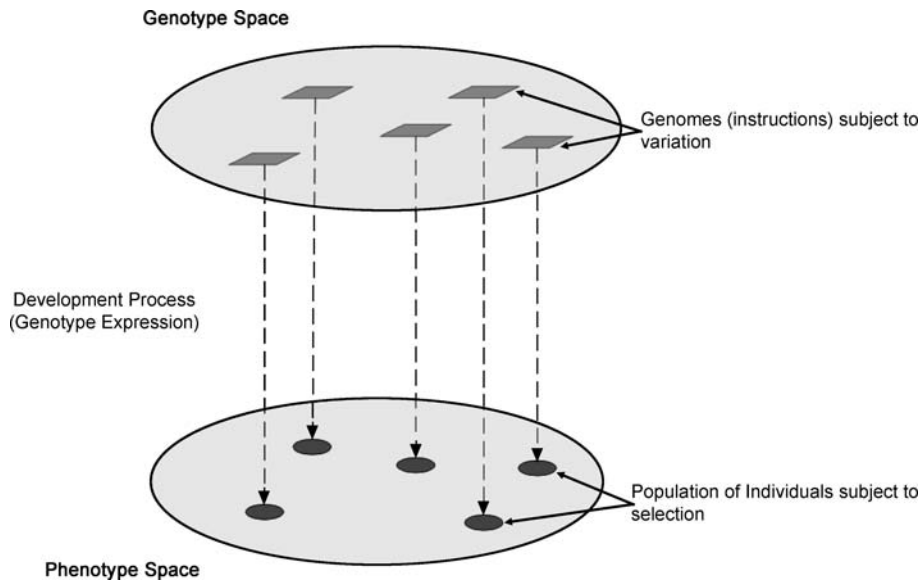
Other experiments with CAs investigated the simplest representations that could replicate themselves and produce emergent structures. *Langton’s Loop* is a self-replicating two-dimensional cellular automaton, much simpler than von Neumann’s [41]. Although not complex enough to be a universal computer, Langton’s Loop was the simplest known structure that could reproduce itself. *Langton’s Ant* is a two-dimensional CA with a simple set of rules, but complicated emergent behavior. Following a simple set of rules for moving from cell to cell, a simulated ant displays unexpectedly complex behavior. After an initial period of chaotic movements in the vicinity of its initial location, the ant begins to build a recurrent pattern of regular structures that repeats itself indefinitely [42].

Langton’s Ant has behaviors complex enough to be a universal computer.

Genotype/Phenotype Distinction Biologists distinguish between the *genotype* and the *phenotype* as hallmarks of biological systems. The genotype is the template – the set of instructions, the specification, the blueprint – for an organism. DNA is the genotype for living organisms, for example. A DNA strand contains the complete instructions for the replication and development of the organism. The phenotype is the organism – the machine, the product, the result – that develops from the instructions in the genotype (Fig. 4).

Morphogenesis is the developmental process by which the phenotype develops in accord with the genotype, through interactions with and resources obtained from its environment. In a famous paper, Turing [66] modeled the dynamics of morphogenesis, and more generally, the problem of how patterns self-organize spontaneously in nature. Turing used differential equations to model a simple set of reaction-diffusion chemical reactions. Turing demonstrated that only a few assumptions were necessary to bring about the emergence of wave patterns and gradients of chemical concentration, suggestive of morphological patterns that commonly occur in nature. *Reaction-diffusion systems* are characterized by the simultaneous processes of attraction and repulsion, and are the basis for the agent behavioral rules (attraction and repulsion) in many social agent-based models.

More recently, Bonabeau extended Turing’s treatment of morphogenesis to a theory of pattern formation based on agent-based modeling. Bonabeau [12] states the reason for relying on ABM: “because pattern-forming sys-



Agent Based Modeling and Artificial Life, Figure 4
Genotype and Phenotype Relations

tems based on agents are (relatively) more easily amenable to experimental observations.”

Information Processes One approach to building systems from a genotype specification is based on the methodology of *recursively generated objects*. Such recursive systems are compact in their specification, and their repeated application can result in complex structures, as demonstrated by cellular automata.

Recursive systems are logic systems in which strings of symbols are recursively rewritten based a minimum set of instructions. Recursive systems, or term replacement systems, as they have been called, can result in complex structures. Examples of recursive systems include cellular automata, as described above, and *Lindenmayer systems*, called *L-systems* [46]. An L-system consists of a formal grammar, which is a set of rules for rewriting strings of symbols. L-systems have been used extensively for modeling living systems, for example, plant growth and development, producing highly realistic renderings of plants, with intricate morphologies and branching structures.

Wolfram [71] used symbolic recursion as a basis for developing *Mathematica*, the computational mathematics system based on *symbolic processing* and term replacement. Unlike numeric programming languages, a symbolic programming language allows a variable to be a basic object and does not require a variable to be assigned a value before it is used in a program.

Any agent-based model is essentially a recursive system. Time is simulated by the repeated application of the

agent updating rules. The genotype is the set of rules for the agent behaviors. The phenotype is set of the patterns and structures that emerge from the computation. As in cellular automata and recursive systems, extremely complex structures emerge in agent-based models that are often unpredictable from examination of the agent rules.

Emergence One of the primary motivations for the field of ALife is to understand *emergent* processes, that is, the processes by which life emerges from its constituent elements. Langton writes: “The ‘key’ concept in ALife, is emergent behavior.” (p. 2 in [44]). Complex systems exhibit patterns of emergence that are not predictable from inspection of the individual elements. Emergence is described as unexpected, unpredictable or otherwise surprising. That is, the modeled system exhibits behaviors that are not explicitly built into the model. Unpredictability is due to the non-linear effects that result from the interactions of entities having simple behaviors. Emergence by these definitions is something of a subjective process.

In biological systems, emergence is a central issue whether it be the emergence of the phenotype from the genotype, the emergence of protein complexes from genomic information networks [39], or the emergence of consciousness from networks of millions of brain cells.

One of the motivations for agent-based modeling is to explore the emergent behaviors exhibited by the simulated system. In general, agent-based models often exhibit patterns and relationships that emerge from agents interactions. An example is the observed formation of groups of

agents that collectively act in coherent and coordinated patterns. Complex adaptive systems, widely investigated by Holland in his agent-based model Echo [37], are often structured in hierarchies of emergent structures. Emergent structures can collectively form higher-order structures, using the lower-level structures as building blocks. An emergent structure itself can take on new emergent behaviors. These structures in turn affect the agents from which the structure has emerged in a process called *downward causation* [32]. For example, in the real world people organize and identify with groups, institutions, nations, etc. They create norms, laws, and protocols that in turn act on the individuals comprising the group.

From the perspective of agent-based modeling, emergence has some interesting challenges for modeling:

- How does one operationally define emergence with respect to agent-based modeling?
- How does one automatically identify and measure the emergence of entities in a model?
- How do agents that comprise an emergent entity perceived by an observer recognize that they are part of that entity?

Artificial Chemistry *Artificial chemistry* is a subfield of ALife. One of the original goals of Artificial chemistry was to understand how life could originate from pre-biotic chemical processes. Artificial chemistry studies self-organization in chemical reaction networks by simulating chemical reactions between artificial molecules. Artificial chemistry specifies well-understood chemical reactions and other information such as reaction rates, relative molecular concentrations, probabilities of reaction, etc. These form a network of possibilities. The artificial substances and the networks of chemical reactions that emerge from the possibilities are studied through computation. Reactions are specified as recursive algebras and activated as term replacement systems [30].

Hypercycles The emergence of *autocatalytic sets*, or *hypercycles*, has been a prime focus of artificial chemistry [22]. A hypercycle is a self-contained system of molecules and a self-replicating, and thereby self-sustaining, cyclic linkage of chemical reactions. Hypercycles evolve through a process by which self-replicating entities compete for selection.

The hypercycle model illustrates how an ALife process can be adopted to the agent-based modeling domain. Inspired by the hypercycle model, Padgett [54] developed an agent-based model of the co-evolution of economic production and economic firms, focusing on skills. Padgett

used the model to establish three principles of social organization that provide foundations for the evolution of technological complexity:

- Structured topology (how interaction networks form)
- Altruistic learning (how cooperation and exchange emerges), and
- Stigmergy (how agent communication is facilitated by using the environment as a means of information exchange among agents).

Digital Organisms The widespread availability of personal computers spurred the development of ALife programs used to study evolutionary processes *in silico*. Tierra was the first system devised in which computer programs were successfully able to evolve and adapt [59]. Avida extended Tierra to account for the spatial distribution of organisms and other features [52,68]. Echo is a simulation framework for implementing models to investigate mechanisms that regulate diversity and information-processing in complex adaptive systems (CAS), systems comprised of many interacting adaptive agents [36,37]. In implementations of Echo, populations evolve interaction networks, resembling species communities in ecological systems, which regulate the flow of resources.

Systems such as Tierra, Avida, and Echo simulate populations of digital organisms, based on the genotype/phenotype schema. They employ computational algorithms to mutate and evolve populations of organisms living in a simulated computer environment. Organisms are represented as strings of symbols, or agent attributes, in computer memory. The environment provides them with resources (computation time) they need to survive, compete, and reproduce. Digital organisms interact in various ways and develop strategies to ensure survival in resource-limited environments.

Digital organisms are extended to agent-based modeling by implementing individual-based models of food webs in a system called DOVE [69]. Agent-based models allow a more complete representation of agent behaviors and their evolutionary adaptability at both the individual and population levels.

ALife and Computing

Creating life-like forms through computation is central to Artificial Life. Is it possible to create life through computation? The capabilities and limitations of computation constrain the types of artificial life that can be created. The history of ALife has close ties with important events in the history of computation.

Alan Turing [65] investigated the limitations of computation by developing an abstract and idealized computer, called a *Universal Turing Machine* (UTM). A UTM has an infinite tape (memory) and is therefore an idealization of any actual computer that may be realized. A UTM is capable of computing anything that is computable, that is, anything that can be derived via a logical, deductive series of statements. Are the algorithms used in today's computers, and in ALife calculations and agent-based models in particular, as powerful as universal computers?

Any system that can effectively simulate a small set of logical operations (such as AND and NOT) can effectively produce any possible computation. Simple rule systems in cellular automata were shown to be equivalent to universal computers [67,72], and in principle able to compute anything that is computable – *perhaps, even life!*

Some have argued that life, in particular human consciousness, is not the result of a logical-deductive or algorithmic process and therefore not computable by a Universal Turing Machine. This problem is more generally referred to as the *mind-body problem* [48]. Dreyfus [20] argues against the assumption often made in the field of artificial intelligence that human minds function like general purpose symbol manipulation machines. Penrose [56] argues that the rational processes of the human mind transcend formal logic systems. In a somewhat different view, biological naturalism contends [63] that human behavior might be able to be simulated, but human consciousness is outside the bounds of computation.

Such philosophical debates are as relevant to agent-based modeling as they are to artificial intelligence, for they are the basis of answering the question of what kind of systems and processes agent-based models will ultimately be able, or unable, to simulate.

Artificial Life Algorithms

ALife use several *biologically-inspired computational algorithms* [53]. Bioinspired algorithms include those based on Darwinian evolution, such as evolutionary algorithms, those based on neural structures, such as neural networks, and those based on decentralized decision making behaviors observed in nature. These algorithms are commonly used to model adaptation and learning in agent-based modeling or to optimize the behaviors of whole systems.

Evolutionary Computing *Evolutionary computing* includes a family of related algorithms and programming solution techniques inspired by evolutionary processes, especially the genetic processes of DNA replication and cell

division [21]. These techniques are known as *evolutionary algorithms* and include the following [7]:

- Genetic algorithms [34,35,36,38,51]
- Evolution strategies [60]
- Learning classifier systems [38]
- Genetic programming [40]
- Evolutionary programming [28]

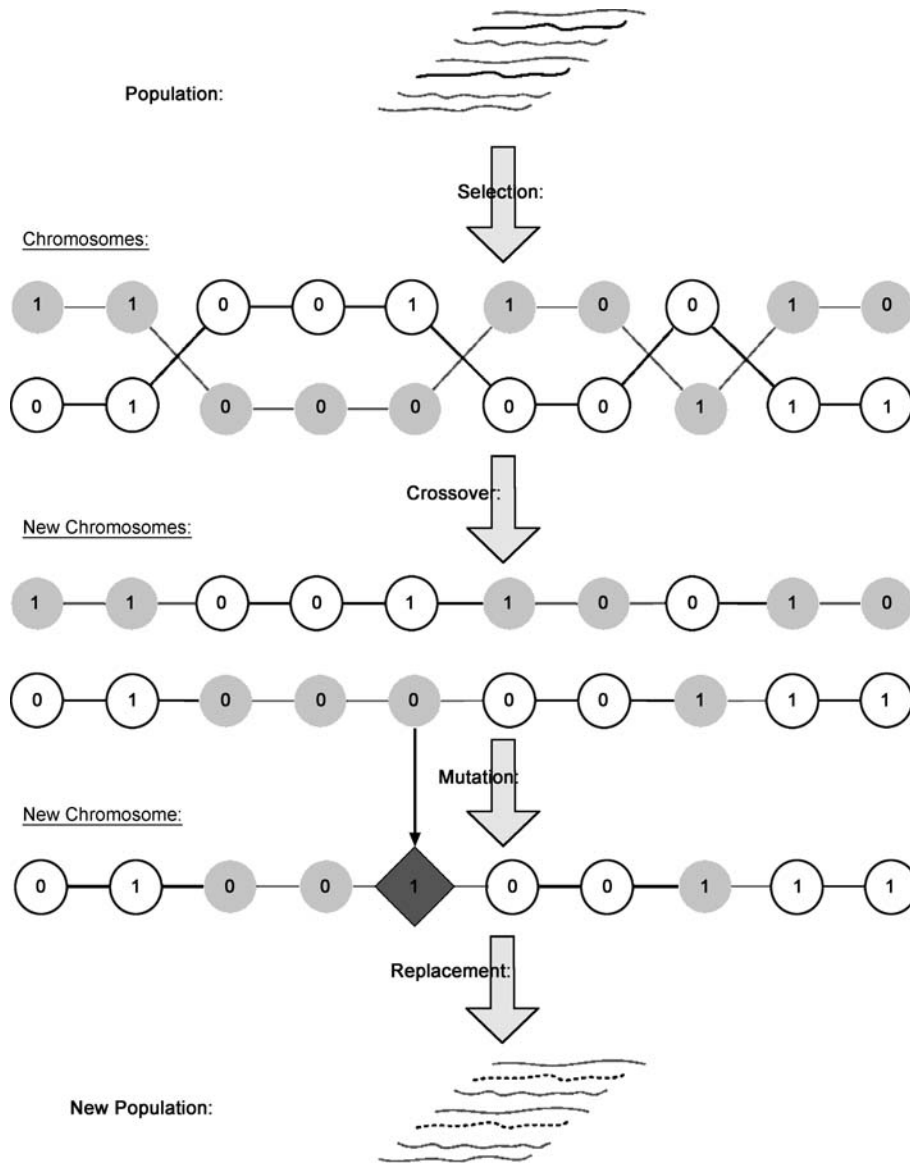
Genetic algorithms (GA) model the dynamic processes by which populations of individuals evolve to improved levels of fitness for their particular environment over repeated generations. GAs illustrate how evolutionary algorithms process a population and apply the genetic operations of mutation and crossover (see Fig. 5). Each behavior is represented as a chromosome consisting of a series of symbols, for example, as a series of 0s and 1s. The encoding process establishing correspondence between behaviors and their chromosomal representations is part of the modeling process.

The general steps in a genetic algorithm are as follows:

1. *Initialization*: Generate an initial population of individuals. The individuals are unique and include specific encoding of attributes in chromosomes that represents the characteristics of the individuals.
2. *Evaluation*: Calculate the fitness of all individuals according to a specified fitness function.
3. *Checking*: If any of the individuals has achieved an acceptable level of fitness, stop, the problem is solved. Otherwise, continue with selection.
4. *Selection*: Select the best pair of individuals in the population for reproduction according to their high fitness levels.
5. *Crossover*: Combine the chromosomes for the two best individuals through a crossover operation and produce a pair of offspring.
6. *Mutation*: Randomly mutate the chromosomes for the offspring.
7. *Replacement*: Replace the least fit individuals in the population with the offspring.
8. Continue at Step 2

Steps 5 and 6 above, the operations of crossover and mutation, comprise the set of genetic operators inspired by nature. This series of steps for a GA comprise a basic framework rather than a specific implementation. Actual GA implementations include numerous variations and alternative implementations in several of the GA steps.

Evolution strategies (ES) are similar to genetic algorithms but rely on mutation as its primary genetic operator.



Agent Based Modeling and Artificial Life, Figure 5
Genetic Algorithm

Learning classifier systems (LCS) build on genetic algorithms and adaptively assign relative weights to sensor-action sets that result in the most positive outcomes relative to a goal.

Genetic programming (GP) has similar features to genetic algorithms, but instead of using 0s and 1s or other symbols for comprising chromosomes, GPs combine logical operations and directives in a tree structure. In effect, chromosomes in GPs represent whole computer programs that perform a variety of functions with varying degrees of success and efficiencies. GP chromosomes are evaluated

against fitness or performance measures and recombined. Better performing chromosomes are maintained and expand their representation in the population. For example, an application of a GP is to evolve a better-performing rule set that represents an agent's behavior.

Evolutionary programming (EP) is a similar technique to genetic programming, but relies on mutation as its primary genetic operator.

Biologically Inspired Computing Artificial neural networks (ANN) are another type of commonly used bio-

logically inspired algorithm [50]. An artificial neural network uses mathematical models based on the structures observed in neural systems. An artificial neuron contains a stimulus-response model of neuron activation based on thresholds of stimulation. In modeling terms, neural networks are equivalent to nonlinear, statistical data modeling techniques. Artificial neural networks can be used to model complex relationships between inputs and outputs and to find patterns in data that are dynamically changing. An ANN is adaptive in that changes in its structure are based on external or internal information that flows through the network. The adaptive capability makes ANN an important technique in agent-based models.

Swarm intelligence refers to problem solving techniques, usually applied to solving optimization problems that are based on decentralized problem solving strategies that have been observed in nature. These include:

- Ant colony optimization [19].
- Particle swarm optimization [16].

Swarm intelligence algorithms simulate the movement and interactions of large numbers of ants or particles over a search space. In terms of agent-based modeling, the ants or particles are the agents, and the search space is the environment. Agents have position and state as attributes. In the case of particle swarm optimization, agents also have velocity.

Ant colony optimization (ACO) mimics techniques that ants use to forage and find food efficiently [13,24]. The general idea of ant colony optimization algorithms is as follows:

1. In a typical ant colony, ants search randomly until one of them finds food.
2. Then they return to their colony and lay down a chemical pheromone trail along the way.
3. When other ants find such a pheromone trail, they are more likely to follow the trail rather than to continue to search randomly.
4. As other ants find the same food source, they return to the nest, reinforcing the original pheromone trail as they return.
5. As more and more ants find the food source, the ants eventually lay down a strong pheromone trail to the point that virtually all the ants are directed to the food source.
6. As the food source is depleted, fewer ants are able to find the food and fewer ants lay down a reinforcing pheromone trail; the pheromone naturally evaporates and eventually no ants proceed to the food source, as

the ants shift their attention to searching for new food sources.

In an ant colony optimization computational model, the optimization problem is represented as a graph, with nodes representing places and links representing possible paths. An ant colony algorithm mimics ant behavior with simulated ants moving from node to node in the graph, laying down pheromone trails, etc. The process by which ants communicate indirectly by using the environment as an intermediary is known as *stigmergy* [13], and is commonly used in agent-based modeling.

Particle swarm optimization (PSO) is another decentralized problem solving technique in which a swarm of particles is simulated as it moves over a search space in search of a global optimum. A particle stores its best position found so far in its memory and is aware of the best positions obtained by its neighboring particles. The velocity of each particle adapts over time based on the locations of the best global and local solutions obtained so far, incorporating a degree of stochastic variation in the updating of the particle positions at each iteration.

Artificial Life Algorithms and Agent-Based Modeling Biologically-inspired algorithms are often used with agent-based models. For example, an agent's behavior and its capacity to learn from experience or to adapt to changing conditions can be modeled abstractly through the use of genetic algorithms or neural networks. In the case of a GA, a chromosome effectively represents a single agent action (output) given a specific condition or environmental stimulus (input). Behaviors that are acted on and enable the agent to respond better to environmental challenges are reinforced and acquire a greater share of the chromosome pool. Behaviors that fail to improve the organism's fitness diminish in their representation in the population.

Evolutionary programming can be used to directly evolve programs that represent agent behaviors. For example, Manson [49] develops a bounded rationality model using evolutionary programming to solve an agent multi-criteria optimization problem.

Artificial neural networks have also been applied to modeling adaptive agent behaviors, in which an agent derives a statistical relationship between the environmental conditions it faces, its history, and its actions, based on feedback on the success or failures of its actions and the actions of others. For example, an agent may need to develop a strategy for bidding in a market, based on the success of its own and other's previous bids and outcomes.

Finally, swarm intelligence approaches are agent-based in their basic structure, as described above. They can

also be used for system optimization through the selection of appropriate parameters for agent behaviors.

ALife Summary

Based on the previous discussion, the essential features of an ALife program can be summarized as follows:

- *Population*: A population of organisms or individuals is considered. The population may be diversified, and individuals may vary their characteristics, behaviors, and accumulated resources, in both time and space.
- *Interaction*: Interaction requires sensing of the immediate locale, or neighborhood, on the part of an individual. An organism can simply become “aware” of other organisms in its vicinity or it may have a richer set of interactions with them. The individual also interacts with its (non-agent) environment in its immediate locale. This requirement introduces spatial aspects into the problem, as organisms must negotiate the search for resources through time and space.
- *Sustainment and renewal*: Sustainment and renewal requires the acquisition of resources. An organism needs to sense, find, ingest, and metabolize resources, or nourishment as an energy source for processing into other forms of nutrients. Resources may be provided by the environment, i. e., outside of the agents themselves, or by other agents. The need for sustainment leads to competition for resources among organisms. Competition could also be a precursor to cooperation and more complex emergent social structures if this proves to be a more effective strategy for survival.
- *Self-reproduction and replacement*: Organisms reproduce by following instructions at least partially embedded within themselves and interacting with the environment and other agents. Passing on traits to the next generation implies a requirement for trait transmission. Trait transmission requires encoding an organism’s traits in a reduced form, that is, a form that contains less than the total information representing the entire organism. It also requires a process for transforming the organism’s traits into a viable set of possible new traits for a new organism. Mutation and crossover operators enter into such a process. Organisms also leave the population and are replaced by other organisms, possibly with different traits. The organisms can be transformed through changes in their attributes and behaviors, as in, for example, learning or aging. The populations of organisms can be transformed through the introduction of new organisms and replacement, as in evolutionary adaptation.

As we will see in the section that follows, many of the essential aspects of ALife have been incorporated into the development of agent-based models.

ALife in Agent-Based Modeling

This section briefly touches on the ways in which ALife has motivated agent-based modeling. The form of agent-based models, in terms of their structure and appearance, is directly based on early models from the field of ALife. Several application disciplines in agent-based modeling have been spawned and infused by ALife concepts. Two are covered here. These are how agent-based modeling is applied to social and biological systems.

Agent-Based Modeling Topologies

Agent-based modeling owes much to artificial life in both form and substance. Modeling a population of heterogeneous agents with a diverse set of characteristics is a hallmark of agent-based modeling. The agent perspective is unique among simulation approaches, unlike the process perspective or the state-variable approach taken by other simulation approaches.

As we have seen, agents interact with a small set of neighbor agents in a local area. Agent neighborhoods are defined by how agents are connected, the agent interaction topology. Cellular automata represent agent neighborhoods by using a grid in which the agents exist in the cells, one agent per cell, or as the nodes of the lattice of the grid. The cells immediately surrounding an agent comprise the agent’s neighborhood and the agents that reside in the neighborhood cells comprise the neighbors. Many agent-based models have been based on this cellular automata spatial representation. The transition from a cellular automata, such as the game of Life, to an agent-based model is accomplished by allowing agents to be distinct from the cells on which they reside and allowing the agents to move from cell to cell across the grid. Agents move according to the dictates of their behaviors, interacting with other agents that happen to be in their local neighborhoods along the way.

Agent interaction topologies have been extended beyond cellular automata to include networks, either predefined and static as in the case of autocatalytic chemical networks, or endogenous and dynamic, according to the results of agent interactions that occur in the model. Networks allow an agent’s neighborhood to be defined more generally and flexibly, and in the case of social agents, more accurately describe social agents’ interaction patterns. In addition to cellular automata grids and networks, agent interaction topologies have also been ex-

tended across a variety of domains. In summary, agent interaction topologies include:

- Cellular automata grids (agents are cells or are within in cells) or lattices (agents are grid points),
- Networks, in which agents of vertices and agent relationships are edges,
- Continuous space, in one, two or three dimensions;
- Aspatial random interactions, in which pairs of agents are randomly selected; and
- Geographical Information Systems (GIS), in which agents move over geographically-defined patches, relaxing the one-agent per cell restriction.

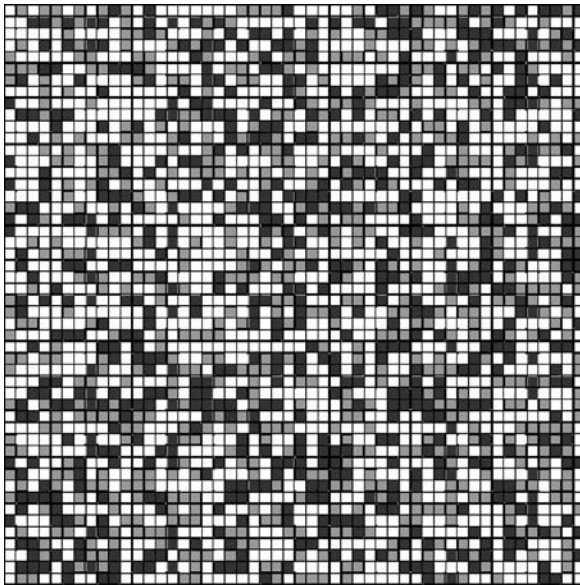
Social Agent-Based Modeling

Early social agent-based models were based on ALife's cellular automata approach. In applications of agent-based modeling to social processes, agents represent people or groups of people, and agent relationships represent processes of social interaction [33].

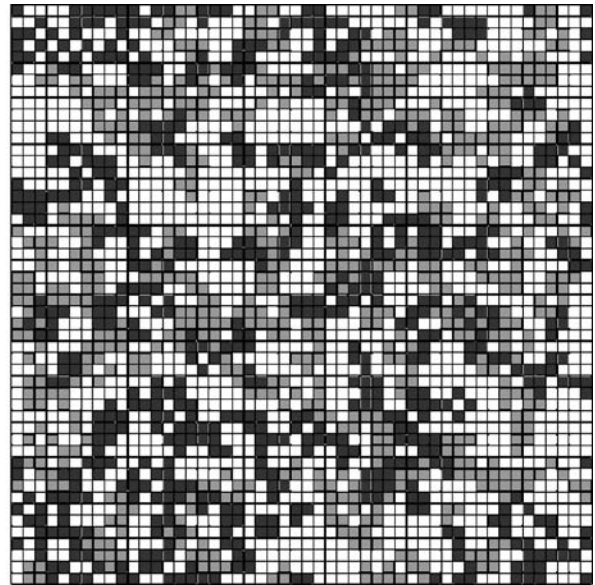
Social Agents Sakoda [61] formulated one of the first social agent-based models, the Checkerboard Model, which had some of the key features of a cellular automaton. Following a similar approach, Schelling developed a model

of housing segregation in which agents represent homeowners and neighbors, and agent interactions represent agents' perceptions of their neighbors [62]. Schelling studied housing segregation patterns and posed the question of whether it is possible to get highly segregated settlement patterns even if most individuals are, in fact, "color-blind?" The Schelling model demonstrated that segregated housing areas can develop spontaneously in the sense that system-level patterns can emerge that are not necessarily implied by or consistent with the objectives of the individual agents (Fig. 6). In the model, agents operated according to a fixed set of rules and were not adaptive.

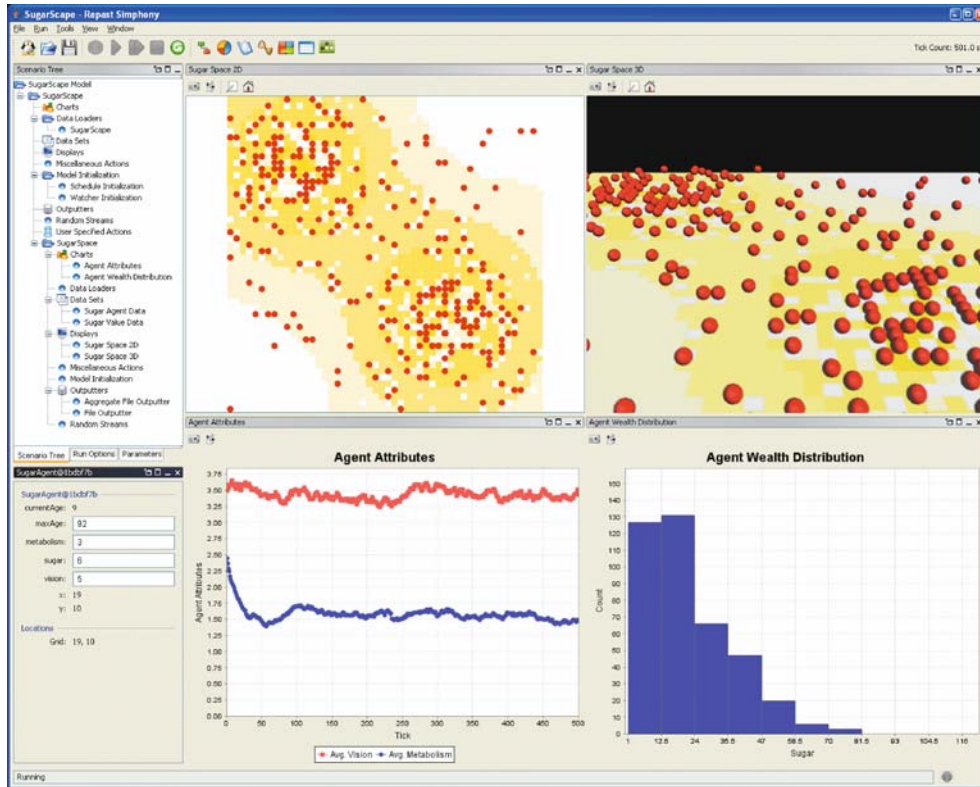
Identifying the social interaction mechanisms for how cooperative behavior emerges among individuals and groups has been addressed using agent-based modeling and evolutionary game theory. Evolutionary game theory accounts for how the repeated interactions of players in a game-theoretic framework affect the development and evolution of the players' strategies. Axelrod showed, using a cellular automata approach, in which agents on the grid employed a variety of different strategies, that a simple Tit-For-Tat strategy of reciprocal behavior toward individuals is enough to establish sustainable cooperative behavior [4,5]. In addition, Axelrod investigated strategies that were self-sustaining and robust in that they reduced the possibility of invasion by agents having other strategies.



Initial Random Distribution of Two Agents Types



Distribution of Agents After Repeated Interactions, Showing Spontaneous Housing Segregation Patterns



Agent Based Modeling and Artificial Life, Figure 7
Sugarscape Artificial Society Simulation in the Repast Agent-Based Modeling Toolkit

Epstein and Axtell introduced the notion of an external environment that agents interact with in addition to other agents. In their groundbreaking Sugarscape model of artificial societies, agents interacted with their environment depending on their location in the grid [26]. This allowed agents to access environmental variables, extract resources, etc., based on location. In numerous computational experiments, Sugarscape agents emerged with a variety of characteristics and behaviors, highly suggestive of a realistic, although rudimentary and abstract, society (Fig. 7). They observed emergent processes that they interpreted as death, disease, trade, wealth, sex and reproduction, culture, conflict and war, as well as externalities such as pollution. As agents interacted with their neighbors as they moved around the grid, the interactions resulted in a contact network, that is, a network consisting of nodes and links. The nodes are agents and the links indicate the agents that have been neighbors at some point in the course of their movements over the grid. Contact networks were the basis for studying contagion and epidemics in the Sugarscape model. Understanding the agent rules that govern how networks are structured and grow, how

quickly information is communicated through networks, and the kinds of relationships that networks embody are important aspects of modeling agents.

Culture and Generative Social Science Dawkins, who has written extensively on aspects of Darwinian evolution, coined the term *meme* as the smallest element of culture that is transmissible between individuals, similar to the notion of the gene as being the primary unit of transmitting genetic information [17]. Several social agent-based models are based on a meme representation of culture as shared or collective agent attributes.

In the broadest terms, social agent-based simulation is concerned with social interaction and social processes. Emergence enters into social simulation through *generative social science* whose goal is to model social processes as emergent processes and their emergence as the result of social interactions. Epstein has argued that social processes are not fully understood unless one is able to theorize how they work at a deep level and have social processes emerge as part of a computational model [25]. More recent work has treated culture as a fluid and dynamic process subject

to interpretation of individual agents, more complex than the genotype/phenotype framework would suggest.

ALife and Biology

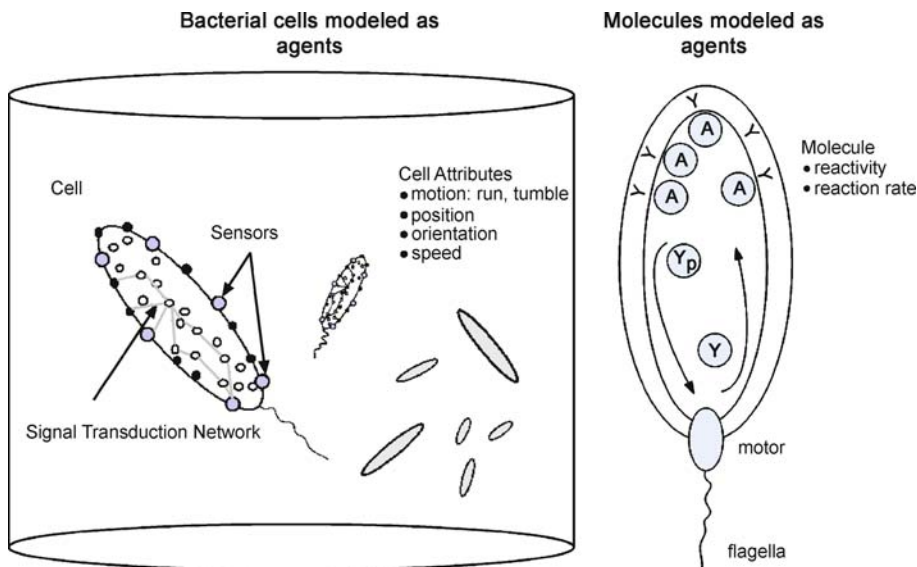
ALife research has motivated many agent-based computational models of biological systems, and at all scales, ranging from the cellular level, or even the subcellular molecular level, as the basic unit of agency, to complex organisms embedded in larger structures such as food webs or complex ecosystems.

From Cellular Automata to Cells Cellular automata are a natural application to modeling cellular systems [2,27]. One approach uses the cellular automata grid and cells to model structures of stationary cells comprising a tissue matrix. Each cell is a tissue agent. Mobile cells consisting of pathogens and antibodies are also modeled as agents. Mobile agents diffuse through tissue and interact with tissue and other co-located mobile cells. This approach is the basis for agent-based models of the immune system. Celada and Seiden [15] used bit-strings to model the cell receptors in a cellular automaton model of the immune system called IMMSIM. This approach was extended to a more general agent-based model in and implemented to maximize the number of cells that could be modeled in the CIMMSIM and ParImm systems [9]. The Basic Immune Simulator uses a general agent-based framework

(the Repast agent-based modeling toolkit) to model the interactions between the cells of the innate and adaptive immune system [29]. These approaches for modeling the immune system have inspired several agent-based models of intrusion detection for computer networks (see for example, [6]), and have found use in modeling the development and spread of cancer [58].

At the more macroscopic level, agent-based epidemic models have been developed using network topologies. These models include people and some representation of pathogens as individual agents for natural [11] and potentially man-made [14] epidemics.

Modeling bacteria and their associated behaviors in their natural environments is another direction of agent-based modeling. Expanding beyond the basic cellular automata structure into continuous space and network topologies, Emonet, et al. [23] developed AgentCell, a multi-scale agent-based model of *E. Coli* bacteria motility (Fig. 8). In this multi-scale agent-based simulation, molecules within a cell are modeled as individual agents. The molecular reactions comprising the signal transduction network for chemotaxis are modeled using an embedded stochastic simulator, StochSim [46]. This multi-scale approach allows the motile (macroscopic) behavior of colonies of bacteria to be modeled as a direct result of the modeled micro-level processes of protein production within the cells, which are based on individual molecular interactions.



Agent Based Modeling and Artificial Life, Figure 8
AgentCell Multi-Scale Agent-Based Model of Bacterial Chemotaxis

Artificial Ecologies Early models of ecosystems used approaches adapted from physical modeling, especially models of idealized gases based on statistical mechanics. More recently, individual-based models have been developed to represent the full range of individual diversity by explicitly modeling individual attributes or behaviors, and aggregating across individuals for an entire population [18]. Agent-based approaches model a diverse set of agents and their interactions based on their relationships, incorporating adaptive behaviors as appropriate. For example, food webs represent the complex, hierarchical network of agent relationships in local ecosystems [55]. Agents are individuals or species representatives. Adaptation and learning for agents in such food webs can be modeled to explore diversity, relative population sizes, and resiliency to environmental insult.

Adaptation and Learning in Agent-Based Models Biologists consider adaptation to be an essential part of the process of evolutionary change. Adaptation occurs at two levels: the individual level and the population level. In parallel with these notions, agents in an ABM adapt by changing their individual behaviors or by changing their proportional representation in the population. Agents adapt their behaviors at the individual level through learning from experience in their modeled environment.

With respect to agent-based modeling, theories of learning by individual agents or collectives of agents, as well as algorithms for how to model learning, become important. Machine learning is a field consisting of algorithms for recognizing patterns in data (such as data mining) through techniques such as supervised learning, unsupervised learning and reinforcement learning [3,10]. Genetic algorithms [34] and related techniques such as learning classifier systems [38] are commonly used to represent agent learning in agent-based models. In ABM applications, agents learn through interactions with the simulated environment in which they are embedded as the simulation precedes through time, and agents modify their behaviors accordingly.

Agents may also adapt collectively at the population level. Those agents having behavioral rules better suited to their environments survive and thrive, and those agents not so well suited are gradually eliminated from the population.

Future Directions

Agent-based modeling continues to be inspired by ALife – in the fundamental questions it is trying to answer, in the algorithms that it employs to model agent behaviors and

solve agent-based models, and in the computational architectures that are employed to implement agent-based models. The future of the fields of both ALife and ABM will continue to be intertwined in essential ways in the coming years.

Computational advances will continue at an ever-increasing pace, opening new vistas for computational possibilities in terms of expanding the scale of models that are possible. Computational advances will take several forms, including advances in computer hardware including new chip designs, multi-core processors, and advanced integrated hardware architectures. Software that take advantage of these designs and in particular computational algorithms and modeling techniques and approaches will continue to provide opportunities for advancing the scale of applications and allow more features to be included in agent-based models as well as ALife applications. These will be opportunities for advancing applications of ABM to ALife in both the realms of scientific research and in policy analysis.

Real-world optimization problems routinely solved by business and industry will continue to be solved by ALife-inspired algorithms. The use of ALife-inspired agent-based algorithms for solving optimization problems will become more widespread because of their natural implementation and ability to handle ill-defined problems.

Emergence is a key theme of ALife. ABM offers the capability to model the emergence of order in a variety of complex and complex adaptive systems. Inspired by ALife, identifying the fundamental mechanisms responsible for higher order emergence and exploring these with agent-based modeling will be an important and promising research area.

Advancing social sciences beyond the genotype/phenotype framework to address the generative nature of social systems in their full complexity is a requirement for advancing computational social models. Recent work has treated culture as a fluid and dynamic process subject to interpretation of individual agents, more complex in many ways than that provided by the genotype/phenotype framework.

Agent-based modeling will continue to be the avenue for exploring new constructs in ALife. If true artificial life is ever developed *in silico*, it will most likely be done using the methods and tools of agent-based modeling.

Bibliography

Primary Literature

1. Adami C (1998) Introduction to Artificial Life. TELOS, Santa Clara

2. Alber MS, Kiskowski MA, Glazier JA, Jiang Y (2003) On Cellular Automaton Approaches to Modeling Biological Cells. In: Rosenthal J, Gilliam DS (eds) *Mathematical Systems Theory in Biology, Communication, and Finance*, IMA Volume. Springer, New York, pp 1–39
3. Alpaydm E (2004) *Introduction to Machine Learning*. MIT Press, Cambridge
4. Axelrod R (1984) *The Evolution of Cooperation*. Basic Books, New York
5. Axelrod R (1997) *The Complexity of Cooperation: Agent-Based Models of Competition and Collaboration*. Princeton University Press, Princeton
6. Azzedine B, Renato BM, Kathia RLJ, Joao Bosco MS, Mirela SMAN (2007) An Agent Based and Biological Inspired Real-Time Intrusion Detection and Security Model for Computer Network Operations. *Comp Commun* 30(13):2649–2660
7. Back T (1996) *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford University Press, New York
8. Berlekamp ER, Conway JH, Guy RK (2003) *Winning Ways for Your Mathematical Plays*, 2nd edn. AK Peters, Natick
9. Bernaschi M, Castiglione F (2001) Design and Implementation of an Immune System Simulator. *Computers in Biology and Medicine* 31(5):303–331
10. Bishop CM (2007) *Pattern Recognition and Machine Learning*. Springer, New York
11. Bobashev GV, Goedecke DM, Yu F, Epstein JM (2007) A Hybrid Epidemic Model: Combining the Advantages of Agent-Based and Equation-Based Approaches. In: Henderson SG, Biller B, Hsieh M-H, Shortle J, Tew JD, Barton RR (eds) *Proc. 2007 Winter Simulation Conference*, Washington, pp 1532–1537
12. Bonabeau E (1997) From Classical Models of Morphogenesis to Agent-Based Models of Pattern Formation. *Artif Life* 3:191–211
13. Bonabeau E, Dorigo M, Theraulaz G (1999) *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, New York
14. Carley KM, Fridsma DB, Casman E, Yahja A, Altman N, Chen LC, Kaminsky B, Nave D (2006) Biowar: Scalable Agent-Based Model of Bioattacks. *IEEE Trans Syst Man Cybern Part A: Syst Hum* 36(2):252–265
15. Celada F, Seiden PE (1992) A Computer Model of Cellular Interactions in the Immune System. *Immunol Today* 13(2):56–62
16. Clerc M (2006) *Particle Swarm Optimization*. ISTE Publishing Company, London
17. Dawkins R (1989) *The Selfish Gene*, 2nd edn. Oxford University Press, Oxford
18. DeAngelis DL, Gross LJ (eds) (1992) *Individual-Based Models and Approaches in Ecology: Populations, Communities and Ecosystems*. *Proceedings of a Symposium/Workshop*, Knoxville, 16–19 May 1990. Chapman & Hall, New York. ISBN 0-412-03171-X
19. Dorigo M, Stützle T (2004) *Ant Colony Optimization*. MIT Press, Cambridge
20. Dreyfus HL (1979) *What Computers Can't Do: The Limits of Artificial Intelligence*. Harper & Row, New York
21. Eiben AE, Smith JE (2007) *Introduction to Evolutionary Computing*, 2nd edn. Springer, New York
22. Eigen M, Schuster P (1979) *The Hypercycle: A Principle of Natural Self-Organization*. Springer, Berlin
23. Emonet T, Macal CM, North MJ, Wickersham CE, Cluzel P (2005) *AgentCell: A Digital Single-Cell Assay for Bacterial Chemotaxis*. *Bioinformatics* 21(11):2714–2721
24. Engelbrecht AP (2006) *Fundamentals of Computational Swarm Intelligence*. Wiley, Hoboken
25. Epstein JM (2007) *Generative Social Science: Studies in Agent-Based Computational Modeling*. Princeton University Press, Princeton
26. Epstein JM, Axtell R (1996) *Growing Artificial Societies: Social Science from the Bottom Up*. MIT Press, Cambridge
27. Ermentrout GB, Edelstein-Keshet L (1993) Cellular Automata Approaches to Biological Modeling. *J Theor Biol* 160(1):97–133
28. Fogel LJ, Owens AJ, Walsh MJ (1966) *Artificial Intelligence through Simulated Evolution*. Wiley, Hoboken
29. Folcik VA, An GC, Orosz CG (2007) The Basic Immune Simulator: An Agent-Based Model to Study the Interactions between Innate and Adaptive Immunity. *Theor Biol Med Model* 4(39):1–18. <http://www.tbiomed.com/content/4/1/39>
30. Fontana W (1992) Algorithmic Chemistry. In: Langton CG, Taylor C, Farmer JD, Rasmussen S (eds) *Artificial Life II: Proceedings of the Workshop on Artificial Life*, Santa Fe, February 1990, Santa Fe Institute Studies in the Sciences of the Complexity, vol X. Addison-Wesley, Reading, pp 159–209
31. Gardner M (1970) The Fantastic Combinations of John Conway's New Solitaire Game Life. *Sci Am* 223:120–123
32. Gilbert N (2002) Varieties of Emergence. In: Macal C, Sallach D (eds) *Proceedings of the Agent 2002 Conference on Social Agents: Ecology, Exchange and Evolution*, Chicago, 11–12 Oct 2002, pp 1–11. Available on CD and at www.agent2002.anl.gov
33. Gilbert N, Troitzsch KG (1999) *Simulation for the Social Scientist*. Open University Press, Buckingham
34. Goldberg DE (1989) *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading
35. Goldberg DE (1994) Genetic and Evolutionary Algorithms Come of Age. *Commun ACM* 37(3):113–119
36. Holland JH (1975) *Adaptation in Natural and Artificial Systems*. University of Michigan, Ann Arbor
37. Holland J (1995) *Hidden Order: How Adaptation Builds Complexity*. Addison-Wesley, Reading
38. Holland JH, Booker LB, Colombetti M, Dorigo M, Goldberg DE, Forrest S, Riolo RL, Smith RE, Lanzi PL, Stolzmann W, Wilson SW (2000) What Is a Learning Classifier System? In: Lanzi PL, Stolzmann W, Wilson SW (eds) *Learning Classifier Systems, from Foundations to Applications*. Springer, London, pp 3–32
39. Kauffman SA (1993) *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, Oxford
40. Koza JR (1992) *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge. 840 pp
41. Langton CG (1984) Self-Reproduction in Cellular Automata. *Physica D* 10:135–144
42. Langton CG (1986) Studying Artificial Life with Cellular Automata. *Physica D* 22:120–149
43. Langton CG (1989) Preface. In: Langton CG (ed) *Artificial Life: Proceedings of an Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems*. Los Alamos, Sept 1987, Addison-Wesley, Reading, pp xv–xxvi
44. Langton CG (1989) *Artificial Life*. In: Langton CG (ed) *Artificial Life: The Proceedings of an Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems*. Los Alamos, Sept 1987, Santa Fe Institute Studies in the Sciences of Complexity, vol VI. Addison-Wesley, Reading, pp 1–47

45. Langton CG (1992) Life at the Edge of Chaos. In: Langton CG, Taylor C, Farmer JD, Rasmussen S (eds) *Artificial Life II: Proceedings of the Workshop on Artificial Life*. Santa Fe, Feb 1990, Santa Fe Institute Studies in the Sciences of the Complexity, vol X. Addison-Wesley, Reading, pp 41–91
46. Le Novère N, Shimizu TS (2001) Stochsim:Modelling of Stochastic Biomolecular Processes. *Bioinformatics* 17(6):575–576
47. Lindenmeyer A (1968) Mathematical Models for Cellular Interaction in Development. *J Theor Biol* 18:280–315
48. Lucas JR (1961) Minds, Machines and Godel. *Philosophy* 36(137):112–127
49. Manson SM (2006) Bounded Rationality in Agent-Based Models: Experiments with Evolutionary Programs. *Intern J Geogr Inf Sci* 20(9):991–1012
50. Mehrotra K, Mohan CK, Ranka S (1996) *Elements of Artificial Neural Networks*. MIT Press, Cambridge
51. Mitchell M, Forrest S (1994) Genetic Algorithms and Artificial Life. *Artif Life* 1(3):267–289
52. Ofria C, Wilke CO (2004) Avida: A Software Platform for Research in Computational Evolutionary Biology. *Artif Life* 10(2):191–229
53. Olariu S, Zomaya AY (eds) (2006) *Handbook of Bioinspired Algorithms and Applications*. Chapman, Boca Raton, pp 679
54. Padgett JF, Lee D, Collier N (2003) Economic Production as Chemistry. *Ind Corp Chang* 12(4):843–877
55. Peacor SD, Riolo RL, Pascual M (2006) Plasticity and Species Coexistence: Modeling Food Webs as Complex Adaptive Systems. In: Pascual M, Dunne JA (eds) *Ecological Networks: Linking Structure to Dynamics in Food Webs*. Oxford University Press, New York, pp 245–270
56. Penrose R (1989) *The Emperor's New Mind: Concerning Computers, Minds, and the Laws of Physics*. Oxford University Press, Oxford
57. Poundstone W (1985) *The Recursive Universe*. Contemporary Books, Chicago. 252 pp
58. Preziosi L (ed) (2003) *Cancer Modelling and Simulation*. Chapman, Boca Raton
59. Ray TS (1991) An Approach to the Synthesis of Life (Tierra Simulator). In: Langton CG, Taylor C, Farmer JD, Rasmussen S (eds) *Artificial Life II: Proceedings of the Workshop on Artificial Life*. Wesley, Redwood City, pp 371–408
60. Rechenberg I (1973) *Evolutionsstrategie: Optimierung Technischer Systeme Nach Prinzipien Der Biologischen Evolution*. Frommann-Holzboog, Stuttgart
61. Sakoda JM (1971) The Checkerboard Model of Social Interaction. *J Math Soc* 1:119–132
62. Schelling TC (1971) Dynamic Models of Segregation. *J Math Soc* 1:143–186
63. Searle JR (1990) Is the Brain a Digital Computer? Presidential Address to the American Philosophical Association
64. Taub AH (ed) (1961) *John Von Neumann: Collected Works*. vol V: Design of Computers, Theory of Automata and Numerical Analysis (Delivered at the Hixon Symposium, Pasadena, Sept 1948). Pergamon Press, Oxford
65. Turing AM (1938) On Computable Numbers with an Application to the Entscheidungsproblem. *Process Lond Math Soc* 2(42):230–265
66. Turing AM (1952) The Chemical Basis of Morphogenesis. *Philos Trans Royal Soc B* 237:37–72
67. von Neumann J (1966) *Theory of Self-Reproducing Automata*, edited by Burks AW. University of Illinois Press, Champaign
68. Wilke CO, Adami C (2002) The Biology of Digital Organisms. *Trends Ecol Evol* 17(11):528–532
69. Wilke CO, Chow SS (2006) Exploring the Evolution of Ecosystems with Digital Organisms. In: Pascual M, Dunne JA (eds) *Ecological Networks: Linking Structure to Dynamics in Food Webs*. Oxford University Press, New York, pp 271–286
70. Wolfram S (1984) *Universality and Complexity in Cellular Automata*. *Physica D*:1–35
71. Wolfram S (1999) *The Mathematica Book*, 4th edn. Wolfram Media/Cambridge University Press, Champaign
72. Wolfram S (2002) *A New Kind of Science*. Wolfram Media, Champaign

Books and Reviews

- Artificial Life (journal) web page (2008) <http://www.mitpressjournals.org/loi/artl>. Accessed 8 March 2008
- Banks ER (1971) *Information Processing and Transmission in Cellular Automata*. Ph.D. dissertation, Massachusetts Institute of Technology
- Batty M (2007) *Cities and Complexity: Understanding Cities with Cellular Automata, Agent-Based Models, and Fractals*. MIT Press, Cambridge
- Bedau MA (2002) The Scientific and Philosophical Scope of Artificial Life, *Leonardo* 35:395–400
- Bedau MA (2003) Artificial Life: Organization, Adaptation and Complexity from the Bottom Up. *TRENDS Cognit Sci* 7(11):505–512
- Copeland BJ (2004) *The Essential Turing*. Oxford University Press, Oxford. 613 pp
- Ganguly N, Sikdar BK, Deutsch A, Canright G, Chaudhuri PP (2008) A Survey on Cellular Automata. www.cs.unibo.it/bison/publications/CAsurvey.pdf
- Griffeath D, Moore C (eds) (2003) *New Constructions in Cellular Automata*. Santa Fe Institute Studies in the Sciences of Complexity Proceedings. Oxford University Press, New York, 360 pp
- Gutowitz H (ed) (1991) *Cellular Automata: Theory and Experiment*. Special Issue of *Physica D* 499 pp
- Hraber T, Jones PT, Forrest S (1997) The Ecology of Echo. *Artif Life* 3:165–190
- International Society for Artificial Life web page (2008) www.alife.org. Accessed 8 March 2008
- Jacob C (2001) *Illustrating Evolutionary Computation with Mathematics*. Academic Press, San Diego. 578 pp
- Michael CF, Fred WG, Jay A (2005) *Simulation Optimization: A Review, New Developments, and Applications*. In: Proceedings of the 37th conference on Winter simulation, Orlando
- Miller JH, Page SE (2007) *Complex Adaptive Systems: An Introduction to Computational Models of Social Life*. Princeton University Press, Princeton
- North MJ, Macal CM (2007) *Managing Business Complexity: Discovering Strategic Solutions with Agent-Based Modeling and Simulation*. Oxford University Press, New York
- Pascual M, Dunne JA (eds) (2006) *Ecological Networks: Linking Structure to Dynamics in Food Webs*. Santa Fe Institute Studies in the Sciences of the Complexity. Oxford University Press, New York
- Simon H (2001) *The Sciences of the Artificial*. MIT Press, Cambridge
- Sims K (1991) Artificial Evolution for Computer Graphics. *ACM SIG-GRAPH '91*, Las Vegas, July 1991, pp 319–328

- Sims K (1994) Evolving 3D Morphology and Behavior by Competition. *Artificial Life IV*, pp 28–39
- Terzopoulos D (1999) *Artificial Life for Computer Graphics*. Commun ACM 42(8):33–42
- Toffoli T, Margolus N (1987) *Cellular Automata Machines: A New Environment for Modeling*. MIT Press, Cambridge, 200 pp
- Tu X, Terzopoulos D, *Artificial Fishes: Physics, Locomotion, Perception, Behavior*. In: *Proceedings of SIGGRAPH'94*, 24–29 July 1994, Orlando, pp 43–50
- Weisbuch G (1991) *Complex Systems Dynamics: An Introduction to Automata Networks*, translated from French by Ryckebusch S. Addison-Wesley, Redwood City
- Wiener N (1948) *Cybernetics, or Control and Communication in the Animal and the Machine*. Wiley, New York
- Wooldridge M (2000) *Reasoning About Rational Agents*. MIT Press, Cambridge

Agent Based Modeling and Computer Languages

MICHAEL J. NORTH, CHARLES M. MACAL
Argonne National Laboratory, Decision and Information Sciences Division, Center for Complex Adaptive Agent Systems Simulation (CAS²), Argonne, USA

Article Outline

Glossary
Definition of the Subject
Introduction
Types of Computer Languages
Requirements of Computer Languages for Agent-Based Modeling
Example Computer Languages Useful for Agent-Based Modeling
Future Directions
Bibliography

Glossary

Agent An agent is a self-directed component in an agent-based model.

Agent-based model An agent-based model is a simulation made up of a set of agents and an agent interaction environment.

Annotations Annotations are a Java feature for including metadata in compiled code.

Attributes Attributes are a C# feature for including metadata in compiled code.

Aspects Aspects are a way to implement dispersed but recurrent tasks in one location.

Bytecode Bytecode is compiled Java binary code.

Common intermediate language Common Intermediate

Language (CIL) is compiled binary code for the Microsoft .NET Framework. CIL was originally called Microsoft Intermediate Language (MSIL).

Computational algebra systems Computational algebra systems (CAS) are computational mathematics systems that calculate using symbolic expressions.

Computational mathematics systems Computational Mathematics Systems (CMS) are software programs that allow users to apply powerful mathematical algorithms to solve problems through a convenient and interactive user interface. CMS typically supply a wide range of built-in functions and algorithms.

Class A class is the object-oriented inheritable binding of procedures and data that provides the basis for creating objects.

Computer language A computer language is a method of noting directives for computers. Computer programming languages, or more simply programming languages, are an important category of computer languages.

Computer programming language Please see glossary entry for “Programming language”.

C# C# (Archer [1]) is an object-oriented programming language that was developed and is maintained by Microsoft. C# is one of many languages that can be used to generate Microsoft .NET Framework code. This code is run using a ‘virtual machine’ that potentially gives it a consistent execution environment on different computer platforms.

C++ C++ is a widely used object-oriented programming language that was created by Bjarne Stroustrup (Stroustrup [39]) at AT&T. C++ is widely noted for both its object-oriented structure and its ability to be easily compiled into native machine code.

Design pattern Design patterns form a “common vocabulary” describing tried and true solutions for commonly faced software design problems (Coplien [6]).

Domain-specific language Domain-specific languages (DSL’s) are computer languages that are highly customized to support a well defined application area or ‘domain’. DSL’s commonly include a substantial number of keywords that are nouns and verbs in the area of application as well as overall structures and execution patterns that correspond closely with the application area.

Declarative language According to Watson [47] a “declarative language (or non-procedural language) involves the specification of a set of rules defining the solution to the problem; it is then up to the computer to determine how to reach a solution consistent with the given rules”.

Dynamic method invocation Dynamic method invocation, combined with reflection, is a Java and C# approach to higher-order programming.

Encapsulation Encapsulation is the containment of details inside a module.

Field A field is a piece of object-oriented data.

Functional language According to Watson [47] “in functional languages (sometimes called applicative languages) the fundamental operation is function application”.

Function pointers Function pointers are part of C++’s approach to higher-order programming. Runtime Type Identification is another component of this approach.

Generics Generics are a Java and C# feature for generalizing classes.

Goto statement A goto statement is an unconditional jump in a code execution flow.

Headless A headless program executes without the use of a video monitor. This is generally done to rapidly execute models while logging results to files or databases.

Higher-order programming

According to Reynolds [35], higher-order programming involves the use of “procedures or labels ... as data” such that they “can be used as arguments to procedures, as results of functions, or as values of assignable variables”.

Imperative language According to Watson [47] in imperative languages “there is a fundamental underlying dependence on the assignment operation and on variables implemented as computer memory locations, whose contents can be read and altered”.

Inheritance Inheritance is the ability of an object-oriented class to assume the methods and data of another class called the parent class.

Java Java (Foxwell [12]) is a widely used object-oriented programming language that was developed and is maintained by Sun Microsystems. Java is known for its widespread cross-platform availability on many different types of hardware and operating systems. This capability comes from Java’s use of a ‘virtual machine’ that allows code to have a consistent execution environment on many different computer platforms.

Logic programming language

According to Watson [47] “in a logic programming language, the programmer needs only to supply the problem specification in some formal form, as it is the responsibility of the language system to infer a method of solution”.

Macro language Macro languages are simple domain-

specific languages that are used to write script for tools such as spreadsheets.

Mathematica *Mathematica* is a commercial software program for computational mathematics. Information on *Mathematica* can be found at <http://www.wolfram.com/>.

Method A method is an object-oriented procedure.

Methodological individualism A reductionist approach to social science originally developed by Max Weber that focuses on the interaction of well defined and separate individuals (Heath [20]). Alternatives theories usually focus on more holistic views of interaction (Heath [20]).

MATLAB The MATrix Laboratory (MATLAB) is a commercial software program for computational mathematics. Information on MATLAB can be found at <http://www.mathworks.com/>.

Mobile agents Mobile agents are light-weight software proxies that roam the world-wide web and perform various functions programmed by their owners such as gathering information from web sites.

Module According to Stevens et al. [38], “the term module is used to refer to a set of one or more contiguous program statements having a name by which other parts of the system can invoke it and preferably having its own distinct set of variable names”.

NetLogo NetLogo (Wilensky [48]) is an agent-based modeling and simulation platform that uses a domain-specific language to define models. NetLogo models are built using a metaphor of turtles as agents and patches as environmental components (Wilensky [48]). NetLogo itself is Java-based. NetLogo is free for use in education and research. More information on NetLogo and downloads can be found at <http://ccl.northwestern.edu/netlogo/>.

Non-procedural language Please see glossary entry for “Declarative language”.

Object An object is the instantiation of a class to produce executable instances.

Object-oriented language Object-oriented languages are structured languages that have special features for binding data with procedures; inheritance; encapsulation; and polymorphism. Careful abstraction that omits unimportant details is an important design principle associated with the use of object-oriented languages.

Objective-C Objective-C is an object-oriented language that extends the C language.

Observer The observer is a NetLogo agent that has a view of an entire model. There is exactly one observer in every NetLogo model.

Patch A patch is a NetLogo agent with a fixed location on a master grid.

Polymorphism Polymorphism is the ability of an object-oriented class to respond to multiple related messages, often method calls with the same name but different parameters.

Procedural language According to Watson [47] “procedural languages ... are those in which the action of the program is defined by a series of operations defined by the programmer”.

Programming language A programming language is a computer language that allows any computable activity to be expressed.

Record A record is an independently addressable collection of data items.

Reflection Reflection, combined with dynamic method invocation, is a Java and C# approach to higher-order programming.

Repast The Recursive Porous Agent Simulation Toolkit (Repast) is a free and open source family of agent-based modeling and simulation platforms (ROAD [44]). Information on Repast and free downloads can be found at <http://repast.sourceforge.net/>.

Repast simphony Repast Simphony (Repast S) is the newest member of the Repast family of free and open source agent-based modeling and simulation platforms (North et al. [32], North et al. [30]). The Java-based Repast S system includes advanced features for specifying, executing, and analyzing agent-based simulations.

Repast simphony score Repast Simphony Score is an XML metadata file format that describes the components (e.g., agents and spaces) allowed in a Repast Simphony simulation.

Runtime type identification Runtime Type Identification (RTTI) is part of C++’s approach to higher-order programming. Function pointers are another component of this approach.

Structured language Structured languages are languages that divide programs into separate modules each of which has one controlled entry point, a limited number of exit points, and no internal jumps (Dijkstra [10]).

Swarm Swarm (Swarm Development Group [40]) is a free and open source agent-based modeling and simulation platform maintained by the Swarm Development Group. The core Swarm system uses Objective-C. A Java-based “Java Swarm” wrapper for the Objective-C core is also available. Information on Swarm and free downloads can be found at <http://www.swarm.org/>.

Templates Templates are a C++ feature for generalizing classes.

Turtle A turtle is a mobile NetLogo agent.

Virtual machine A virtual machine is a software environment that allows user code to have a consistent execution environment on many different computer platforms.

Unstructured language Unstructured languages are languages that rely on step-by-step solutions such that the solutions can contain arbitrary jumps between steps.

Definition of the Subject

Agent-based modeling is a bottom-up approach to representing and investigating complex systems. Agent-based models can be implemented either computationally (e.g., through computer simulation) or non-computationally (e.g., with participatory simulation). The close match between the capabilities of computational platforms and the requirements of agent-based modeling make these platforms a natural choice for many agent-based models. Of course, realizing the potential benefits of this natural match necessitates the use of computer languages to express the designs of agent-based models. A wide range of computer programming languages can play this role including both domain-specific and general purpose languages. The domain-specific languages include business-oriented languages (e.g., spreadsheet programming tools); science and engineering languages (e.g., Mathematica); and dedicated agent-based modeling languages (e.g., NetLogo). The general purpose languages can be used directly (e.g., Java programming) or within agent-based modeling toolkits (e.g., Repast). The choice that is most appropriate for each modeling project depends on both the requirements of that project and the resources available to implement it.

Introduction

The term agent-based modeling (ABM) refers to the computational modeling of a system as comprised of a number of independent, interacting entities, which are referred to as ‘agents’. Generally, an agent-based system is made up of agents that interact, adapt, and sustain themselves while interacting with other agents and adapting to a changing environment. The fundamental feature of an agent is its autonomy, the capability of the agent to act independently without the need for direction from external sources. Agents have behaviors that make them active rather than passive entities. Agent behaviors allow agents to take in information from their environment, which in-

cludes their interactions with other agents, process the information and make some decision about their next action, and take the action. Jennings [21] provides a rigorous computer science view of agency emphasizing the essential characteristic of autonomous behavior.

Beyond the essential characteristic of autonomy, there is no universal agreement on the precise definition of the term “agent”, as used in agent-based modeling. Some consider any type of independent component, whether it be a software model or a software model of an extant individual, to be an agent [4]. An independent component’s behaviors can be modeled as consisting of anything from simple reactive decision rules to multi-dimensional behavior complexes based on adaptive artificial intelligence (AI) techniques.

Other authors insist that a component’s behavior must be adaptive in order for the entity to be considered an agent. The agent label is reserved for components that can adapt to their environment, by learning from the successes and failures with their interactions with other agents, and change their behaviors in response. Casti [5] argues that agents should contain both base-level rules for behavior as well as a higher-level set of “rules to change the rules”. The base-level rules provide responses to the environment while the “rules to change the rules” provide adaptation [5].

From a practical modeling standpoint, agent characteristics can be summarized as follows:

- Agents are identifiable as self-contained individuals. An agent has a set of characteristics and rules governing its behaviors.
- Agents are autonomous and self-directed. An agent can function independently in its environment and in its interactions with other agents, at least over a limited range of situations that are of interest.
- An agent is situated, living in an environment with which it interacts along with other agents. Agents have the ability to recognize and distinguish the traits of other agents. Agents also have protocols for interaction with other agents, such as for communication, and the capability to respond to the environment.
- An agent may be goal-directed, having targets to achieve with respect to its behaviors. This allows an agent to compare the outcome of its behavior to its goals. An agent’s goals need not be comprehensive or well-defined. For example, an agent does not necessarily have formally stated objectives it is trying to maximize.
- An agent might have the ability to learn and adapt its behaviors based on its experiences. An agent might

have rules that modify its behavior over time. Generally, learning and adaptation at the agent level requires some form of memory to be built into the agents behaviors.

Often, in an agent-based model, the population of agents varies over time, as agents are born and die. Another form of adaptation can occur at the agent population level. Agents that are fit are better able to sustain themselves and possibly reproduce as time in the simulation progresses, while agents that have characteristics less suited to their continued survival are excluded from the population.

Another basic assumption of agent-based modeling is that agents have access only to local information. Agents obtain information about the rest of the world only through their interactions with the limited number of agents around them at any one time, and from their interactions with a local patch of the environment in which they are situated.

These aspects of how agent-based modeling treats agents highlight the fact that the full range of agent diversity can be incorporated into an agent-based model. Agents are diverse and heterogeneous as well as dynamic in their attributes and behavioral rules. There is no need to make agents homogeneous through aggregating agents into groups or by identifying the ‘average’ agent as representative of the entire population. Behavioral rules vary in their sophistication, how much information is considered in the agent decisions (i.e., cognitive ‘load’), the agent’s internal models of the external world including the possible reactions or behaviors of other agents, and the extent of memory of past events the agent retains and uses in its decisions. Agents can also vary by the resources they have manage to accumulate during the simulation, which may be due to some advantage that results from specific attributes. The only limit on the number of agents in an agent-based model is imposed by the computational resources required to run the model.

As a point of clarification, agent-based modeling is also known by other names. ABS (agent-based systems), IBM (individual-based modeling), and MAS (multi-agent systems) are widely-used acronyms, but ‘ABM’ will be used throughout this discussion. The term ‘agent’ has connotations other than how it is used in ABM. For example, ABM agents are different from the typical agents found in mobile agent systems. ‘Mobile agents’ are light-weight software proxies that roam the world-wide web and perform various functioned programmed by their owners such as gathering information from web sites. To this extent, mobile agents are autonomous and share this characteristic with agents in ABM.

Types of Computer Languages

A ‘computer language’ is a method of noting directives for computers. ‘Computer programming languages,’ or ‘programming languages,’ are an important category of computer languages. A programming language is a computer language that allows any computable activity to be expressed. This article focuses on computer programming languages rather than the more general computer languages since virtually all agent-based modeling systems require the power of programming languages. This article sometimes uses the simpler term ‘computer languages’ when referring to computer programming languages. According to Watson [47]:

Programming languages are used to describe algorithms, that is, sequences of steps that lead to the solution of problems ... A programming language can be considered to be a ‘notation’ that can be used to specify algorithms with precision.

Watson [47] goes on to say that “programming languages can be roughly divided into four groups: imperative languages, functional languages, logic programming languages, and others”. Watson [47] states that in imperative languages “there is a fundamental underlying dependence on the assignment operation and on variables implemented as computer memory locations, whose contents can be read and altered”. However, “in functional languages (sometimes called applicative languages) the fundamental operation is function application” [47]. Watson cites LISP as an example. Watson [47] continues by noting that “in a logic programming language, the programmer needs only to supply the problem specification in some formal form, as it is the responsibility of the language system to infer a method of solution”.

A useful feature of most functional languages, many logic programming languages, and some imperative languages is higher-order programming. According to Reynolds [35]:

In analogy with mathematical logic, we will say that a programming language is higher-order if procedures or labels can occur as data, i. e., if these entities can be used as arguments to procedures, as results of functions, or as values of assignable variables. A language that is not higher-order will be called first-order.

Watson [47] offers that “another way of grouping programming languages is to classify them as procedural or declarative languages”. Elaborating, Watson [47] states that:

Procedural languages ... are those in which the action of the program is defined by a series of operations defined by the programmer. To solve a problem, the programmer has to specify a series of steps (or statements) which are executed in sequence.

On the other hand, Watson [47] notes that:

Programming in a declarative language (or non-procedural language) involves the specification of a set of rules defining the solution to the problem; it is then up to the computer to determine how to reach a solution consistent with the given rules ... The language Prolog falls into this category, although it retains some procedural aspects. Another widespread non-procedural system is the spreadsheet program.

Imperative and functional languages are usually procedural while logic programming languages are generally declarative. This distinction is important since it implies that most imperative and functional languages require users to define how each operation is to be completed while logic programming languages only require users to define what is to be achieved. However, when faced with multiple possible solutions with different execution speeds and memory requirements, imperative and functional languages offer the potential for users to explicitly choose more efficient implementations over less efficient ones. Logic programming languages generally need to infer which solution is best from the problem description and may or may not choose the most efficient implementation. Naturally, this potential strength of imperative and functional languages may also be cast as a weakness. With imperative and functional language users need to correctly choose a good implementation among any competing candidates that may be available.

Similarly to Watson [47], Van Roy and Haridi [45] define several common computational models namely those that are object-oriented, those that are logic-based, and those that are functional. Object-oriented languages are procedural languages that bind procedures (i. e., ‘encapsulated methods’) to their corresponding data (i. e., ‘fields’) in nested hierarchies (i. e., ‘inheritance’ graphs) such that the resulting ‘classes’ can be instantiated to produce executable instances (i. e., ‘objects’) that respond to multiple related messages (i. e., ‘polymorphism’). Logic-based languages correspond to Watson’s [47] logic programming languages. Similarly, Van Roy and Haridi [45] functional languages correspond to those of Watson [47].

Two additional types of languages can be added to Van Roy and Haridi’s [45] list of three. These are unstruc-

tured and structured languages [10]. Both unstructured and structured languages are procedural languages.

Unstructured languages are languages that rely on step-by-step solutions such that the solutions can contain arbitrary jumps between steps [10]. BASIC, COBOL, FORTRAN, and C are examples of unstructured languages. The arbitrary jumps are often implemented using ‘goto’ statements. Unstructured languages were famously criticized by Edsger Dijkstra in his classic paper “Go To Statement Considered Harmful” [10]. This and related criticism lead to the introduction of structured languages.

Structured languages are languages that divide programs into separate modules each of which has one controlled entry point, a limited number of exit points, and no internal jumps [10]. Following Stevens et al. [38] “the term module is used to refer to a set of one or more contiguous program statements having a name by which other parts of the system can invoke it and preferably having its own distinct set of variable names”. Structured language modules, often called procedures, are generally intended to be small. As such, large numbers of them are usually required to solve complex problems. Standard Pascal is an example of structured, but not object-oriented, language. As stated earlier, C is technically an unstructured language (i. e., it allows jumps within procedures and ‘long jumps’ between procedures), but it is used so often in a structured way that many people think of it as a structured language.

The quality of modularization in structured language code is often considered to be a function of coupling and cohesion [38]. Coupling is the tie between modules such that the proper functioning of one module depends on the functioning of another module. Cohesion is the ties within a module such that proper functioning of one line of code in a module depends on the functioning of another one line of code in the same module. The goal for modules is maximizing cohesion while minimizing coupling.

Object-oriented languages are a subset of structured languages. Object-oriented methods and classes are structured programming modules that have special features for binding data, inheritance, and polymorphism. The previously introduced concepts of coupling and cohesion apply to classes, objects, methods, and fields the same way that they apply to generic structured language modules. Objective-C, C++, C#, and Java are all examples of object-oriented languages. As with C, the languages Objective-C, C++, and C# offer goto statements but they have object-oriented features and are generally used in a structured way. Java is an interesting case in that the word ‘goto’ is reserved as a keyword in the language specification, but it is not intended to be implemented.

It is possible to develop agent-based models using any of the programming languages discussed above namely, unstructured languages, structured languages, object-oriented languages, logic-based languages, and functional languages. Specific examples are provided later in this article. However, certain features of programming languages are particularly well suited for supporting the requirements of agent-based modeling and simulation.

Requirements of Computer Languages for Agent-Based Modeling

The requirements of computer languages for agent-based modeling and simulation include the following:

- There is a need to create well defined modules that correspond to agents. These modules should bind together agent state data and agent behaviors into integrated independently addressable constructs. Ideally these modules will be flexible enough to change structure over time and to optionally allow fuzzy boundaries to implement models that go beyond methodological individualism [20].
- There is a need to create well defined containers that correspond to agent environments. Ideally these containers will be recursively nestable or will otherwise support sophisticated definitions of containment.
- There is a need to create well defined spatial relationships within agent environments. These relationships should include notions of abstract space (e. g., lattices), physical space (e. g., maps), and connectedness (e. g., networks).
- There is a need to easily setup model configurations such as the number of agents; the relationships between agents; the environmental details; and the results to be collected.
- There is a need to conveniently collect and analyze model results.

Each of the kinds of programming languages namely, unstructured languages, structured languages, object-oriented languages, logic-based languages, and functional languages can address these requirements.

Unstructured languages generally support procedure definitions which can be used to implement agent behaviors. They also sometimes support the collection of diverse data into independently addressable constructs in the form of data structures often called ‘records’. However, they generally lack support for binding procedures to individual data items or records of data items. This lack of support for creating integrated constructs also typically limits the

language-level support for agent containers. Native support for implementing spatial environments is similarly limited by the inability to directly bind procedures to data.

As discussed in the previous section, unstructured languages offer statements to implement execution jumps. The use of jumps within and between procedures tends to reduce module cohesion and increase module coupling compared to structured code. The result is reduced code maintainability and extensibility compared to structured solutions. This is a substantial disadvantage of unstructured languages.

In contrast, many have argued that, at least theoretically, unstructured languages can achieve the highest execution speed and lowest memory usage of the language options since nearly everything is left to the application programmers. In practice, programmers implementing agent-based models in unstructured languages usually need to write their own tools to form agents by correlating data with the corresponding procedures. Ironically, these tools are often similar in design, implementation, and performance to some of the structured and object-oriented features discussed later.

Unstructured languages generally do not provide special support for application data configuration, program output collection, or program results analysis. As such, these tasks usually need to be manually implemented by model developers.

In terms of agent-based modeling, structured languages are similar to unstructured languages in that they do not provide tools to directly integrate data and procedures into independently addressable constructs. Therefore, structured language support for agents, agent environments, and agent spatial relationships is similar to that provided by unstructured languages. However, the lack of jump statements in structured languages tends to increase program maintainability and extensibility compared to unstructured languages. This generally gives structured languages a substantial advantage over unstructured languages for implementing agent-based models.

Object-oriented languages build on the maintainability and extensibility advantages of structured languages by adding the ability to bind data to procedures. This binding in the form of classes provides a natural way to implement agents. In fact, object-oriented languages have their roots in Ole-Johan Dahl and Kristen Nygaard's Simula simulation language [7,8,45]! According to Dahl and Nygaard [7]:

SIMULA (SIMULAtion LANGUAGE) is a language designed to facilitate formal description of the layout and rules of operation of systems with discrete

events (changes of state). The language is a true extension of ALGOL 60 [2], i.e., it contains ALGOL 60 as a subset. As a programming language, apart from simulation, SIMULA has extensive list processing facilities and introduces an extended co-routine concept in a high-level language.

Dahl and Nygaard go on to state the importance of specific languages for simulation [7] as follows:

Simulation is now a widely used tool for analysis of a variety of phenomena: nerve networks, communication systems, traffic flow, production systems, administrative systems, social systems, etc. Because of the necessary list processing, complex data structures and program sequencing demands, simulation programs are comparatively difficult to write in machine language or in ALGOL or FORTRAN. This alone calls for the introduction of simulation languages.

However, still more important is the need for a set of basic concepts in terms of which it is possible to approach, understand and describe all the apparently very different phenomena listed above. A simulation language should be built around such a set of basic concepts and allow a formal description which may generate a computer program. The language should point out similarities and differences between systems and force the research worker to consider all relevant aspects of the systems. System descriptions should be easy to read and print and hence useful for communication.

Again, according to Dahl and Nygaard [8]:

SIMULA I (1962–65) and Simula 67 (1967) are the two first object-oriented languages. Simula 67 introduced most of the key concepts of object-oriented programming: both objects and classes, subclasses (usually referred to as inheritance) and virtual procedures, combined with safe referencing and mechanisms for bringing into a program collections of program structures described under a common class heading (prefixed blocks).

The Simula languages were developed at the Norwegian Computing Center, Oslo, Norway by Ole-Johan Dahl and Kristen Nygaard. Nygaard's work in Operational Research in the 1950s and early 1960s created the need for precise tools for the description and simulation of com-

plex man-machine systems. In 1961 the idea emerged for developing a language that both could be used for system description (for people) and for system prescription (as a computer program through a compiler). Such a language had to contain an algorithmic language, and Dahl's knowledge of compilers became essential ... When the inheritance mechanism was invented in 1967, Simula 67 was developed as a general programming language that also could be specialized for many domains, including system simulation.

Generally, object-oriented classes are used to define agent templates and instantiated objects are used to implement specific agents. Agent environment templates and spatial relationships patterns are also typically implemented using classes. Recursive environment nesting as well as abstract spaces, physical spaces, and connectedness can all be represented in relatively straightforward ways. Instantiated objects are used to implement specific agent environments and spatial relationships in individual models. Within these models, model configurations are also commonly implemented as objects instantiated from one or more classes. However, as with unstructured and structured languages, object-oriented languages generally do not provide special support for application data configuration, program output collection, or program results analysis. As such, these tasks usually need to be manually implemented by model developers. Regardless of this, the ability to bind data and procedures provides such a straightforward method for implementing agents that most agent-based models are written using object-oriented languages.

It should be noted that traditional object-oriented languages do not provide a means to modify class and object structures once a program begins to execute. Newer 'dynamic' object-oriented languages such as Groovy [22] offer this capability. This potentially allows agents to gain and lose data items and methods during the execution of a model based on the flow of events in a simulation. This in turn offers the possibility of implementing modules with fuzzy boundaries that are flexible enough to change structure over time.

As discussed in the previous section, logic-based languages offer an alternative to the progression formed by unstructured, structured, and object-oriented languages. Logic-based languages can provide a form of direct support for binding data (e. g., asserted propositions) with actions (e. g., logical predicates), sometimes including the use of higher-order programming. In principle, each agent can be implemented as a complex predicate with multiple nested sub-terms. The sub-terms, which may contain unresolved variables, can then be activated and resolved as

needed during model execution. Agent templates which are analogous to object-oriented classes can be implemented using the same approach but with a larger number of unresolved variables. Agent environments and the resulting relationships between agents can be formed in a similar way. Since each of these constructs can be modified at any time, the resulting system can change structure over time and may even allow fuzzy boundaries. In practice this approach is rarely, if ever, used. As with the previously discussed approaches, logic-based languages usually do not provide special support for application data configuration, program output collection, or program results analysis so these usually need to be manually developed.

Functional languages offer yet another alternative to the previously discussed languages. Like logic-based and object-oriented languages, functional languages often provide a form of direct support for binding data with behaviors. This support often leverages the fact that most functional languages support higher-order programming. As a result, the data is usually in the form of nested lists of values and functions while the behaviors themselves are implemented in the form of functions. Agent templates (i. e., 'classes'), agent environments, and agent relationships can be implemented similarly. Each of the lists can be dynamically changed during a simulation run so the model structure can evolve and can potentially have fuzzy boundaries. Unlike the other languages discussed so far, a major class of functional languages, namely those designed for computational mathematics usually include sophisticated support for program output collection and results analysis. An example is *Mathematica* (Wolfram [49]). If the application data is configured in mathematically regular ways then these systems may also provide support for application data setup.

Example Computer Languages Useful for Agent-Based Modeling

Domain-Specific Languages

Domain-specific languages (DSL's) are computer languages that are highly customized to support a well defined application area or 'domain'. DSL's commonly include a substantial number of keywords that are nouns and verbs in the area of application as well as overall structures and execution patterns that correspond closely with the application area. DSL's are intended to allow users to write in a language that is closely aligned with their area of expertise.

DSL's often gain their focus by losing generality. For many DSL's there are activities that can be programmed in most computer languages that cannot be programmed

in the given DSL. This is consciously done to simplify the DSL's design and make it easier to learn and use. If a DSL is properly designed then the loss of generality is often inconsequential for most uses since the excluded activities are chosen to be outside the normal range of application. However, even the best designed DSL's can occasionally be restrictive when the bounds of the language are encountered. Some DSL's provide special extension points that allow their users to program in a more general language such as C or Java when the limits of the DSL are reached. This feature is extremely useful, but requires more sophistication on the part of the user in that they need to know and simultaneously use both the DSL and the general language.

DSL's have the potential to implement specific features to support 'design patterns' within a given domain. Design patterns form a "common vocabulary" describing tried and true solutions for commonly faced software design problems (Coplien [6]). Software design patterns were popularized by Gamma et al. [13]. North and Macal [29] describe three design patterns for agent-based modeling itself.

In principle, DSL's can be unstructured, structured, object-oriented, logic-based, or functional. In practice, DSL's are often structured languages or object-oriented languages and occasionally are functional languages. Commonly used ABM DSL's include business-oriented languages (e. g., spreadsheet programming tools); science and engineering languages (e. g., *Mathematica*); and dedicated agent-based modeling languages (e. g., NetLogo).

Business Languages Some of the most widely used business computer languages are those available in spreadsheet packages. Spreadsheets are usually programmed using a 'macro language'. As discussed further in North and Macal [29], any modern spreadsheet program can be used to do basic agent-based modeling. The most common convention is to associate each row of a primary spreadsheet worksheet with an agent and use consecutive columns to store agent properties. Secondary worksheets are then used to represent the agent environment and to provide temporary storage for intermediate calculations. A simple loop is usually used to scan down the list of agents and to allow each one to execute in turn. The beginning and end of the scanning loop are generally used for special setup activities before and special cleanup activities after each round. An example agent spreadsheet from North and Macal [29] is shown in Fig. 1 and Fig. 2. Agent spreadsheets have both strengths and weaknesses compared to the other ABM tools. Agent spreadsheets tend to be easy to build but they also tend to have limited capabilities. This bal-

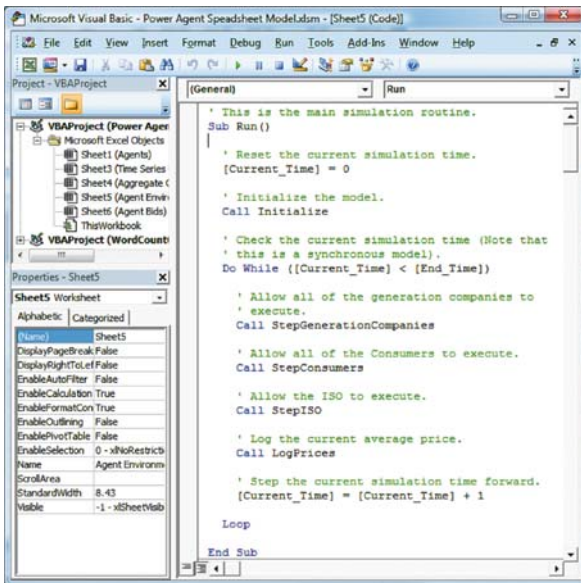
ance makes spreadsheets ideal for agent-based model exploration, scoping, and prototyping. Simple agent models can be implemented on the desktop using environments outside of spreadsheets as well.

Science and Engineering Languages Science and engineering languages embodied in commercial products such as *Mathematica*, MATLAB, Maple, and others can be used as a basis for developing agent-based models. Such systems usually have a large user base, are readily available on desktop platforms, and are widely integrated into academic training programs. They can be used as rapid prototype development tools or as components of large-scale modeling systems. Science and engineering languages have been applied to agent-based modeling. Their advantages include a fully integrated development environment, their interpreted (as opposed to compiled) nature provides immediate feedback to users during the development process, and a packaged user interface. Integrated tools provide support for data import and graphical display. Macal [25] describes the use of *Mathematica* and MATLAB in agent-based simulation and Macal and Howe [26] detail investigations into linking *Mathematica* and MATLAB to the Repast ABM toolkit to make use of Repast's simulation scheduling algorithms. In the following sections we focus on MATLAB and *Mathematica* as representative examples of science and engineering languages.

MATLAB and *Mathematica* are both examples of Computational Mathematics Systems (CMS). CMS allow users to apply powerful mathematical algorithms to solve problems through a convenient and interactive user interface. CMS typically supply a wide range of built-in functions and algorithms. MATLAB, *Mathematica*, and Maple are examples of commercially available CMS whose origins go back to the late 1980s. CMS are structured in two main parts: (1) the user interface that allows dynamic user interaction, and (2) the underlying computational engine, or kernel, that performs the computations according to the user's instructions. Unlike conventional programming languages, CMS are interpreted instead of compiled, so there is immediate feedback to the user, but some performance penalty is paid. The underlying computational engine is written in the C programming language for these systems, but C coding is unseen by the user. The most recent releases of CMS are fully integrated systems, combining capabilities for data input and export, graphical display, and the capability to link to external programs written in conventional languages such as C or Java using inter-process communication protocols. The powerful features of CMS, their convenience of use, the need to

Agent Name	Agent Type	Used Capacity (MWh)	Available Capacity (MWh)	Target Capacity (MWh)	Cost or Value (\$/MWh)	Initial Price (\$)	Current Price (\$)	Current Profit (\$)	Last Profit (\$)
Company 1	Genco	141	141	58	\$9.09	\$10.00	\$10.00	\$815.14	\$237.98
Company 2	Genco	129	129	37	\$9.04	\$10.00	\$10.00	\$751.77	\$223.73
Company 3	Genco	113	113	15	\$10.99	\$10.99	\$10.99	\$438.30	\$0.00
Company 4	Genco	102	199	197	\$9.87	\$9.87	\$14.87	\$508.92	\$180.03
Company 5	Genco	0	195	186	\$10.30	\$10.30	\$15.30	\$0.00	\$92.84
Company 6	Genco	192	192	177	\$10.39	\$10.39	\$10.39	\$859.80	\$73.89
Company 7	Genco	163	163	103	\$10.78	\$10.78	\$10.78	\$667.21	\$0.00
Company 8	Genco	151	151	76	\$10.86	\$10.86	\$10.86	\$605.16	\$0.00
Company 9	Genco	0	200	200	\$9.95	\$9.95	\$14.95	\$0.00	\$165.07
Company 10	Genco	149	149	73	\$10.87	\$10.87	\$10.87	\$595.83	\$0.00
Consumer 1	CONSUMER	1.68		1.68	\$27.98		\$14.87		
Consumer 2	CONSUMER	3.96		3.96	\$23.42		\$14.87		
Consumer 3	CONSUMER	2.60		2.60	\$23.67		\$14.87		
Consumer 4	CONSUMER	1.15		1.15	\$19.60		\$14.87		
Consumer 5	CONSUMER	3.85		3.85	\$30.00		\$14.87		
Consumer 6	CONSUMER	0.95		3.00	\$10.85		\$14.87		
Consumer 7	CONSUMER	0.03		0.60	\$10.40		\$14.87		

Agent Based Modeling and Computer Languages, Figure 1
An example agent spreadsheet [29]



Agent Based Modeling and Computer Languages, Figure 2
An example agent spreadsheet code [29]

learn only a limited number of instructions on the part of the user, and the immediate feedback provided to users are features of CMS that make them good candidates for developing agent-based simulations.

A further distinction can be made among CMS. A subset of CMS are what is called Computational Algebra Systems (CAS). CAS are computational mathematics systems that calculate using symbolic expressions. CAS owe their origins to the LISP programming language, which was the earliest functional programming language [24]. Macsyma (www.scientek.com/macsyma) and Scheme [37] (www.swiss.ai.mit.edu/projects/scheme) are often mentioned as important implementations leading to present day CAS. Typical uses of CAS are equation solving, symbolic integration and differentiation, exact calculations in linear algebra, simplification of mathematical expressions, and variable precision arithmetic. Computational mathematics systems consist of numeric processing systems or symbolic processing systems, or possibly a combination of both. Especially when algebraic and numeric capabilities are combined into a multi-paradigm programming environment, new modeling possibilities open up for developing sophisticated agent-based simulations with minimal coding.

Mathematica *Mathematica* is a commercially available numeric processing system with enormous integrated numerical processing capability (<http://www.wolfram.com>). Beyond numeric processing, *Mathematica* is a fully functional programming language. Unlike MATLAB, *Mathematica* is a symbolic processing system that uses term replacement as its primary operation. Symbolic processing

means that variables can be used before they have values assigned to them; in contrast a numeric processing language requires that every variable have a value assigned to it before it is used in the program. In this respect, although *Mathematica* and MATLAB may appear similar and share many capabilities, *Mathematica* is fundamentally much different than MATLAB, with a much different style of programming and ultimately with a different set of capabilities applicable to agent-based modeling.

Mathematica's symbolic processing capabilities allow one to program in multiple programming styles, either as alternatives or in combination, such as functional programming, logic programming, procedural programming, and even object-oriented programming styles. Like MATLAB, *Mathematica* is also an interpreted language, with the kernel of *Mathematica* running in the background in C. In terms of data types, everything is an expression in *Mathematica*. An expression is a data type with a head and a list of arguments in which even the head of the expression is part of the expression's arguments.

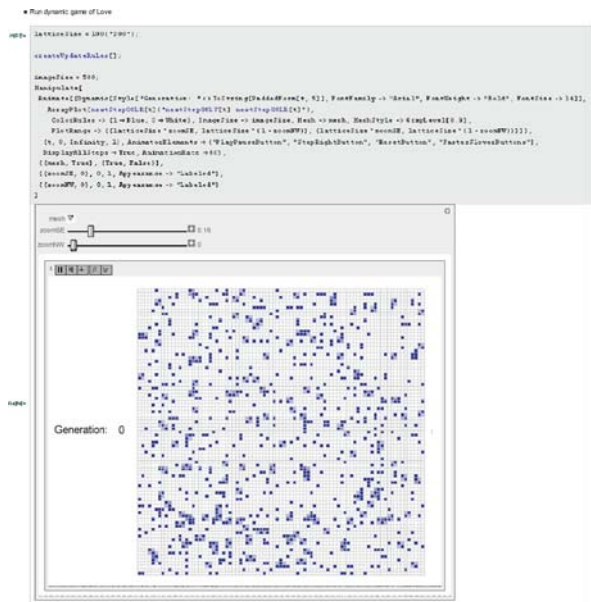
The *Mathematica* user interface consists of what is referred to as a notebook (Fig. 3). A *Mathematica* notebook is a fully integratable development environment and a complete publication environment. The *Mathematica* Application Programming Interface (API) allows programs written in C, FORTRAN, or Java to interact with *Mathematica*. The API has facilities for dynamically calling routines from *Mathematica* as well as calling *Mathematica* as a computational engine.

Figure 3 shows *Mathematica* desktop notebook environment. A *Mathematica* notebook is displayed in its own window. Within a notebook, each item is contained in a cell. The notebook cell structure has underlying coding that is accessible to the user.

In *Mathematica*, a network representation consists of combining lists of lists, or more generally expressions of expressions, to various depths. For example, in *Mathematica*, an agent can be represented explicitly as an expression that includes a head named `agent`, a sequence of agent attributes, and a list of the agent's neighbors. Agent data and methods are linked together by the use of what are called up values.

Example references for agent-based simulation using *Mathematica* include Gaylord and Davis [15], Gaylord and Nishidate [16], and Gaylord and Wellin [17]. Gaylord and D’Andria [14] describe applications in social agent-based modeling.

MATLAB The MATrix LABoratory (MATLAB) is a numeric processing system with enormous integrated numerical processing capability (<http://www.mathworks.com>).

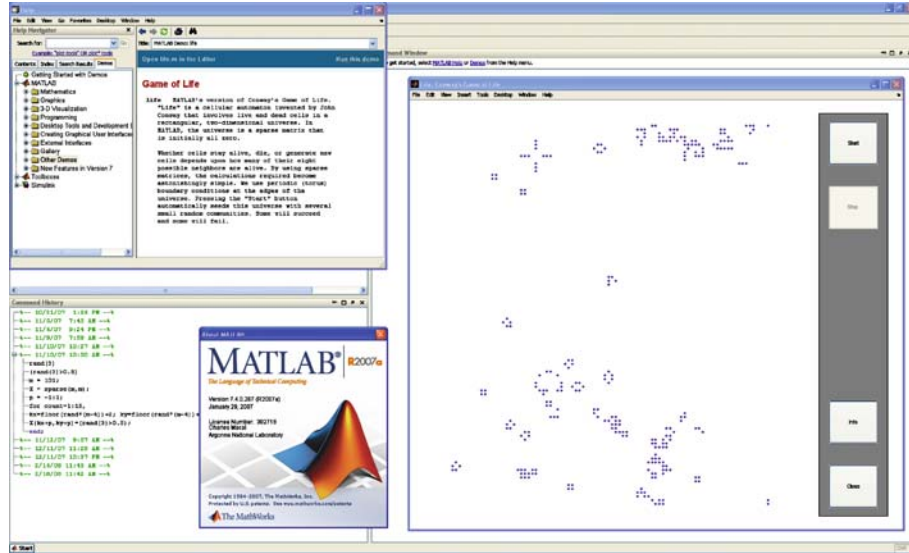


Agent Based Modeling and Computer Languages, Figure 3 Example Mathematica cellular automata model

com). It uses a scripting-language approach to programming. MATLAB is a high-level matrix/array language with control flow, functions, data structures, input/output, and object-oriented programming features. The user interface consists of the MATLAB Desktop, which is a fully integrated and mature development environment. There is an application programming interface (API). The MATLAB API allows programs written in C, Fortran, or Java to interact with MATLAB. There are facilities for calling routines from MATLAB (dynamic linking) as well as routines for calling MATLAB as a computational engine, as well as for reading and writing specialized MATLAB files.

Figure 4 shows the MATLAB Desktop environment illustrating the Game of Life, which is a standard MATLAB demonstration. The desktop consist of four standard windows: a command window, which contains a command line, the primary way of interacting with MATLAB, the workspace, which indicates the values of all the variables currently existing in the session, a command history window that tracks the entered command, and the current directory window. Other windows allow text editing of programs and graphical output display.

When it comes to agent-based simulation, as in most types of coding, the most important indicator of the power of a language for modeling is the extent of and the sophistication of the allowed data types and data structures. As Sedgewick [36] observes:



Agent Based Modeling and Computer Languages, Figure 4
Example MATLAB cellular automata model

For many applications, the choice of the proper data structure is really the only major decision involved in the implementation; once the choice has been made only very simple algorithms are needed [36].

The flexibility of data types plays an important role in developing large-scale, extensible models for agent-based simulation. In MATLAB the primary data type is the double array, which is essentially a two-dimensional numeric matrix. Other data types include logical arrays, cell arrays, structures, and character arrays.

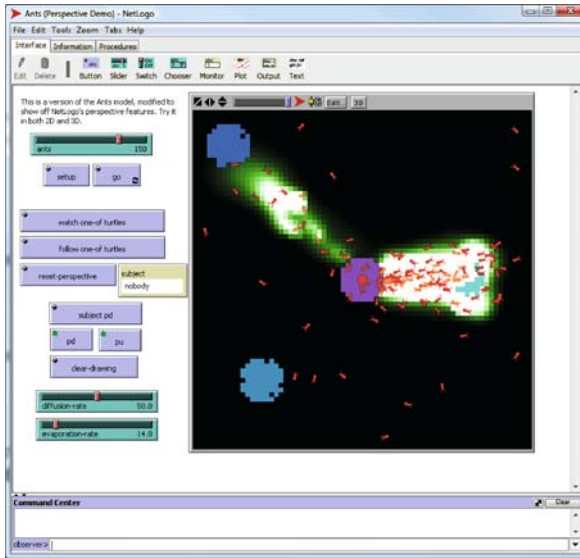
For agent-based simulations that define agent relationships based on networks, connectivity of the links defines the scope of agent interaction and locally available information. Extensions to modeling social networks require the use of more complex data structures than the matrix structure commonly used for grid representations. Extensions from grid topologies to network topologies are straightforward in MATLAB and similarly in *Mathematica*. In MATLAB, a network representation consists of combining cell arrays or structures in various ways.

The MATLAB desktop environment showing the Game of Life demonstration appears in Fig. 4. The Game of Life is a cellular automaton invented by mathematician John Conway that involves live and dead cells in cellular automata grid. In MATLAB, the agent environment is a sparse matrix that is initially set to all zeros. Whether cells stay alive, die, or generate new cells depends upon how many of their eight possible neighbors

are alive. By using sparse matrices, the calculations required become very simple. Pressing the “Start” button automatically seeds this universe with several small random communities and initiates a series of cell updates. After a short period of simulation, the initial random distribution of live (i. e., highlighted) cells develops into sets of sustainable patterns that endure for generations.

Several agent-based models using MATLAB have been published in addition to the Game of Life. These include a model of political institutions in modern Italy [3], a model of pair interactions and attitudes [34], a bargaining model to simulate negotiations between water users [43], and a model of sentiment and social mitosis based on Heider’s Balance Theory [18,46]. The latter model uses Euler, a MATLAB-like language. Thorngate argues for the use of MATLAB as an important tool to teach simulation programming techniques [42].

Dedicated Agent-Based Modeling Languages Dedicated agent-based modeling languages are DSL’s that are designed to specifically support agent-based modeling. Several such languages currently exist. These languages are functionally differentiated by the underlying assumptions their designers made about the structures of agent-based models. The designers of some of these languages assume quite a lot about the situations being modeled and use this information to provide users with pre-completed or template components. The designers of other languages make comparatively fewer assumptions and encourage users to



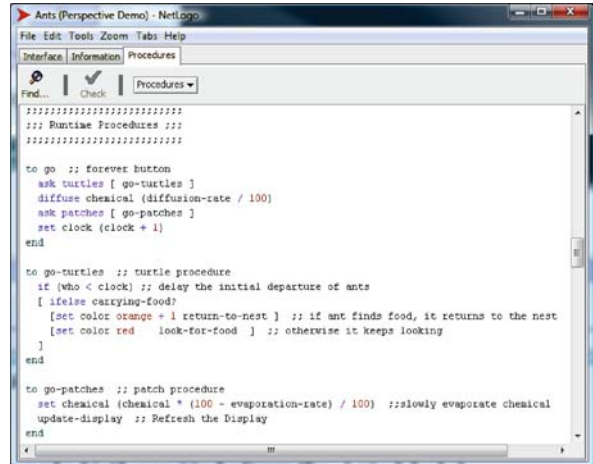
Agent Based Modeling and Computer Languages, Figure 5
Example NetLogo ant colony model [48]

implement a wider range of models. However, more work is often needed to build models in these systems. This article will discuss two selected examples, namely NetLogo and the visual interface for Repast Symphony.

NetLogo NetLogo is an education-focused ABM environment (Wilensky [48]). The NetLogo language uses a modified version of the Logo programming language (Harvey [19]). NetLogo itself is Java-based and is free for use in education and research. More information on NetLogo and downloads can be found at <http://ccl.northwestern.edu/netlogo/>.

NetLogo is designed to provide a basic computational laboratory for teaching complex adaptive systems concepts. NetLogo was originally developed to support teaching, but it can be used to develop a wider range of applications. NetLogo provides a graphical environment to create programs that control graphic ‘turtles’ that reside in a world of ‘patches’ that is monitored by an ‘observer’. NetLogo’s DSL is limited to its turtle and patch paradigm. However, NetLogo models can be extended using Java to provide for more general programming capabilities. An example NetLogo model of an ant colony [48] (center) feeding on three food sources (upper left corner, lower left corner, and middle right) is shown in Fig. 5. Example code [48] from this model is shown in Fig. 6.

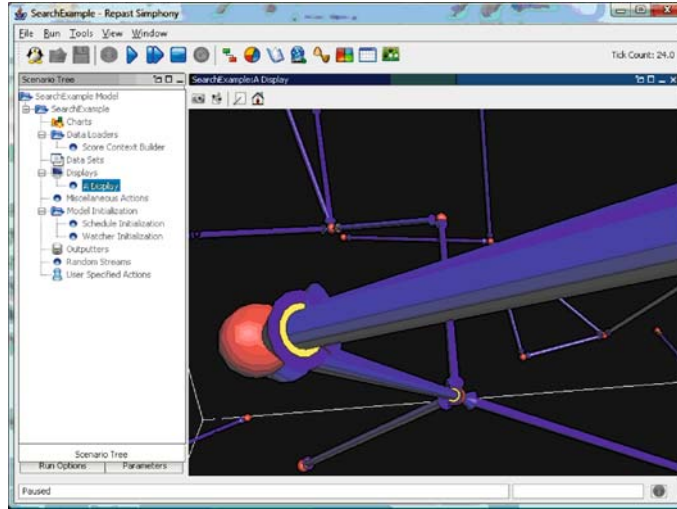
Repast Symphony Visual Interface The Recursive Porous Agent Simulation Toolkit (Repast) is a free and open



Agent Based Modeling and Computer Languages, Figure 6
Example NetLogo code from the ant colony model [48]

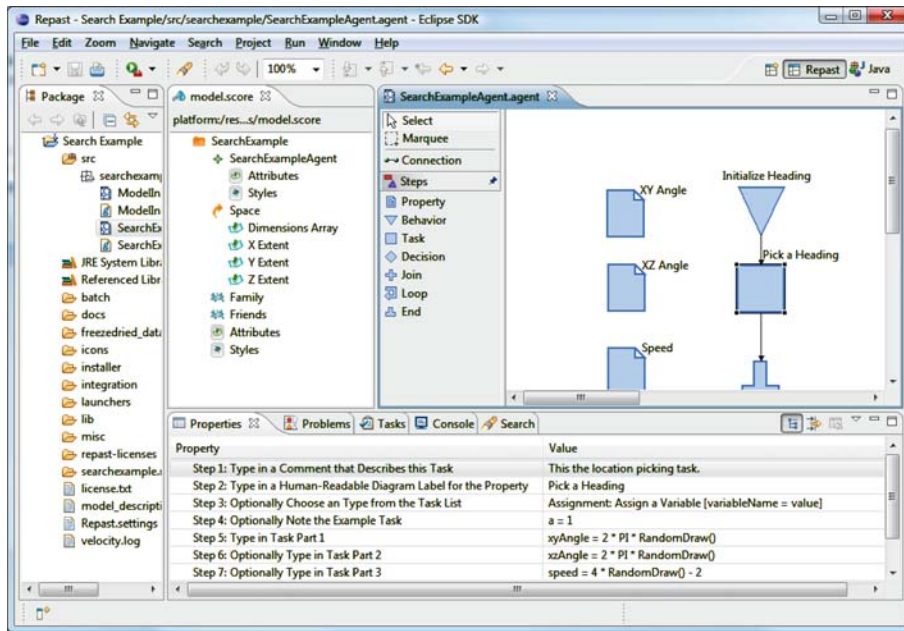
source family of agent-based modeling and simulation platforms (ROAD [44]). Information on Repast and free downloads can be found at <http://repast.sourceforge.net/>. Repast Symphony (Repast S) is the newest member of the Repast family [30,32]. The Java-based Repast S system includes advanced features for specifying, executing, and analyzing agent-based simulations. Repast Symphony offers several methods for specifying agents and agent environments including visual specification, specification with the dynamic object-oriented Groovy language [22], and specification with Java. In principle, Repast S’s visual DSL can be used for any kind of programming, but models beyond a certain level of complexity are better implemented in Groovy or Java. As discussed later, Groovy and Java are general purpose languages. All of Repast S’s languages can be fluidly combined in a single model. An example Repast S flocking model is shown in Fig. 7 [33]. The visual specification approach uses a tree to define model contents (Fig. 8 middle panel with “model.score” header) and a flowchart to define agent behaviors (Fig. 8 right side panel with “SearchExampleAgent.agent” header). In all cases, the user has a choice of a visual rich point-and-click interface or a ‘headless’ batch interface to execute models.

General Languages Unlike DSL’s, general languages are designed to take on any programming challenge. However, in order to meet this challenge they are usually more complex than DSL’s. This tends to make them more difficult to learn and use. Lahtinen et al. [23] documents some of the challenges users face in learning general purpose programming languages. Despite these issues, general purpose programming languages are essential for al-



Agent Based Modeling and Computer Languages, Figure 7

Example Repast Symphony flocking model [33]



Agent Based Modeling and Computer Languages, Figure 8

Example Repast Symphony visual behavior from the flocking model [33]

lowing users to access the full capabilities of modern computers. Naturally, there are a huge number of general purpose programming languages. This article considers these options from two perspectives. First, general language toolkits are discussed. These toolkits provide libraries of functions to be used in a general purpose host language. Second, the use of three raw general purpose languages, namely Java, C#, and C++, is discussed.

General Language Toolkits As previously stated, general language toolkits are libraries that are intended to be used in a general purpose host language. These toolkits usually provide model developers with software for functions such as simulation time scheduling, results visualization, results logging, and model execution as well as domain-specific tools [31]. Users of raw general purpose languages have to write all of the needed features by themselves by hand.

A wide range of general language toolkits currently exist. This article will discuss two selected examples, namely Swarm and the Groovy and Java interfaces for Repast Simphony.

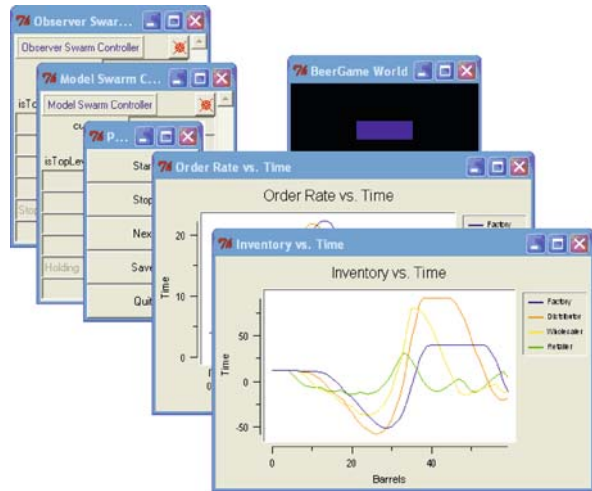
Swarm [28] is a free and open source agent-based modeling library. Swarm seeks to create a shared simulation platform for agent modeling and to facilitate the development of a wide range of models. Users build simulations by incorporating Swarm library components into their own programs. Information on Swarm and free downloads can be found at <http://www.swarm.org/> from Marcus Daniels [9]:

Swarm is a set of libraries that facilitate implementation of agent-based models. Swarm's inspiration comes from the field of Artificial Life. Artificial Life is an approach to studying biological systems that attempts to infer mechanism from biological phenomena, using the elaboration, refinement, and generalization of these mechanisms to identify unifying dynamical properties of biological systems ... To help fill this need, Chris Langton initiated the Swarm project in 1994 at the Santa Fe Institute. The first version was available by 1996, and since then it has evolved to serve not only researchers in biology, but also anthropology, computer science, defense, ecology, economics, geography, industry, and political science.

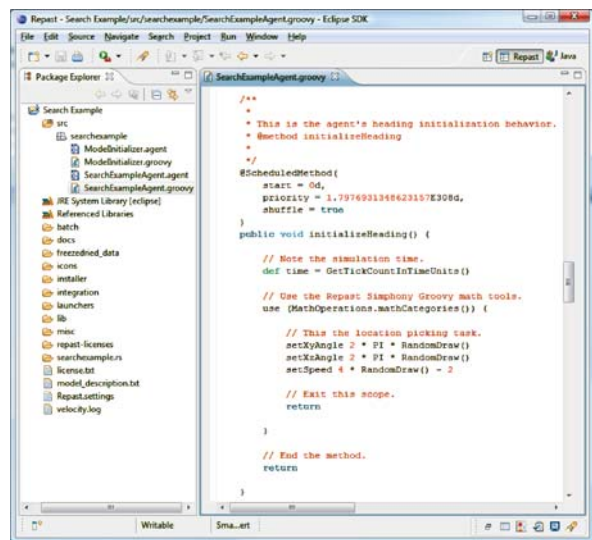
The Swarm simulation system has two fundamental components. The core component runs general-purpose simulation code written in Objective-C, Tcl/Tk, and Java. This component handles most of the behind the scenes details. The external wrapper components run user-specific simulation code written in either Objective-C or Java. These components handle most of the center stage work. An example Swarm supply chain model is shown in Fig. 9.

Repast Simphony Java and Groovy As previously discussed, Repast is a free and open source family of agent-based modeling and simulation platforms (ROAD [44]). Information on Repast and free downloads can be found at <http://repast.sourceforge.net/>. Repast S is the newest member of the Repast family [30,32]. The Java-based Repast S system includes advanced features for specifying, executing, and analyzing agent-based simulations. An example Repast S flocking model is shown in Fig. 7 [33].

Repast Simphony offers several intermixable methods for specifying agents and agent environments including visual specification, specification with the dynamic object-oriented Groovy language [22], and specification with



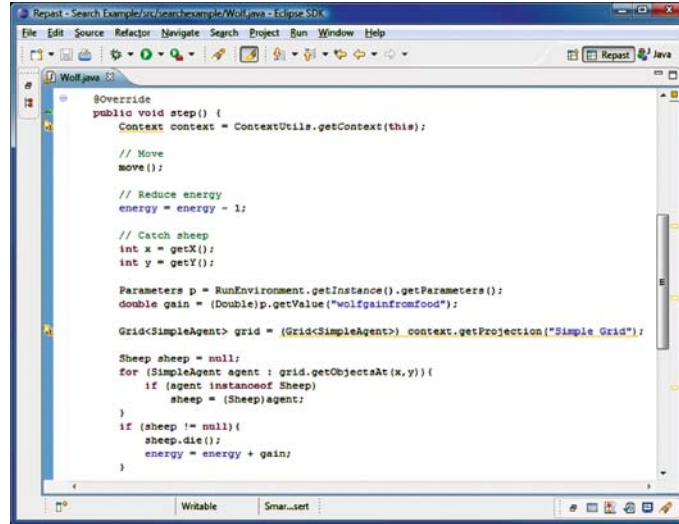
Agent Based Modeling and Computer Languages, Figure 9
Example Swarm supply chain model



Agent Based Modeling and Computer Languages, Figure 10
Example Repast Simphony Groovy code from the flocking model in Fig. 7 [33]

Java. The Groovy approach uses the dynamic object-oriented Groovy language as shown in Fig. 10. The Java approach for an example predator-prey model is shown in Fig. 11.

Java [12] is a widely used object-oriented programming language that was developed and is maintained by Sun Microsystems. Java is known for its widespread 'cross-platform' availability on many different types of hard-



Agent Based Modeling and Computer Languages, Figure 11
Example Repast Simphony Java code for a predator-prey model [41]

ware and operating systems. This capability comes from Java's use of a 'virtual machine' that allows binary code or 'bytecode' to have a consistent execution environment on many different computer platforms. A large number of tools are available for Java program development including the powerful Eclipse development environment [11] and many supporting libraries. Java uses reflection and dynamic method invocation to implement a variant of higher-order programming. Reflection is used for runtime class structure examination while dynamic method invocation is used to call newly referenced methods at runtime. Java's object-orientation, cross platform availability, reflection, and dynamic method invocation along with newer features such as annotations for including metadata in compiled code, generics for generalizing class, and aspects to implement dispersed but recurrent tasks make it a good choice for agent-based model development.

C# C# [1] is an object-oriented programming language that was developed and is maintained by Microsoft. C# is one of many languages that can be used to generate Microsoft .NET Framework code or Common Intermediate Language (CIL). Like Java bytecode, CIL is run using a 'virtual machine' that potentially gives it a consistent execution environment on different computer platforms. A growing number of tools are emerging to support C# development. C#, and the Microsoft .NET Framework more generally, are in principle cross platform, but in practice they are mainly executed under Microsoft Windows.

The Microsoft .NET Framework provides for the compilation into CIL of many different languages such as C#, Managed C++, and Managed Visual Basic to name just a few. Once these languages are compiled to CIL, the resulting modules are fully interoperable. This allows users to conveniently develop integrated software using a mixture of different languages. Like Java, C# supports reflection and dynamic method invocation for higher-order programming. C#'s object-orientation, multi-lingual integration, generics, attributes for including metadata in compiled code, aspects, reflection, and dynamic method invocation make it well suited for agent-based model development, particularly on the Microsoft Windows platform.

C++ C++ is a widely used object-oriented programming language that was created by Bjarne Stroustrup (Stroustrup [39]) at AT&T. C++ is widely noted for both its object-oriented structure and its ability to be easily compiled into native machine code. C++ gives users substantial access to the underlying computer but also requires substantial programming skills.

Most C++ compilers are actually more properly considered C/C++ compilers since they can compile non-object-oriented C code as well as object-oriented C++ code. This allows sophisticated users to the opportunity highly optimize selected areas of model code. However, this also opens the possibility of introducing difficult to resolve errors and hard to maintain code. It is also more difficult to port C++ code from one computer architecture to an-

other than it is for virtual machine-based languages such as Java.

C++ can use a combination of Runtime Type Identification (RTTI) and function pointers to implement higher-order programming. Similar to the Java approach, C++ RTTI can be used for runtime class structure examination while function pointers can be used to call newly referenced methods at runtime. C++'s object-orientation, RTTI, function pointers, and low-level machine access make it a reasonable choice for the development of extremely large or complicated agent-based models.

Future Directions

Future developments in computer languages could have enormous implications for the development of agent-based modeling. Some of the challenges of agent-based modeling for the future include (1) scaling up models to handle large numbers of agents running on distributed heterogeneous processors across the grid, (2) handling the large amounts of data generated by agent models and making sense out of it, and (3) developing user-friendly interfaces and modular components in a collaborative environment that can be used by domain experts with little or no knowledge of standard computer coding techniques. Visual and natural language development environments that can be used by non-programmers are continuing to advance but remain to be proven at reducing the programming burden. There are a variety of next steps for the development of computer languages for agent-based modeling including the further development of DSL's; increasing visual modeling capabilities; and the development of languages and language features that better support pattern-based development. DSL's are likely to become increasingly available as agent-based modeling grows into a wider range of domains. More agent-based modeling systems are developing visual interfaces for specifying model structures and agent behaviors. Many of these visual environments are themselves DSL's. The continued success of agent-based modeling will likely yield an increasing number of design patterns. Supporting and even automating implementations of these patterns may form a natural source for new language features. Many of these new features are likely to be implemented within DSL's.

Bibliography

1. Archer T (2001) *Inside C#*. Microsoft Press, Redmond
2. Backus J, Bauer F, Green J, Katz C, McCarthy J, Naur P, Perlis A, Rutishauser H, Samuelson K, Vauquois B, Wegstein J, van Wijngaarden A, Woodger M (1963) Revised Report on the Algorithmic Language ALGOL 60. In: Naur P (ed) *Communications of the ACM*, vol 6. ACM, New York, pp 1–17
3. Bhavnani R (2003) Adaptive agents, political institutions and civic traditions in modern Italy. *J Artif Soc Soc Simul* 6(4). Available at <http://jasss.soc.surrey.ac.uk/6/4/1.html>
4. Bonabeau E (2001) Agent-based modeling: methods and techniques for simulating human systems. *Proc Natl Acad Sci* 99(3):7280–7287
5. Casti J (1997) *Would-be worlds: how simulation is changing the world of science*. Wiley, New York
6. Coplien J (2001) Software patterns home page. Available as <http://hillside.net/patterns/>
7. Dahl O-J, Nygaard K (1966) SIMULA – an ALGOL-based simulation language. *Commun ACM* 9:671–678
8. Dahl O-J, Nygaard K (2001) How object-oriented programming started. Available at http://heim.ifi.uio.no/~kristen/FORSKNINGS/DOK_MAPPE/F_OO_start.html
9. Daniels M (1999) Integrating simulation technologies with swarm. In: *Proc of the agent 1999 workshop on agent simulation: applications, models, and tools*. Argonne National Laboratory, Argonne
10. Dijkstra E (1968) Go to statement considered harmful. *Commun ACM* 11(3):147–148
11. Eclipse (2008) Eclipse home page. Available at <http://www.eclipse.org/>
12. Foxwell H (1999) Java 2 software development kit. Linux J
13. Gamma E, Helm R, Johnson R, Vlissides J (1995) *Design patterns: elements of reusable object-oriented software*. Addison-Wesley, Wokingham
14. Gaylord R, D'Andria L (1998) *Simulating society: a mathematica toolkit for modeling socioeconomic behavior*. Springer/TELOS, New York
15. Gaylord R, Davis J (1999) Modeling nonspatial social interactions. *Math Edu Res* 8(2):1–4
16. Gaylord R, Nishidate K (1994) *Modeling nature: cellular automata simulations with Mathematica*. Springer, New York
17. Gaylord R, Wellin P (1995) *Computer simulations with Mathematica: explorations in complex physical and biological systems*. Springer/TELOS, New York
18. Guetzkow H, Kotler P, Schultz R (eds) (1972) *Simulation in social and administrative science*. Prentice Hall, Englewood Cliffs
19. Harvey B (1997) *Computer science logo style*. MIT Press, Boston
20. Heath J (2005) Methodological individualism. In: Zalta E (ed) *Stanford encyclopedia of philosophy*. Stanford University, Stanford, Available at <http://plato.stanford.edu>
21. Jennings N (2000) On agent-based software engineering. *Artif Intell* 117:277–296
22. Koenig D, Glover A, King P, Laforge G, Skeet J (2007) *Groovy in action*. Manning Publications, Greenwich
23. Lahtinen E, Ala-Mutka K, Jarvinen H-M (2005) A study of the difficulties of novice programmers. In: *Proc of the 10th annual SIGCSE conference on innovation and technology in computer science education*. Caparica, Portugal. ACM
24. McCarthy J (1960) Recursive functions of symbolic expressions and their computation by machine I. *J ACM* 3:184–195
25. Macal C (2004) Agent-based modeling and social simulation with Mathematica and MATLAB. In: Macal C, Sallach D, North M (eds) *Proc of the agent 2004 conference on social dynamics: interaction, reflexivity and emergence*. Argonne National Laboratory, Argonne
26. Macal C, Howe T (2005) Linking repast to computational mathematics systems: Mathematica and MATLAB. In: Macal C, Sallach D, North M (eds) *Proc of the agent 2005 conference*

- on generative social processes, models, and mechanisms. Argonne National Laboratory, Argonne
27. Macal C, North M (2007) Tutorial on Agent-based Modeling and Simulation: Desktop ABMS. In: Henderson SG, Biller B, Hsieh MH, Shortle J, Tew JD, Barton RR (eds) Proceedings of the 2007 Winter Simulation Conference, December 9-12, 2007, pp 95–106, <http://www.informs-sim.org/wsc07papers/011.pdf>
 28. Minar N, Burkhart R, Langton C, Askenazi M (1996) The swarm simulation system: a toolkit for building multi-agent simulations. Available at <http://alumni.media.mit.edu/~nelson/research/swarm/>
 29. North M, Macal C (2007) Managing business complexity: discovering strategic solutions with agent-based modeling and simulation. Oxford, New York
 30. North M, Howe T, Collier N, Vos J (2005) Repast symphony runtime system. In: Macal C, North M, Sallach D (eds) Proc of the agent 2005 conference on generative social processes, models, and mechanisms. Argonne National Laboratory, Argonne
 31. North M, Collier N, Vos R (2006) Experiences creating three implementations of the repast agent modeling toolkit. ACM Trans Model Comput Simul 16(1):1–25. ACM, New York
 32. North M, Tataru E, Collier N, Ozik J (2007) Visual agent-based model development with repast symphony. In: Macal C, North M, Sallach D (eds) Proc of the agent 2007 conference on complex interaction and social emergence. Argonne National Laboratory, Argonne
 33. North M, Howe T, Collier N, Tataru E, Ozik J, Macal C (2008) Search and emergence in agent-based models. In: Agent-based societies: Societal and cultural interactions. IGI Global Publishing, New York
 34. Pearson D, Boudarel M-R (2001) Pair interactions: real and perceived attitudes. J Artif Soc Soc Simul 4(4). Available at <http://www.soc.surrey.ac.uk/JASSS/4/4/4.html>
 35. Reynolds J (1998) Definitional Interpreters for Higher-Order Programming Languages. In: Higher-Order and Symbolic Computation. Kluwer, Boston, pp 363–397
 36. Sedgewick R (1988) Algorithms, 2nd edn. Addison-Wesley, Reading, pp 657
 37. Springer G, Freeman D (1989) Scheme and the art of programming. McGraw-Hill, New York
 38. Stevens W, Meyers G, Constantine L (1974) Structured design. IBM Syst J 2
 39. Stroustrup B (2008) Bjarne Stroustrup's FAQ. Available at http://www.research.att.com/~bs/bs_faq.html#invention
 40. Swarm Development Group (2008) SDG home page. Available at <http://www.swarm.org/>
 41. Tataru E, North M, Howe T, Collier N, Vos J (2006) An Introduction to Repast Symphony Modeling using a simple Predator-Prey Example. In: Proceedings of the Agent 2006 Conference on Social Agents: Results and Prospects. Argonne National Laboratory, Argonne
 42. Thorngate W (2000) Teaching social simulation with MATLAB. J Artif Soc Soc Simul 3(1). Available at <http://www.soc.surrey.ac.uk/JASSS/3/1/forum/1.html>
 43. Thoyer S, Morardet S, Rio P, Simon L, Goodhue R, Rausser G (2001) A bargaining model to simulate negotiations between water users. J Artif Soc Soc Simul 4(2). Available at <http://www.soc.surrey.ac.uk/JASSS/4/2/6.html>
 44. ROAD (2008) Repast home page. Available at <http://repast.sourceforge.net/>
 45. van Roy P, Haridi S (2004) Concepts, techniques, and models of computer programming. MIT Press, Cambridge
 46. Wang Z, Thorngate W (2003) Sentiment and social mitosis: implications of Heider's balance theory. J Artif Soc Soc Simul 6(3). Available at <http://jasss.soc.surrey.ac.uk/6/3/2.html>
 47. Watson D (1989) High-level languages and their compilers. Addison-Wesley, Wokingham
 48. Wilensky U (1999) NetLogo. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. <http://ccl.northwestern.edu/netlogo/>
 49. Wolfram Research (2008) Mathematica home page. Available at <http://www.wolfram.com/>

Agent Based Modeling, Large Scale Simulations

HAZEL R. PARRY

Central Science Laboratory, York, UK

Article Outline

Glossary

Definition of the Subject

Introduction

Large Scale Agent Based Models:

Guidelines for Development

Parallel Computing

Example

Future Directions

Acknowledgments

Bibliography

Glossary

Agent A popular definition of an agent, particularly in AI research, is that of Wooldridge [45], pp. 29: “an agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives”. In particular, it is the autonomy, flexibility, inter-agent communication, reactivity and proactiveness of the agents that distinguishes the paradigm and gives power to agent-based models and multi-agent simulation [15,21]. Multi-agent systems (MAS) comprise of numerous agents, which are given rules by which they act and interact with one another to achieve a set of goals.

Block mapping A method of partitioning an array of elements between nodes of a distributed system, where the array elements are partitioned as evenly as possible into blocks of consecutive elements and assigned to processors. The size of the blocks approximates to

the number of array elements divided by the number of processors.

Complexity Complexity and complex systems pertain to ideas of randomness and irregularity in a system, where individual-scale interactions may result in either very complex or surprisingly simple patterns of behavior at the larger scale. Complex agent-based systems are therefore usually made up of agents interacting in a non-linear fashion. The agents are capable of generating emergent behavioral patterns, of deciding between rules and of relying upon data across a variety of scales. The concept allows for studies of interaction between hierarchical levels rather than fixed levels of analysis.

Cyclic mapping A method of partitioning an array of elements between nodes of a distributed system, where the array elements are partitioned by cycling through each node and assigning individual elements of the array to each node in turn.

Grid Computer ‘Grids’ are comprised of a large number of disparate computers (often desktop PCs) that are treated as a virtual cluster when linked to one another via a distributed communication infrastructure (such as the internet or an intranet). Grids facilitate sharing of computing, application, data and storage resources. Grid computing crosses geographic and institutional boundaries, lacks central control, and is dynamic as nodes are added or removed in an uncoordinated manner. BOINC computing is a form of distributed computing is where idle time on CPUs may be used to process information (<http://boinc.berkeley.edu/>).

Ising-type model Ising-type models have been primarily used in the physical sciences. They simulate behavior in which individual elements (e. g., atoms, animals, social behavior, etc.) modify their behavior so as to conform to the behavior of other individuals in their vicinity. Conway’s Game of Life is a Ising-type model, where cells are in one of two states: dead or alive. In biology, the technique is used to model neural networks and flocking birds, for example.

Message passing (MP) Message passing (MP) is the principle way by which parallel clusters of machines are programmed. It is a widely-used, powerful and general method of enabling distribution and creating efficient programs [30]. Key advantages of using MP architectures are an ability to scale to many processors, flexibility, ‘future-proofing’ of programs and portability [29].

Message passing interface (MPI) A computing standard that is used for programming parallel systems. It is implemented as a library of code that may be used to en-

able message passing in a parallel computing system. Such libraries have largely been developed in C and Fortran, but are also used with other languages such as Java (MPIJava <http://www.hpjava.org>). It enables developers of parallel software to write parallel programs that are both portable and efficient.

Multiple instruction multiple data (MIMD)

Parallelization where different algorithms are applied to different data items on different processors.

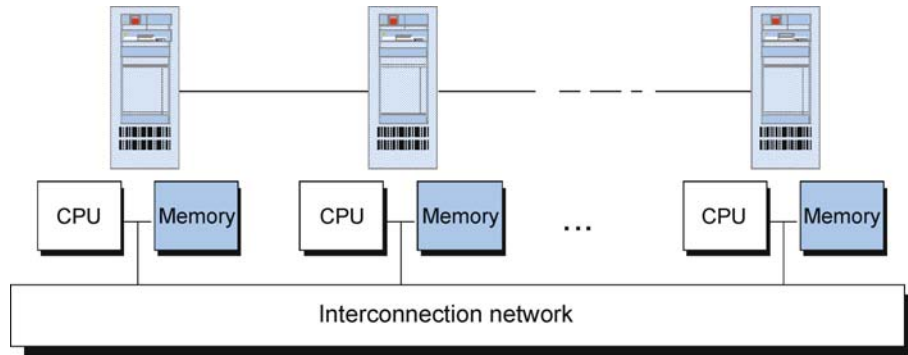
Parallel computer architecture A parallel computer architecture consists of a number of identical units that contain CPUs (Central Processing Units) which function as ordinary serial computers. These units, called nodes, are connected to one another (Fig. 1). They may transfer information and data between one another (e. g. via MPI) and simultaneously perform calculations on different data.

Single instruction multiple data (SIMD) SIMD techniques exploit data level parallelism: when a large mass of data of a uniform type needs the same instruction performed on it. An example is a vector or array processor. An application that may take advantage of SIMD is one where the same value is being added (or subtracted) to a large number of data points.

Vector computer/vector processor Vector computers contain a CPU designed to run mathematical operations on multiple data elements simultaneously (rather than sequentially). This form of processing is essentially a SIMD approach. The Cray Y-MP and the Convex C3880 are two examples of vector processors used for supercomputing in the 1980s and 1990s. Today, most recent commodity CPU designs include some vector processing instructions.

Definition of the Subject

‘Large scale’ simulations in the context of agent-based modelling are not only simulations that are large in terms of the size of the simulation (number of agents simulated), but they are also complex. Complexity is inherent in agent-based models, as they are usually composed of dynamic, heterogeneous, interacting agents. Large scale agent-based models have also been referred to as ‘Massively Multi-agent Systems (MMAS)’ [18]. MMAS is defined as “‘beyond resource limitation”: the number of agents exceeds local computer resources, or the situations are too complex to design/program given human cognitive resource limits’ [18], Preface. Therefore, for agent-based modelling ‘large scale’ is not simply a size problem, it is also a problem of managing complexity to ensure scalability of the agent model.



Agent Based Modeling, Large Scale Simulations, Figure 1

A network with interconnected separate memory and processors (after [30], pp. 19)

Multi-agent simulation models increase in scale as the modeller requires many agents to investigate whole system behavior, or the modeller wishes to fully examine the response of a single agent in a realistic context. Two key problems may be introduced by increasing the scale of a multi-agent system: (1) Computational resources limit the simulation time and/or data storage capacity and (2) Agent model analysis may become more difficult. Difficulty in analyzing the model may be due to the model system having a large number of complex components or to memory for model output storage being restricted by computer resources.

Introduction

Many systems that are now simulated using agent-based models are systems where agent numbers are large and potentially complex. These large scale models are constructed under a number of diverse scientific disciplines, with differing histories of agent simulation and methodologies emerging with which to deal with large scale simulations: for example molecular physics, social science (e. g. crowd simulation, city growth), telecommunications, ecology and military research. The primary methodology to emerge is parallel computing, where an agent model is distributed across a number of CPUs to increase the memory and processing power available to the simulation. However, there are a range of potential methods with which to simulate large numbers of agents. Some suggestions are listed in Table 1.

The simplest solution to enable larger scale agent simulation is usually to improve the computer hardware that is used and run the model on a server or invest in a more powerful PC. However, this option may be too costly or may not provide enough scaling so other options may then be considered. Another simple solutions may be to reduce

the number of agents or to revert to a simpler modelling approach such as a population model, but both of these solutions would significantly alter the model and the philosophy behind the model, probably not addressing the research question that the model was initially constructed for. There are a number of unique advantages and insights that may be gained from an agent-based approach. Agent simulations are often constructed to enable the analysis of emergent spatial patterns and individual life histories. The realism of an agent-based approach may be lost by using a simpler modelling technique.

The structure of agent simulations (often with asynchronous updating and heterogeneous data types) would mean that running a simulation on a vector computer may make little difference to the simulation performance. This is because an agent model typically has few elements that could take advantage of SIMD: rarely the same value will be added (or subtracted) to a large number of data points. Vector processors are less successful when a program does not have a regular structure, and they don't scale to arbitrarily large problems (the upper limit on the speed of a vector program will be some multiple of the speed of the CPU [30]).

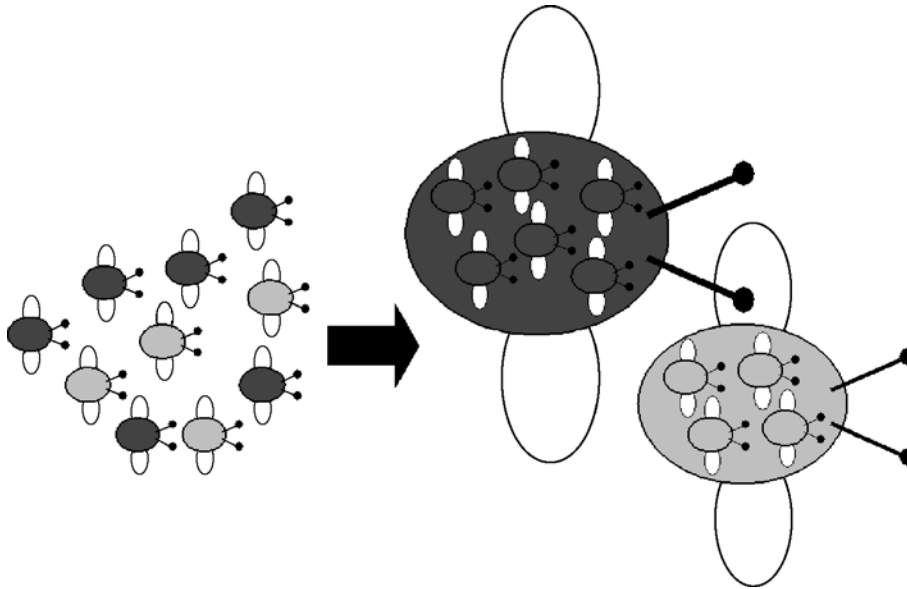
Another relatively simple option is to implement an aggregation of the individual agents into 'super-agents', such as the 'super-individual' approach of Scheffer et al. [35]. The basic concept of this approach is shown in Fig. 2. These 'super-agents' are formed from individual agents that share the same characteristics, such as age and sex. However, it may not be possible to group agents in a simulation in this way and, importantly, this method has been proven ineffective in a spatial context [31,32].

The most challenging solution, to reprogram the model in parallel, is a popular solution due to the shortcomings of the other approaches outlined above. A parallel solution may also have some monetary cost and require

Agent Based Modeling, Large Scale Simulations, Table 1

Potential solutions to implement when faced with a large number of agents to model

Solution	Pro	Con
Reduce the number of agents in order for model to run	No reprogramming of model	Unrealistic population. Alters model behavior
Revert to a population based modelling approach	Could potentially handle any number of individuals	Lose insights from agent approach. Unsuitable for research questions. Construction of entirely new model (non-agent-based)
Invest in an extremely powerful computer	No reprogramming of model	High cost
Run the model on a vector computer	Potentially more efficient as more calculations may be performed in a given time	This approach only works more efficiently with SIMD, probably unsuitable for agent-based models
Super-individuals [35]	Relatively simple solution, keeping model formulation the same	Reprogramming of model. Inappropriate in a spatial context [31,32]
Invest in a powerful computer network and reprogram the model in parallel	Makes available high levels of memory and processing power	High cost. Advanced computing skills required for restructuring of model



Agent Based Modeling, Large Scale Simulations, Figure 2

'Super-agents': Grouping of individuals into single objects that represent the collective

advanced computing skills to implement, but it can potentially greatly increase the scale of the agent simulation. Extremely large scale object-oriented simulations that simulate individual particles on massively parallel computer systems have been successfully developed in the physical sciences of fluid dynamics, meteorology and materials science. In the early 1990s, work in the field of molecular-dynamics (MD) simulations proved parallel platforms to be highly successful in enabling large-scale MD simulation of up to 131 million particles [26]. Today the same

code has been tested and used to simulate up to 320 billion atoms on the BlueGene/L architecture containing 131,072 IBM PowerPC440 processors [22]. These simulations include calculations based upon the short-range interaction between the individual atoms, thus in some ways approximate to agent simulations although in other ways molecular-dynamic simulations lack the complexity of most agent-based models (see Table 2).

There are significant decisions to be made when considering the application of a computing solution such as

Agent Based Modeling, Large Scale Simulations, Table 2

Key elements of a 'bottom-up' simulation that may affect the way in which it may scale. Agent simulations tend to be complex (to the right of the table), though may have some elements that are less complex, such as local or fixed interactions

Element	Least complex	→	Most complex
Spatial structure	Aspatial or Lattice of cells (1d, 2d or 3d +)		Continuous space
Internal state	Simple representation (boolean true or false)		Complex representation (many states from an enumerable set) or fuzzy variable values
Agent heterogeneity	No		Yes
Interactions	Local and fixed (within a neighborhood)		Multiple different ranges and stochastic
Synchrony of model updates	Synchronous update		Not synchronous: asynchrony due to state-transition rules

parallel programming to solve the problem of large numbers of agents. In addition to the issue of reprogramming the model to run on a parallel computer architecture, it is also necessary to consider the additional complexity of agents (as opposed to atoms), so that programming models and tools facilitate the deployment, management and control of agents in the distributed simulation [10]. For example, distributed execution resources and timelines must be managed, full encapsulation of agents must be enforced, and tight control over message-based multi-agent interactions is necessary [10]. Agent models can vary in complexity, but most tend to be complex especially in the key model elements of spatial structure and agent heterogeneity. Table 2 gives an indication of the relative complexity of model elements found in models that focus on individual interactions (which encompasses both multi-agent models and less complex, 'Ising'-type models).

The following sections detail guidelines for the development of a large scale agent-based model, highlighting in particular the challenges faced in writing large scale, high performance Agent based Modelling (ABM) simulations and giving a suggested development protocol. Following this, an example is given of the parallelization of a simple agent-based model, showing some of the advantages but also the pitfalls of this most popular solution. Key challenges, including difficulties that may arise in the analysis of agent-based models at a large scale, are highlighted. Alternative solutions are then discussed and some conclusions are drawn on the way in which large scale agent-based simulation may develop in coming years.

Large Scale Agent Based Models: Guidelines for Development

Key Considerations

There is no such thing as a standard agent-based model, or even a coherent methodology for agent simulation development (although recent literature in a number of fields sets out some design protocols, e.g. Gilbert [11] and

Grimm et al. [12]). Thus, there can be no standard method to develop a large scale agent-based model. However, there are certain things to consider when planning to scale up a model. Some key questions to ask about the model are as follows:

1. What program design do you already have and what is the limitation of this design?
 - (a) What is it the memory footprint for any existing implementation?
 - (b) What are your current run times?
2. What are your scaling requirements?
 - (a) How much do you need to scale now?
 - (b) How far do you need to scale eventually?
 - (c) How soon do you need to do it?
3. How simple is your model and how is it structured?
4. What are your agent complexities?
5. What are your output requirements?

The first question is to identify the limitations in the program design that you are using and to focus on the primary 'bottlenecks' in the model. These limitations will either be due to memory or speed (or perhaps both). Therefore it will be necessary to identify the memory footprint for your existing model, and analyze run times, identifying where the most time is taken or memory used by the simulation. It is primarily processor power that controls the speed of the simulation. Runtime will also increase massively once Random Access Memory (RAM) is used up, as most operating systems will resort to virtual memory (i.e. hard drive space), thus a slower mechanism with mechanical parts rather than solid-state technology engages. At this stage, it may be such that simple adjustments to the code may improve the scalability of the model. However, if the code is efficient, other solutions will then need to be sought.

The second question is how much scaling is actually necessary for the model. It may be such that a simple or interim solution (e.g. upgrading computer hardware) may be acceptable whilst only moderate scaling is required,

but longer term requirements should also be considered – a hardware upgrade may be a quick fix but if the model may eventually be used for much larger simulations it is necessary to plan for the largest scaling that will potentially be required.

The third question, relating to model simplicity and structure, is key to deciding a methodology that can be used to scale a model up. A number of factors will affect whether a model will be easy to distribute in parallel, for example. These include whether the model iterates at each time step or is event driven, whether it is aspatial or spatial and the level/type of agent interaction (both with one another and with the environment). More detail on the implications of these factors is given in Sect. “Parallel Computing”.

Agent complexity, in addition to model structure, may limit the options available for scaling up a model. For example, a possible scaling solution may be to group individual agents together as ‘super-individuals’ [35]. However, if agents are too complex it may not be possible to determine a simple grouping system (such as by age), as agent behavior may be influenced heavily by numerous other state variables.

Output requirements are also important to consider. These may already be limiting the model, in terms of memory for data storage. Even if they are not currently limiting the model in this way, once the model is scaled up output data storage needs may be an issue, for example, if the histories of individual agents need to be stored. In addition, the way that output data is handled by the model may be altered if the model structure is altered (e.g. if agents are grouped together output will be at an aggregate level). Thus, an important consideration is to ensure that output data is comparable to the original model and that it is feasible to output once the model structure is altered.

A Protocol

In relation to the key considerations highlighted above, a simple protocol for developing a large scale agent-based simulation can be defined as follows:

1. Optimize existing code.
2. Clearly identify scaling requirements (both for now and in the future).
3. Consider simple solutions first (e.g. a hardware upgrade).
4. Consider more challenging solutions.
5. Evaluate the suitability of the chosen scaling solution on a simplified version of the model before implementing on the full model.

The main scaling solution to implement (e.g. from Table 1) is defined by the requirements of the model. Implementation of more challenging solutions should be done in stages, where perhaps a simplified version of the model is implemented on a larger scale. Agent simulation development should originate with a local, flexible ‘prototype’ and then as the model development progresses and stabilizes larger scale implementations can be experimented with [10]. This is necessary for a parallel implementation of a model, for example, as a simplified model enables and assessment of whether it is likely to provide the desired improvements in model efficiency. This is particularly the case for improvements in model speed, as this depends on improved processing performance that is not easily calculated in advance.

Parallel Computing

Increasing the capacity of an individual computer in terms of memory and processing power has limited ability to perform large scale agent simulations, particularly due to the time the machine would take to run the model using a single processor. However, by using multiple processors and a mix of distributed and shared memory working simultaneously, the scale of the problem for each individual computer is much reduced. Subsequently, simulations can run in a fraction of the time that would be taken to perform the same complex, memory intensive, operations. This is the essence of parallel computing. ‘Parallel computing’ encompasses a wide range of computer architectures, from a HPC (High performance computing) Linux box, to dedicated multi-processor/multi-core systems (such as a Beowulf cluster), super clusters, local computer clusters or Grids and public computing facilities (e.g. Grid computers, such as the White Rose Grid, UK <http://www.wrgrid.org.uk/>). The common factor is that these systems consist of a number of interconnected ‘nodes’ (processing units), that may perform simultaneous calculations on different data. These calculations may be the same or different, depending whether a ‘Single Instruction Multiple Data’ (SIMD) or ‘Multiple Instruction Multiple data’ (MIMD) approach is implemented.

In terms of MAS, parallel computing has been used to develop large scale agent simulations in a number of disciplines. These range from ecology, e.g. [1,17,27,40,41,42,43] and biology, e.g. [5,7] to social science, e.g. [38] and computer science, e.g. [34], including artificial intelligence and robotics, e.g. [3,4].

Several key challenges arise when implementing an agent model in parallel, which may affect the increase in performance achieved. These include load balancing be-

tween nodes, synchronizing events to ensure causality, monitoring of the distributed simulation state, managing communication between nodes and dynamic resource allocation [39]. Good load balancing and inter-node communication with event synchronisation are central to the development of an efficient parallel simulation, and are further discussed below.

Load Balancing

In order to ensure the most efficient use of memory and processing resources in a parallel computing system the data load must be balanced between processors and the work load equally distributed. If this is not the case then one computer may be idle as others are working, resulting in time delays and inefficient use of the system's capacity. There are a number of ways in which data can be 'mapped' to different nodes and the most appropriate depends on the model structure. Further details and examples are given in Pacheco [30], including 'block mapping' and 'cyclic mapping'. An example of 'block mapping' load balancing is given below, in Sect. "Example".

In many simulations the computational demands on the nodes may alter over time, as the intensity of the agents' or environment's processing requirements varies on each node over time. In this case, dynamic load balancing techniques can be adopted to further improve the parallel model performance. For example, Jang [19] and Jang and Agha [20], use a form of dynamic load balancing with object migration they term "Adaptive Actor Architecture". Each agent platform monitors the workload of its computer node and the communication patterns of agents executing on it in order to redistribute agents according to their communication localities as agent platforms become overloaded. However, this approach does introduce additional processing overheads, so is only worth implementing for large scale agent simulations where some agents communicate with one another more intensely than other agents (communication locality is important) or communication patterns are continuously changing so static agent allocation is not efficient.

Communication Between Nodes

It is important to minimize inter-node communication when constructing a parallel agent simulation, as this may slow the simulation down significantly if the programmer is not careful [37,38]. The structure of the model itself largely determines the way in which data should be split and information transferred between nodes to maximize efficiency. Agent simulations generally by definition act spatially within an environment. Thus, an important first

consideration is whether to split the agents or the environment between nodes. The decision as to whether to split the agents between processors or elements of the environment such as grid cells largely depends upon the complexity of the environment, the mobility of the agents, and the number of interactions between the agents. If the environment is relatively simple (thus information on the whole environment may be stored on all nodes), it is probably most efficient to distribute the agents. This is particularly the case if the agents are highly mobile, as a key problem when dividing the environment between processors is the transfer of agents or information between processors. However, if there are complex, spatially defined interactions between agents, splitting agents between nodes may be problematic, as agents may be interacting with other agents that are spatially local in the context of the whole simulation but are residing on different processors. Therefore conversely, if the agents are not very mobile but have complex, local interactions and/or the agents reside in a complex environment, it is probably best to split the environment between nodes [25]. Further efficiency may be achieved by clustering the agents which communicate heavily with each other [37].

In models where there is high mobility and high interaction it is often possible, especially for ecological models, to find a statistical commonality that can be used as a replacement for more detailed interaction. For example, as will be shown in our example, if the number of local agent interactions is the only important aspect of the interactions, a density map of the agents, transferred to a central node, aggregated and redistributed, might allow agents to be divided between nodes without the issue of having to do detailed inter-agent communication between nodes a large number of times.

The way in which the simulation iterates may influence the approach taken when parallelizing the model. The model may update synchronously at a given time step or asynchronously (usually because the system is event-driven). In addition, agents may update asynchronously but the nodes may be synchronized at each time step or key model stage. Asynchronous updating may be a problem if there is communication between nodes, as some nodes may have to wait for others to finish processes before communication takes place and further processing is possible, resulting in blocking (see below). Communication between nodes then becomes highly complex [44]. It is important that messages communicating between agents are received in the correct order, however a common problem in distributed simulations is ensuring that this is so as other factors, such as latency in message transmission across the network, may affect communication [44].

A number of time management mechanisms exist that may be implemented to manage message passing in order to ensure effective node to node communication, e. g. [8].

Blocking and Deadlocking

Deadlock occurs when two or more processes are waiting for communication from one of the other processes. When programming a parallel simulation it is important to avoid deadlock to ensure the simulation completes. The simplest example is when two processors are programmed to receive from the other processor before that processor has sent. This may be simply resolved by changing the order that tasks are executed, or to use ‘non-blocking’ message passing. Where blocking is used, processing on nodes waits until a message is transmitted. However, when ‘non-blocking’ is used, processing continues even if the message hasn’t been transmitted yet. The use of a non-blocking MPI may reduce computing times, and work can be performed while communication is in progress.

Example

To demonstrate some of the benefits and pitfalls of parallel programming for a large scale agent-based model, a simple example is given here. This summarizes a simplified agent-based model of aphid population dynamics in agricultural landscapes of the UK, which was parallelized to cope with millions of agents, as described in detail in Parry [31], Parry and Evans [32] and Parry et al. [33].

A key problem with the original, non-parallel, aphid simulation was that it was hindered by memory requirements, which were far larger than could be accommodated at any individual processing element. This is a common computing problem [6]. The data storage required for each aphid object in a landscape scale simulation quickly exceeded the storage capacity of a PC with up to 2097 MB of RAM. The combined or ‘virtual shared’ memory of several computers was used to cope with the amount of data needed, using a Single Instruction Multiple-Data approach (SIMD).

Message-passing techniques were used to transfer information between processors, to distribute the agents in the simulation across a Beowulf cluster (a 30-node distributed memory parallel computer). A Message-passing Interface (MPI) for Java was used, MPIJava (<http://www.hpjava.org>). ‘MPIJava wraps around the open-source’ open-source native MPI ‘LAM’ (<http://www.lam-mpi.org/>). Further details on the methods used to incorporate the MPI into the model are given in Parry [31] and Parry et al. [33].

Effective parallelization minimizes the passing of information between nodes, as it is processor intensive. In the example model, only the environment object and information on the number of agents to create on each node are passed from a single control node to each of the other nodes in the cluster, and only density information is returned to the control node for redistribution and display. The control node manages the progress of the model, acts as a central communication point for the model and handles any code that may not be distributed to all nodes (such as libraries from an agent toolkit or a GUI). Structuring a model without a control node is possible, or the control node may also be used to process data, depending on the requirements of the simulation. Transfer of density values, rather than agents, significantly reduced the computational overheads for message passing between the nodes. The model was simple enough that specific inter-agent communication between nodes was not necessary.

Even distribution of data between nodes was achieved by splitting immigrant agents evenly across the system, with each node containing information on the environment and local densities passed from the control node. The number of immigrants to be added to each node was calculated by a form of ‘block mapping’, pp. 35 in [30], which partitioned the number of immigrants into blocks which were then assigned to each node. So, if there were three nodes ($n = 3$) and thirteen immigrants ($i = 13$), the immigrants mapped to each node would be as follows:

$$i_0, i_1, i_2, i_3 \rightarrow n_1$$

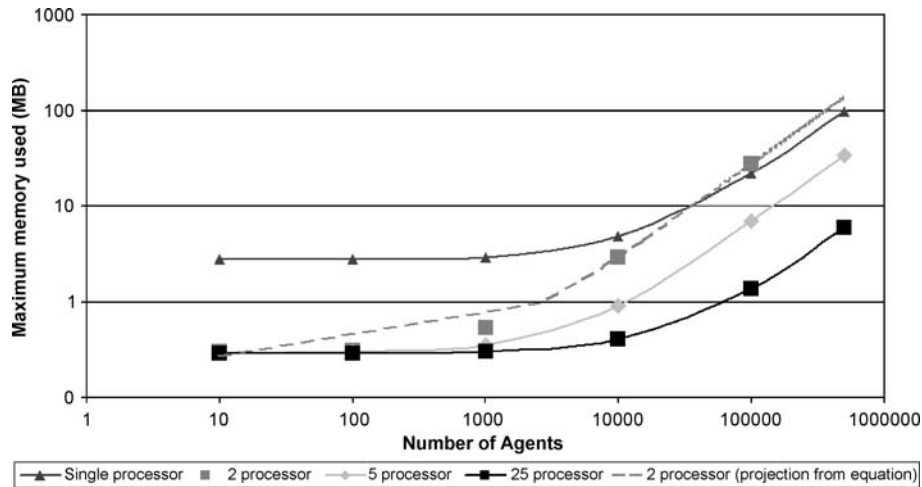
$$i_4, i_5, i_6, i_7 \rightarrow n_2$$

$$i_8, i_9, i_{10}, i_{11}, i_{12} \rightarrow n_3.$$

As thirteen does not divide evenly by three, the thirteenth agent is added to the final node.

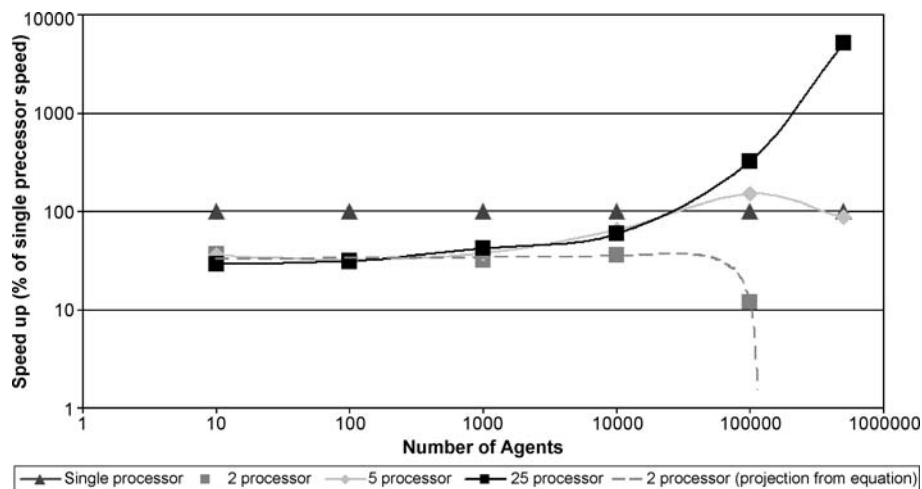
Benefits

Simulation runtime and memory availability was greatly improved by implementing the simple aphid model in parallel across a large number of nodes. The greatest improvement in simulation runtime and memory availability was seen when the simulation was run across the maximum number of nodes (25) (Figs. 3 and 4). The largest improvement in speed given by the parallel model in comparison to the non-parallel model is when more than 500,000 agents are run across twenty-five nodes, although the parallel model is slower by comparison for lower numbers. This means that additional processing power is required in the parallel simulation compared to the original model, such that only when very large numbers of agents are run does it become more efficient.



Agent Based Modeling, Large Scale Simulations, Figure 3

Plot of the mean maximum memory used (per node) against number of agents for the model: comparison between simulations using 2, 5 and 25 nodes and the non-parallel model (single processor)



Agent Based Modeling, Large Scale Simulations, Figure 4

Plot of the percentage speed up (per node) from the non-parallel model against number of agents for the model: comparison between simulations using 2, 5 and 25 nodes and the non-parallel model (single processor)

Pitfalls

Although there are clear benefits of distributing the example simulation across a large number of nodes, the results highlight that the parallel approach is not always more efficient than the original single processor implementation of a model. In the example, the two node simulation used more memory on the worker node than the non-parallel model when the simulation had 100,000 agents or above. This is due to additional memory requirements introduced by message passing and extra calculations required in the parallel implementation (which are less sig-

nificant when more nodes are used as these requirements remain relatively constant).

The results also highlight that adding more processors does not necessarily increase the model speed. The example model shows that for simulations run on two nodes (one control node, one worker node) the simulation takes longer to run in parallel compared to the non-parallel model. Message passing time delay and the modified structure of the code are responsible. The greatest improvement in speed is when more than 500,000 agents are run across twenty-five nodes, however when lower numbers of nodes are used the relationship between the num-

ber of nodes and speed is complex: for 100,000 agents five nodes are faster than the non-parallel model, but for 500,000 the non-parallel model is faster. Overall, these results suggest that when memory is sufficient on a single processor, it is unlikely to ever be efficient to parallelize the code, as when the number of individuals was low the parallel simulation took longer and was less efficient than the non-parallel model run on a single node.

This demonstrates that in order to effectively parallelize an agent model, the balance between the advantage of increasing the memory availability and the cost of communication between nodes must be assessed. By following an iterative development process as suggested in Sect. “A Protocol”, the threshold below which parallelization is not efficient and whether this option is suitable for the model should become apparent in the early stages of model development. Here, the simplified study confirmed the value of further development to build a parallel version of the full model.

Future Directions

Complexity and Model Analysis

In addition to the processing and data handling issues faced by large scale agent-based simulations, as agent simulations increase in scale and complexity, model analysis may become more difficult and the system may become intractable. There is no clearly defined way of dealing with increased difficulties of model analysis introduced by the greater complexity of large scale agent-based models, in the same way that there is no clear way to deal with the complexity inherent in most agent-based systems. However, some guidelines to model analysis have recently been developed for agent simulations, e.g. [12], and some suggestions are put forward on ways in which agent simulation complexity may be described, for example by the use of a model ‘overview’, ‘design concepts’ and ‘details’ (ODD) protocol for agent model description. In particular, the protocol requires that design concepts are linked to general concepts identified in the field of Complex Adaptive Systems [13], including emergence and stochasticity.

To address issues of complexity in a large scale agent simulation, one possibility is to include an additional layer of structuring, ‘organizational design’, into the system. This is where the peer group of an agent, its roles and responsibilities are assigned in the model and made explicit, pp. 121 in [16]. In general agent simulations already have an implicit organization structure; Horling and Lesser [16] argue that explicit organizational design highlights hidden inefficiencies in the model and allows the model to take full advantages of the resources available.

Grid Computing

For large scale agent-based models, many researchers have reached the limitations of ordinary PCs. However, there are ‘critical tensions’ [10] between agent simulations built on ordinary PCs and heterogeneous, distributed parallel programming approaches. The architecture of a distributed system is very different to that of an ordinary PC, thus to transfer a simulation to a computer cluster additional system properties must be taken into account, including management of the distribution of the simulation and concurrency [10]. This is particularly apparent when parallelization is attempted on heterogeneous, non-dedicated systems such as a public Grid. The Grid may offer a number of advantages for large scale agent-based simulation, such as collaboration between modellers, access to resources and geographically distributed datasets [47]. However, in such systems, issues of infrastructure reliability, functional completeness and the state of documentation for some kinds of environments exist [10]. In order to use such a system the ‘fundamental’ issue of partial failures must be addressed (e.g. with a dynamic agent replication strategy [14]).

Dissemination of Techniques

For parallel computing as a solution to large scale agent-based simulation, there is an interesting and useful future challenge to develop user friendly, high performance, versatile hardware architectures and software systems. Many developers of agent simulations are not computer scientists by training, and still rely upon numerous agent toolkits for simulation development (e.g. Swarm and Repast). Automatic distribution of agents to whatever resources are available would be a great tool for many agent software developers. Therefore, perhaps the greatest challenge would be to develop a system that would allow for parallelisation to be performed in an agent simulation automatically, where agents may be written in a high-level language and could be automatically partitioned to nodes in a network. One example of an attempt to achieve this is Graphcode (<http://parallel.hpc.unsw.edu.au/rks/graphcode/>). Based upon MPI, it maps agent-based models onto parallel computers, where agents are written based upon their graph topography to minimize communication overhead. Another example is HLA_GRID_Repast [47], ‘a system for executing large scale distributed simulations of agent-based systems over the Grid’, for users of the popular Repast agent toolkit. HLA_GRID_Repast is a middleware layer which enables the execution of a federation of multiple interacting instances of Repast models across a grid network with a High

Level Architecture (HLA). This is a ‘centralized coordination approach’ to distributing an agent simulation across a network [39]. Examples of algorithms designed to enable dynamic distribution of agent simulations are given in Scheutz and Schermerhorn [36].

Although parallel computing is often the most effective way of handling large scale agent-based simulations, there are still some significant obstacles to the use of parallel computing for MAS. As shown with the simple example given here, this may not always be the most effective solution depending upon the increase in scale needed and the model complexity. Other possible methods were suggested in Sect. “Introduction”, but these may also be unsuitable. Another option could be to deconstruct the model and simplify only certain elements of the model using either parallel computing or one of the other solutions suggested in Sect. “Introduction”. Such a ‘hybrid’ approach is demonstrated by Zhang and Lui [46], who combine equation-based approaches and multi-agent simulation with a Cellular Automata to simulate the complex interactions in the process of human immune response to HIV. The result is a model where equations are used to represent within-site processes of HIV infection, and agent-based simulation is used to represent the diffusion of the virus between sites.

It is therefore important to consider primarily the various ways in which the model may be altered, hybridized or simplified yet still address the core research questions, before investing money in hardware or investing time in the development of complex computational solutions. Making the transition from a serial application to a parallel version is a process that requires a fair degree of formalism and program restructuring, so is not to be entered into lightly without exploring the other options and the needs of the simulation first.

Overall, it is clear that disparate work is being done in a number of disciplines to facilitate large scale agent-based simulation, and knowledge is developing rapidly. Some of this work is innovative and highly advanced, yet inaccessible to researchers in other disciplines who may be unaware of key developments outside of their field. This chapter synthesizes and evaluates large scale agent simulation to date, providing a reference for a wide range of agent simulation developers.

Acknowledgments

Many thanks to Andrew Evans (Multi-Agent Systems and Simulation Research Group, University of Leeds, UK) and Phil Northing (Central Science Laboratory, UK) for their advice on this chapter.

Bibliography

Primary Literature

1. Abbott CA, Berry MW, Comiskey EJ, Gross LJ, Luh H-K (1997) Parallel Individual-Based Modeling of Everglades Deer Ecology. *IEEE Comput Sci Eng* 4(4):60–78
2. Anderson J (2000) A Generic Distributed Simulation System for Intelligent Agent Design and Evaluation. In: *Proceedings of the AI, Simulation & Planning In High Autonomy Systems*, Arizona
3. Bokma A, Slade A, Kerridge S, Johnson K (1994) Engineering large-scale agent-based systems with consensus. *Robot Comput-Integr Manuf* 11(2):81–91
4. Bouzid M, Chevrier V, Vialle S, Charpillat F (2001) Parallel simulation of a stochastic agent/environment interaction model. *Integr Comput-Aided Eng* 8(3):189–203
5. Castiglione F, Bernaschi M, Succi S (1997) Simulating the Immune Response on a Distributed Parallel Computer. *Int J Mod Phys C* 8(3):527–545
6. Chalmers A, Tidmus J (1996) *Practical Parallel Processing: An Introduction to Problem Solving in Parallel*. International Thomson Computer Press, London
7. Da-Jun T, Tang F, Lee TA, Sarda D, Krishnan A, Goryachev A (2004) Parallel computing platform for the agent-based modeling of multicellular biological systems. In: *Parallel and Distributed Computing: Applications and Technologies. Lecture Notes in Computer Science*, vol 3320, pp 5–8
8. Fujimoto RM (1998) Time management in the high level architecture. *Simul* 71:388–400
9. Gasser L, Kakugawa K (2002) MACE3J: Fast flexible distributed simulation of large, large-grain multi-agent systems. In: *Proceedings of AAMAS*
10. Gasser L, Kakugawa K, Chee B, Esteva M (2005) Smooth scaling ahead: progressive MAS simulation from single PCs to Grids. In: Davidsson P, Logan B, Takadama K (eds) *Multi-agent and multi-agent-based simulation. Joint Workshop MABS 2004* New York, NY, USA, July 19, 2004. Springer, Berlin
11. Gilbert N (2007) *Agent-based Models (Quantitative applications in the social sciences)*. SAGE, London
12. Grimm V, Berger U, Bastiansen F, Eliassen S, Ginot V, Giske J, Goss-Custard J, Grand T, Heinz S, Huse G, Huth A, Jepsen JU, Jorgensen C, Mooij WM, Muller B, Pe'er G, Piu C, Railsback SF, Robbins AM, Robbins MM, Rossmanith E, Ruger N, Strand E, Souissi S, Stillman RA, Vabo R, Visser U, DeAngelis DL (2006) A standard protocol for describing individual-based and agent-based models. *Ecol Model* 198(1–2):115–126
13. Grimm V, Railsback SF (2005) *Individual-based Modeling and Ecology. Princeton Series in Theoretical and Computational Biology*. Princeton University Press, Princeton, 480 pp
14. Guessoum Z, Briot J-P, Faci N (2005) Towards fault-tolerant massively multiagent system. In: Ishida T, Gasser L, Nakashima H (eds) *Massively Multi-Agent Systems I: First International Workshop MMAS 2004*, Kyoto, Japan, December 2004. Springer, Berlin
15. Heppenstall AJ (2004) *Application of Hybrid Intelligent Agents to Modelling a Dynamic, Locally Interacting Retail Market*. Ph D thesis, University of Leeds, UK
16. Horling B, Lesser V (2005) Quantitative organizational models for large-scale agent systems. In: Ishida T, Gasser L, Nakashima H (eds) *Massively Multi-Agent Systems I: First Inter-*

- national Workshop MMAS 2004, Kyoto, Japan, December 2004. Springer, Berlin
17. Immanuel A, Berry MW, Gross LJ, Palmer M, Wang D (2005) A parallel implementation of ALFISH: simulating hydrological compartmentalization effects on fish dynamics in the Florida Everglades. *Simul Model Pract Theory* 13:55–76
 18. Ishida T, Gasser L, Nakashima H (eds) (2005) *Massively Multi-Agent Systems I. First International Workshop, MMAS 2004, Kyoto, Japan*. Springer, Berlin
 19. Jang MW (2006) Agent framework services to reduce agent communication overhead in large-scale agent-based simulations. *Simul Model Pract Theory* 14(6):679–694
 20. Jang MW, Agha G (2005) Adaptive agent allocation for massively multi-agent applications. In: Ishida T, Gasser L, Nakashima H (eds) *Massively Multi-Agent Systems I: First International Workshop MMAS 2004, Kyoto, Japan, December 2004*. Springer, Berlin
 21. Jennings NR (2000) On agent-based software engineering. *Artif Intell* 117:277–296
 22. Kadau K, Germann TC, Lomdahl PS (2006) Molecular dynamics comes of age: 320 billion atom simulation on BlueGene/L. *Int J Mod Phys C* 17(12):1755
 23. Lees M, Logan B, Oguara T, Theodoropoulos G (2003) Simulating Agent-Based Systems with HLA: The case of SIM_AGENT – Part II. In: *Proceedings of the 2003 European Simulation Interoperability Workshop*
 24. Lees M, Logan B, Theodoropoulos G (2002) Simulating Agent-Based Systems with HLA: The case of SIM_AGENT. In: *Proceedings of the 2002 European Simulation Interoperability Workshop*, pp 285–293
 25. Logan B, Theodoropoulos G (2001) The Distributed Simulation of Multi-Agent Systems. *Proc IEEE* 89(2):174–185
 26. Lomdahl PS, Beazley DM, Tamayo P, Gronbechjensen N (1993) Multimillion particle molecular-dynamics on the CM-5. *Int J Mod Phys C: Phys Comput* 4(6):1075–1084
 27. Lorek H, Sonnenschein M (1995) Using parallel computers to simulate individual-oriented models in ecology: a case study. In: *Proceedings: ESM '95 European Simulation Multiconference*, Prague, June 1995
 28. Luke S, Cioffi-Revilla C, Panait L, Sullivan K (2004) MASON: A New Multi-Agent Simulation Toolkit. In: *Proceedings of the 2004 SwarmFest Workshop*
 29. Openshaw S, Turton I (2000) *High performance computing and the art of parallel programming: an introduction for geographers, social scientists, engineers*. Routledge, London
 30. Pacheco PS (1997) *Parallel Programming with MPI*. Morgan Kaufman Publishers, San Francisco
 31. Parry HR (2006) Effects of Land Management upon Species Population Dynamics: A Spatially Explicit, Individual-based Model. Ph D thesis, University of Leeds
 32. Parry HR, Evans AJ (in press) A comparative analysis of parallel processing and super-individual methods for improving the computational performance of a large individual-based model. *Ecological Modelling*
 33. Parry HR, Evans AJ, Heppenstall AJ (2006) Millions of Agents: Parallel Simulations with the Repast Agent-Based Toolkit. In: Trapp R (ed) *Cybernetics and Systems 2006, Proceedings of the 18th European Meeting on Cybernetics and Systems Research*
 34. Popov K, Vlassov V, Rafea M, Holmgren F, Brand P, Haridi S (2003) Parallel agent-based simulation on a cluster of workstations. In: *EURO-PAR 2003 Parallel Processing*, vol 2790, pp 470–480
 35. Scheffer M, Baveco JM, DeAngelis DL, Rose KA, van Nes EH (1995) Super-Individuals: a simple solution for modelling large populations on an individual basis. *Ecol Model* 80:161–170
 36. Scheutz M, Schermerhorn P (2006) Adaptive Algorithms for the Dynamic Distribution and Parallel Execution of Agent-Based Models. *J Parallel Distrib Comput* 66(8):1037–1051
 37. Takahashi T, Mizuta H (2006) Efficient agent-based simulation framework for multi-node supercomputers. In: Perrone LF, Wieland FP, Liu J, Lawson BG, Nicol DM, Fujimoto RM (eds) *Proceedings of the 2006 Winter Simulation Conference*
 38. Takeuchi I (2005) A massively multi-agent simulation system for disaster mitigation. In: Ishida T, Gasser L, Nakashima H (eds) *Massively Multi-Agent Systems I: First International Workshop MMAS 2004, Kyoto, Japan, December 2004*. Springer, Berlin
 39. Timm IJ, Pawlaszczyk D (2005) Large Scale Multiagent Simulation on the Grid. In: Veit D, Schnizler B, Eymann T (eds) *Proceedings of the Workshop on Agent-based grid Economics (AGE 2005) at the IEEE International Symposium on Cluster Computing and the Grid (CCGRID)*. Cardiff University, Cardiff
 40. Wang D, Berry MW, Carr EA, Gross LJ (2006a) A parallel fish landscape model for ecosystem modeling. *Simul* 82(7):451–465
 41. Wang D, Berry MW, Gross LJ (2006b) On parallelization of a spatially-explicit structured ecological model for integrated ecosystem simulation. *Int J High Perform Comput Appl* 20(4):571–581
 42. Wang D, Carr E, Gross LJ, Berry MW (2005a) Toward ecosystem modeling on computing grids. *Comput Sci Eng* 7:44–52
 43. Wang D, Gross L, Carr E, Berry M (2004) Design and implementation of a Parallel Fish Model for South Florida. In: *Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04)*
 44. Wang F, Turner SJ, Wang L (2005b) Agent Communication in Distributed Simulations. In: Davidsson P, Logan B, Takadama K (eds) *Multi-agent and multi-agent-based simulation. Joint Workshop MABS 2004 New York, NY, USA, July 19, 2004*. Springer, Berlin
 45. Wooldridge M (1999) *Intelligent agents*. In: Weiss G (ed) *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. MIT Press, Cambridge, pp 27–78
 46. Zhang S, Lui J (2005) A massively multi-agent system for discovering HIV-immune interaction dynamics. In: Ishida T, Gasser L, Nakashima H (eds) *Massively Multi-Agent Systems I: First International Workshop MMAS 2004, Kyoto, Japan, December 2004*. Springer, Berlin
 47. Zhang Y, Theodoropoulos G, Minson R, Turner SJ, Cai W, Xie Y, Logan B (2005) Grid-aware Large Scale Distributed Simulation of Agent-based Systems. In: *2004 European Simulation Interoperability Workshop (EuroSIW 2005), 05E-SIW-047, Toulouse, France*

Books and Reviews

Agent libraries and toolkits with distributed or parallel features:

- Distributed GenSim: Supports distributed parallel execution [2].
- Ecolab: <http://ecolab.sourceforge.net/>. EcoLab models may also use the Graphcode library to implement a distributed network of agents over an MPI-based computer cluster.

- Graphcode system <http://parallel.hpc.unsw.edu.au/rks/graphcode/>.
- MACE3J: <http://www.isrl.uiuc.edu/amag/mace/> an experimental agent platform supporting deployment of agent simulations across a variety of system architectures [9,10].
- MASON <http://cs.gmu.edu/~eclab/projects/mason/>. MASON was 'not intended to include parallelization of a single simulation across multiple networked processors' [28]. However, it does provide two kinds of simple parallelization:
 1. Any given step in the simulation can be broken into parallel sub-steps each performed simultaneously.
 2. A simulation step can run asynchronously in the background independent of the simulation.
- Repast <http://repast.sourceforge.net> and HLA_GRID_Repast [47]. The Repast toolkit has in-built capabilities for performing batch simulation runs.
- SimAgent: <http://www.cs.bham.ac.uk/research/projects/poplog/packages/simagent.html>. Two developments support distributed versions of SimAgent:
 1. The use of HLA to distribute SimAgent [23,24]
 2. The SWAGES package: <http://www.nd.edu/~airolab/software/index.html>. This allows SimAgent to be distributed over different computers and interfaced with other packages.

Message Passing Interfaces (MPI):

- Background and tutorials <http://www-unix.mcs.anl.gov/mpi/>
- MPICH2 <http://www-unix.mcs.anl.gov/mpi/mpich/>
- MPI forum <http://www.mpi-forum.org/>
- MPIJava <http://www.hpjava.org>
- OpenMP <http://www.openmp.org>
- OpenMPI <http://www.open-mpi.org/>

Parallel computing and distributed agent simulation websites:

- Further references and websites <http://www.cs.rit.edu/~ncs/parallel.html>.
- Introduction to parallel programming http://www.mhpcc.edu/training/workshop/parallel_intro/MAIN.html
- <http://www.agents.cs.nott.ac.uk/research/simulation/simulators/> (implementations of HLA_GRID_Repast and distributed SimAgent).
- Globus Grid computing resources <http://www.globus.org/>.
- Beowulf computer clusters <http://www.beowulf.org/>

Agent Based Modeling, Mathematical Formalism for

REINHARD LAUBENBACHER¹, ABDUL S. JARRAH¹,
HENNING S. MORTVEIT¹, S.S. RAVI²

¹ Virginia Bioinformatics Institute,
Virginia Polytechnic Institute and State University,
Virginia, USA

² Department of Computer Science,
University at Albany – State University of New York,
New York, USA

Article Outline

Glossary

Definition of the Subject

Introduction

Existing Mathematical Frameworks

Finite Dynamical Systems

Finite Dynamical Systems

as Theoretical and Computational Tools

Mathematical Results on Finite Dynamical Systems

Future Directions

Bibliography

Glossary

Agent-based simulation An agent-based simulation of a complex system is a computer model that consists of a collection of agents/variables that can take on a typically finite collection of states. The state of an agent at a given point in time is determined through a collection of rules that describe the agent's interaction with other agents. These rules may be deterministic or stochastic. The agent's state depends on the agent's previous state and the state of a collection of other agents with whom it interacts.

Mathematical framework A mathematical framework for agent-based simulation consists of a collection of mathematical objects that are considered mathematical abstractions of agent-based simulations. This collection of objects should be general enough to capture the key features of most simulations, yet specific enough to allow the development of a mathematical theory with meaningful results and algorithms.

Finite dynamical system A finite dynamical system is a time-discrete dynamical system on a finite state set. That is, it is a mapping from a Cartesian product of finitely many copies of a finite set to itself. This finite set is often considered to be a field. The dynamics is generated by iteration of the mapping.

Definition of the Subject

Agent-based simulations are generative or computational approaches used for analyzing "complex systems." What is a "system?" Examples of systems include a collection of molecules in a container, the population in an urban area, and the brokers in a stock market. The entities or *agents* in these three systems would be molecules, individuals and stock brokers, respectively. The agents in such systems interact in the sense that molecules collide, individuals come into contact with other individuals and bro-

kers trade shares. Such systems, often called *multiagent systems*, are not necessarily complex. The label “complex” is typically attached to a system if the number of agents is large, if the agent interactions are involved, or if there is a large degree of heterogeneity in agent character or their interactions.

This is of course not an attempt to define a complex system. Currently there is no generally agreed upon definition of complex systems. It is not the goal of this article to provide such a definition – for our purposes it will be sufficient to think of a complex system as a collection of agents interacting in some manner that involves one or more of the complexity components just mentioned, that is, with a large number of agents, heterogeneity in agent character and interactions and possibly stochastic aspects to all these parts. The global properties of complex systems, such as their global dynamics, emerge from the totality of local interactions between individual agents over time. While these local interactions are well understood in many cases, little is known about the emerging global behavior arising through interaction. Thus, it is typically difficult to construct global mathematical models such as systems of ordinary or partial differential equations, whose properties one could then analyze. Agent-based simulations are one way to create computational models of complex systems that take their place.

An *agent-based simulation*, sometimes also called an *individual-based* or *interaction-based simulation* (which we prefer), of a complex system is in essence a computer program that realizes some (possibly approximate) model of the system to be studied, incorporating the agents and their rules of interaction. The simulation might be deterministic (i. e., the evolution of agent-states is governed by deterministic rules) or stochastic. The typical way in which such simulations are used is to initialize the computer program with a particular assignment of agent states and to run it for some time. The output is a temporal sequence of states for all agents, which is then used to draw conclusions about the complex system one is trying to understand. In other words, the computer program is the model of the complex system, and by running the program repeatedly, one expects to obtain an understanding of the characteristics of the complex system.

There are two main drawbacks to this approach. First, it is difficult to validate the model. Simulations for most systems involve quite complex software constructs that pose challenges to code validation. Second, there are essentially no rigorous tools available for an analysis of model properties and dynamics. There is also no widely applicable formalism for the comparison of models. For instance, if one agent-based simulation is a simplification of an-

other, then one would like to be able to relate their dynamics in a rigorous fashion. We are currently lacking a mathematically rich formal framework that models agent-based simulations. This framework should have at its core a class of mathematical objects to which one can map agent-based simulations. The objects should have a sufficiently general mathematical structure to capture key features of agent-based simulations and, at the same time, should be rich enough to allow the derivation of substantial mathematical results. This chapter presents one such framework, namely the class of *time-discrete dynamical systems over finite state sets*.

The building blocks of these systems consist of a collection of variables (mapping to agents), a graph that captures the dependence relations of agents on other agents, a local update function for each agent that encapsulates the rules by which the state of each agent evolves over time, and an update discipline for the variables (e. g. parallel or sequential). We will show that this class of mathematical objects is appropriate for the representation of agent-based simulations and, therefore, complex systems, and is rich enough to pose and answer relevant mathematical questions. This class is sufficiently rich to be of mathematical interest in its own right and much work remains to be done in studying it. We also remark that many other frameworks such as probabilistic Boolean networks [80] fit inside the framework described here.

Introduction

Computer simulations have become an integral part of today’s research and analysis methodologies. The ever-increasing demands arising from the complexity and sheer size of the phenomena studied constantly push computational boundaries, challenge existing computational methodologies, and motivate the development of new theories to improve our understanding of the potential and limitations of computer simulation. Interaction-based simulations are being used to simulate a variety of biological systems such as ecosystems and the immune system, social systems such as urban populations and markets, and infrastructure systems such as communication networks and power grids.

To model or describe a given system, one typically has several choices in the construction and design of agent-based models and representations. When agents are chosen to be simple, the simulation may not capture the behavior of the real system. On the other hand, the use of highly sophisticated agents can quickly lead to complex behavior and dynamics. Also, use of sophisticated agents may lead to a system that *scales poorly*. That is, a linear

increase in the number of agents in the system may require a non-linear (e. g., quadratic, cubic, or exponential) increase in the computational resources needed for the simulation.

Two common methods, namely *discrete event simulation* and *time-stepped simulation*, are often used to implement agent-based models [1,45,67]. In the discrete event simulation method, each event that occurs in the system is assigned a time of occurrence. The collection of events is kept in increasing order of their occurrence times. (Note that an event occurring at a certain time may give rise to events which occur later.) When all the events that occur at a particular time instant have been carried out, the simulation clock is advanced to the next time instant in the order. Thus, the differences between successive values of the simulation clock may not be uniform. Discrete event simulation is typically used in contexts such as queuing systems [58]. In the time-stepped method of simulation, the simulation clock is always advanced by the same amount. For each value of the simulation clock, the states of the system components are computed using equations that model the system. This method of simulation is commonly used for studying, e. g., fluid flows or chemical reactions. The choice of model (discrete event versus time-stepped) is typically guided by an analysis of the computational speed they can offer, which in turn depends on the nature of the system being modeled, see, e. g., [37].

Tool-kits for general purpose agent-based simulations include Swarm [29,57] and Repast [68]. Such tool-kits allow one to specify more complex agents and interactions than would be possible using, e. g., ordinary differential equations models. In general, it is difficult to develop a software package that is capable of supporting the simulation of a wide variety of physical, biological, and social systems.

Standard or classical approaches to modeling are often based on continuous techniques and frameworks such as ordinary differential equations (ODEs) and partial differential equations (PDEs). For example, there are PDE based models for studying traffic flow [38,47,85]. These can accurately model the emergence of traffic jams for simple road/intersection configurations through, for example, the formation of shocks. However, these models fail to scale to the size and the specifications required to accurately represent large urban areas. Even if they hypothetically were to scale to the required size, the answers they provide (e. g. car density on a road as a function of position and time) cannot answer questions pertaining to specific travelers or cars. Questions of this type can be naturally described and answered through agent-based models. An example of such a system is TRANSIMS (see Subsect. “TRANSIMS

(Transportation, Analysis, Simulation System)”), where an agent-based simulation scheme is implemented through a cellular automaton model. Another well-known example of the change in modeling paradigms from continuous to discrete is given by lattice gas automata [32] in the context of fluid dynamics. Stochastic elements are inherent in many systems, and this usually is reflected in the resulting models used to describe them. A stochastic framework is a natural approach in the modeling of, for example, noise over a channel in a simulation of telecommunication networks [6]. In an economic market or a game theoretic setting with competing players, a player may sometimes decide to provide incorrect information. The state of such a player may therefore be viewed and modeled by a random variable. A player may make certain probabilistic assumptions about other players’ environment. In biological systems, certain features and properties may only be known up to the level of probability distributions. It is only natural to incorporate this stochasticity into models of such systems.

Since applications of stochastic discrete models are common, it is desirable to obtain a better understanding of these simulations both from an application point of view (reliability, validation) and from a mathematical point of view. However, an important disadvantage of agent-based models is that there are few mathematical tools available at this time for the analysis of their dynamics.

Examples of Agent-Based Simulations

In order to provide the reader with some concrete examples that can also be used later on to illustrate theoretical concepts we describe here three examples of agent-based descriptions of complex systems, ranging from traffic networks to the immune system and voting schemes.

TRANSIMS (Transportation, Analysis, Simulation System) TRANSIMS is a large-scale computer simulation of traffic on a road network [64,66,76]. The simulation works at the resolution level of individual travelers, and has been used to study large US metropolitan areas such as Portland, OR, Washington D.C. and Dallas/Fort Worth. A TRANSIMS-based analysis of an urban area requires (i) a population, (ii) a location-based activity plan for each individual for the duration of the analysis period and (iii) a network representation of all transportation pathways of the given area. The data required for (i) and (ii) are generated based on, e. g., extensive surveys and other information sources. The network representation is typically very close to a complete description of the real transportation network of the given urban area.

TRANSIMS consists of two main modules: the *router* and the cellular automaton based *micro-simulator*. The router maps each activity plan for each individual (obtained typically from activity surveys) into a travel route. The micro-simulator executes the travel routes and sends each individual through the transportation network so that its activity plan is carried out. This is done in such a way that all constraints imposed on individuals from traffic driving rules, road signals, fellow travelers, and public transportation schedules are respected. The time scale is typically 1 s.

The micro-simulator is the part of TRANSIMS responsible for the detailed traffic dynamics. Its implementation is based on *cellular automata* which are described in more detail in Subsect. “Cellular Automata”. Here, for simplicity, we focus on the situation where each individual travels by car. The *road network representation* is in terms of *links* (e.g. road segments) and *nodes* (e.g. intersections). The network description is turned into a cell-network description by discretizing each lane of every link into cells. A cell corresponds to a 7.5 m lane segment, and can have up to four neighbor cells (front, back, left and right).

The *vehicle dynamics* is specified as follows. Vehicles travel with discrete velocities 0, 1, 2, 3, 4 or 5 which are constant between time steps. Each update time step brings the simulation one time unit forward. If the time unit is one second then the maximum speed of $v_{\max} = 5$ cells per time unit corresponds to an actual speed of $5 \times 7.5 \text{ m/s} = 37.5 \text{ m/s}$ which is 135 km/h or approximately 83.9 mph.

Ignoring intersection dynamics, the micro-simulator executes three functions for each vehicle in every update: (a) lane-changing, (b) acceleration and (c) movement. These functions can be implemented through four cellular automata, one each for lane change decision and execution, one for acceleration and one for movement. For instance, the acceleration automaton works as follows. A vehicle in TRANSIMS can increase its speed by at most 1 cell per second, but if the road ahead is blocked, the vehicle can come to a complete stop in the same time. The function that is applied to each cell that has a car in it uses the gap ahead and the maximal speed to determine if the car will increase or decrease its velocity. Additionally, a car may have its velocity decreased one unit as determined by a certain deceleration probability. The random deceleration is an important element of producing realistic traffic flow. A major advantage of this representation is that it leads to very light-weight agents, a feature that is critical for achieving efficient scaling.

CImmSim Next we discuss an interaction-based simulation that models certain aspects of the human immune

system. Comprised of a large number of interacting cells whose motion is constrained by the body’s anatomy, the immune system lends itself very well to agent-based simulation. In particular, these models can take into account three-dimensional anatomical variations as well as small-scale variability in cell distributions. For instance, while the number of T-cells in the human body is astronomical, the number of antigen-specific T-cells, for a specific antigen, can be quite small, thereby creating many spatial inhomogeneities. Also, little is known about the global structure of the system to be modeled.

The first discrete model to incorporate a useful level of complexity was *ImmSim* [22,23], developed by Seiden and Celada as a stochastic cellular automaton. The system includes B-cells, T-cells, antigen presenting cells (APCs), antibodies, antigens, and antibody-antigen complexes. Receptors on cells are represented by bit strings, and antibodies use bit strings to represent their epitopes and peptides. Specificity and affinity are defined by using bit string similarity. The bit string approach was initially introduced in [31]. The model is implemented on a regular two-dimensional grid, which can be thought of as a slice of a lymph node, for instance. It has been used to study various phenomena, including the optimal number of human leukocyte antigens in human beings [22], the autoimmunity and T-lymphocyte selection in the thymus [60], antibody selection and hyper-mutation [24], and the dependence of the selection and maturation of the immune response on the antigen-to-receptor’s affinity [15]. The computational limitations of the Seiden-Celada model have been overcome by a modified model, *CimmSim* [20], implemented on a parallel architecture. Its complexity is several orders of magnitude larger than that of its predecessor. It has been used to model hypersensitivity to chemotherapy [19] and the selection of escape mutants from immune recognition during HIV infection [14]. In [21] the *CimmSim* framework was applied to the study of mechanisms that contribute to the persistence of infection with the Epstein–Barr virus.

A Voting Game The following example describes a hypothetical voting scheme. The voting system is constructed from a collection of voters. For simplicity, it is assumed that only two candidates, represented by 0 and 1, contest in the election. There are N voters represented by the set $\{v_1, v_2, \dots, v_N\}$. Each voter has a candidate preference or a *state*. We denote the state of voter v_i by x_i . Moreover, each voter knows the preferences or states of some of his or her friends (fellow voters). This friendship relation is captured by the *dependency graph* which we describe later in Subsect. “Definitions, Background, and Examples”. In-

formally, the dependency graph has as vertices the voters with an edge between each pair of voters that are friends.

Starting from an initial configuration of preferences, the voters cast their votes in some order. The candidate that receives the most votes is the winner. A number of rules can be formulated to decide how each voter chooses a candidate. We will provide examples of such rules later, and as will be seen, the outcome of the election is governed by the order in which the votes are cast as well as the structure of the dependency graph.

Existing Mathematical Frameworks

The field of agent-based simulation currently places heavy emphasis on implementation and computation rather than on the derivation of formal results. Computation is no doubt a very useful way to discover potentially interesting behavior and phenomena. However, unless the simulation has been set up very carefully, its outcome does not formally validate or guarantee the observed phenomenon. It could simply be caused by an artifact of the system model, an implementation error, or some other uncertainty.

A first step in a theory of agent-based simulation is the introduction of a formal framework that on the one hand is precise and computationally powerful, and, on the other hand, is natural in the sense that it can be used to effectively describe large classes of both deterministic and stochastic systems. Apart from providing a common basis and a language for describing the model using a sound formalism, such a framework has many advantages. At a first level, it helps to clarify the key structure in a system. Domain specific knowledge is crucial to deriving good models of complex systems, but domain specificity is often confounded by domain conventions and terminology that eventually obfuscate the real structure.

A formal, context independent framework also makes it easier to take advantage of existing general theory and algorithms. Having a model formulated in such a framework also makes it easier to establish results. Additionally, expressing the model using a general framework is more likely to produce results that are widely applicable. This type of framework also supports implementation and validation. Modeling languages like UML [16] serve a similar purpose, but tend to focus solely on software implementation and validation issues, and very little on mathematical or computational analysis.

Cellular Automata

In this section we discuss several existing frameworks for describing agent-based simulations. Cellular automata

(CA) were introduced by Ulam and von Neumann [84] as biologically motivated models of computation. Early research addressed questions about the computational power of these devices. Since then their properties have been studied in the context of dynamical systems [40], language theory [52], and ergodic theory [51] to mention just a few areas. Cellular automata were popularized by Conway [35] (Game of Life) and by Wolfram [55,86,88]. Cellular automata (both deterministic and stochastic) have been used as models for phenomena ranging from lattice gases [32] and flows in porous media [77] to traffic analysis [33,63,65].

A cellular automaton is typically defined over a regular grid. An example is a two-dimensional grid such as \mathbb{Z}^2 . Each grid point (i, j) is referred to as a *site* or *node*. Each site has a state $x_{i,j}(t)$ which is often taken to be binary. Here t denotes the time step. Furthermore, there is a notion of a *neighborhood* for each site. The neighborhood N of a site is the collection of sites that can influence the future state of the given site. Based on its current state $x_{i,j}(t)$ and the current states of the sites in its neighborhood N , a function $f_{i,j}$ is used to compute the next state $x_{i,j}(t+1)$ of the site (i, j) . Specifically, we have

$$x_{i,j}(t+1) = f_{i,j}(\tilde{x}_{i,j}(t)), \quad (1)$$

where $\tilde{x}_{i,j}(t)$ denotes the tuple consisting of all the states $x_{i',j'}(t)$ with $(i', j') \in N$. The tuple consisting of the states of all the sites is the CA *configuration* and is denoted $x(t) = (x_{i,j}(t))_{i,j}$. Equation (1) is used to map the configuration $x(t)$ to $x(t+1)$. The cellular automaton map or dynamical system, is the map Φ that sends $x(t)$ to $x(t+1)$.

A central part of CA research is to understand how configurations evolve under iteration of the map Φ and what types of dynamical behavior can be generated. A general introduction to CA can be found in [43].

Hopfield Networks

Hopfield networks were proposed as a simple model of associative memories [42]. A discrete Hopfield neural network consists of an undirected graph $Y(V, E)$. At any time t , each node $v_i \in V$ has a state $x_i(t) \in \{+1, -1\}$. Further, each node $v_i \in V$ has an associated *threshold* $\tau_i \in \mathbb{R}$. Each edge $\{v_i, v_j\} \in E$ has an associated weight $w_{i,j} \in \mathbb{R}$. For each node v_i , the neighborhood N_i of v_i includes v_i and the set of nodes that are adjacent to v_i in Y . Formally,

$$N_i = \{v_i\} \cup \{v_j \in V : \{v_i, v_j\} \in E\}.$$

States of nodes are updated as follows. At time t , node v_i computes the function f_i defined by

$$f_i(t) = \text{sgn} \left(-\tau_i + \sum_{v_j \in N_i} w_{i,j} x_j(t) \right),$$

where sgn is the map from \mathbf{R} to $\{+1, -1\}$, defined by

$$\text{sgn}(x) = \begin{cases} 1, & \text{if } x \geq 0 \text{ and} \\ -1, & \text{otherwise.} \end{cases}$$

Now, the state of v_i at time $t + 1$ is

$$x_i(t + 1) = f_i(t).$$

Many references on Hopfield networks (see for example [42,78]) assume that the underlying undirected graph is complete; that is, there is an edge between every pair of nodes. In the definition presented above, the graph need not be complete. However, this does not cause any difficulties since the missing edges can be assigned weight 0. As a consequence, such edges will not play any role in determining the dynamics of the system. Both synchronous and asynchronous update models of Hopfield neural networks have been considered in the literature. For theoretical results concerning Hopfield networks see [69,70] and the references cited therein. Reference [78] presents a number of applications of neural networks. In [54], a Hopfield model is used to study polarization in dynamic networks.

Communicating Finite State Machines

The model of communicating finite state machines (CFSM) was proposed to analyze protocols used in computer networks. In some of the literature, this model is also referred to as “concurrent transition systems” [36].

In the CFSM model, each agent is a process executing on some node of a distributed computing system. Although there are minor differences among the various CFSM models proposed in the literature [17,36], the basic set-up models each process as a finite state machine (FSM). Thus, each agent is in a certain state at any time instant t . For each pair of agents, there is a bidirectional channel through which they can communicate. The state of an agent at time $t + 1$ is a function of the current state and the input (if any) on one or more of the channels. When an agent (FSM) undergoes a transition from one state to another, it may also choose to send a message to another agent or receive a message from an agent. In general, such systems can be synchronous or asynchronous. As can be seen, CFSMs are a natural formalism for studying protocols used in computer networks. The CFSM model has been used extensively to prove properties (e.g. deadlock freedom, fairness) of a number of protocols used in practice (see [17,36] and the references cited therein).

Other frameworks include interacting particle systems [50], and Petri nets [59]. There is a vast literature on both, but space limitations prevent a discussion here.

Finite Dynamical Systems

Another, quite general, modeling framework that has been proposed is that of *finite dynamical systems*, both synchronous and asynchronous. Here the proposed mathematical object representing an agent-based simulation is a time-discrete dynamical system on a finite state set. The description of the systems is modeled after the key components of an agent-based simulation, namely agents, the dependency graph, local update functions, and an update order. This makes a mapping to agent-based simulations natural. In the remainder of this chapter we will show that finite dynamical systems satisfy our criteria for a good mathematical framework in that they are general enough to serve as a broad computing tool and mathematically rich enough to allow the derivation of formal results.

Definitions, Background, and Examples

Let x_1, \dots, x_n be a collection of variables, which take values in a finite set X . (As will be seen, the variables represent the entities in the system being modeled and the elements of X represent their states.) Each variable x_i has associated to it a “local update function” $f_i: X^n \rightarrow X$, where “local” refers to the fact that f_i takes inputs from the variables in the “neighborhood” of x_i , in a sense to be made precise below. By abuse of notation we also let f_i denote the function $X^n \rightarrow X^n$ which changes the i th coordinate and leaves the other coordinates unchanged. This allows for the sequential composition of the local update functions. These functions assemble to a dynamical system

$$\Phi = (f_1, \dots, f_n): X^n \rightarrow X^n,$$

with the dynamics generated by iteration of Φ . As an example, if $X = \{0, 1\}$ with the standard Boolean operators AND and OR, then Φ is a Boolean network.

The assembly of Φ from the local functions f_i can be done in one of several ways. One can update each of the variables simultaneously, that is,

$$\Phi(x_1, \dots, x_n) = (f_1(x_1, \dots, x_n), \dots, f_n(x_1, \dots, x_n)).$$

In this case one obtains a *parallel dynamical system*.

Alternatively, one can choose to update the states of the variables according to some fixed update order, for example, a permutation $(\pi_1, \pi_2, \dots, \pi_n)$ of the set $\{1, \dots, n\}$. More generally, one could use a *word* on the set

$\{1, \dots, n\}$, that is, $\pi = (\pi_1, \dots, \pi_t)$ where t is the length of the word. The function composition

$$\Phi_\pi = f_{\pi_t} \circ f_{\pi_{t-1}} \circ \dots \circ f_{\pi_1}, \quad (2)$$

is called a *sequential dynamical system* (SDS) and, as before, the dynamics of Φ_π is generated by iteration. The case when π is a permutation on $\{1, \dots, n\}$ has been studied extensively [2,3,5,9]. It is clear that using a different permutation or word σ may result in a different dynamical system Φ_σ . Using a word rather than a permutation allows one to capture the case where some vertices have states that are updated more frequently than others.

Remark 1 Notice that for a fixed π , the function Φ_π is a parallel dynamical system: once the update order π is chosen and the local update functions are composed according to π , that is, the function Φ_π has been computed, then $\Phi_\pi(x_1, \dots, x_n) = g(x_1, \dots, x_n)$ where g is a parallel update dynamical system. However, the maps g_i are not local functions.

The dynamics of Φ is usually represented as a directed graph on the vertex set X^n , called the *phase space* of Φ . There is a directed edge from $\mathbf{v} \in X^n$ to $\mathbf{w} \in X^n$ if and only if $\Phi(\mathbf{v}) = \mathbf{w}$. A second graph that is usually associated with a finite dynamical system is its *dependency graph* $Y(V, E)$. In general, this is a directed graph, and its vertex set is $V = \{1, \dots, n\}$. There is a directed edge from i to j if and only if x_i appears in the function f_j . In many situations, the interaction relationship between pairs of variables is symmetric; that is, variable x_i appears in f_j if and only if x_j appears in f_i . In such cases, the dependency

graph can be thought of as an undirected graph. We recall that the dependency graphs mentioned in the context of the voting game (Subject. “A Voting Game”) and Hopfield networks (Subject. “Hopfield Networks”) are undirected graphs. The dependency graph plays an important role in the study of finite dynamical systems and is sometimes listed explicitly as part of the specification of Φ .

Example 2 Let $X = \{0, 1\}$ (the Boolean case). Suppose we have four variables and the local Boolean update functions are

$$f_1 = x_1 + x_2 + x_3 + x_4,$$

$$f_2 = x_1 + x_2,$$

$$f_3 = x_1 + x_3,$$

$$f_4 = x_1 + x_4,$$

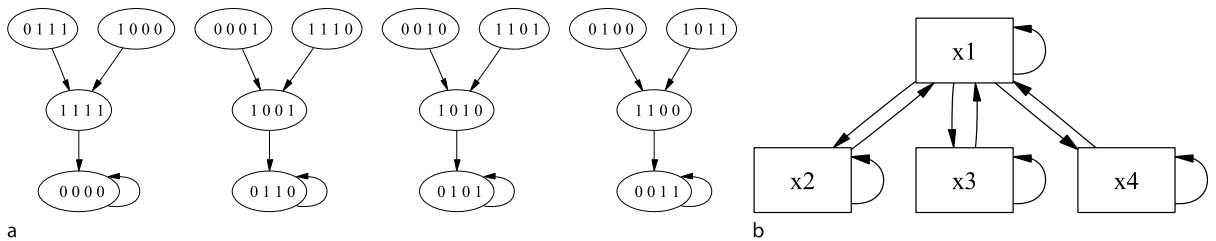
where “+” represents sum modulo 2. The dynamics of the function $\Phi = (f_1, \dots, f_4): X^4 \rightarrow X^4$ is the directed graph in Fig. 1a while the dependency graph is in Fig. 1b.

Example 3 Consider the local functions in the Example 2 above and let $\pi = (2, 1, 3, 4)$. Then

$$\Phi_\pi = f_4 \circ f_3 \circ f_1 \circ f_2: X^4 \rightarrow X^4.$$

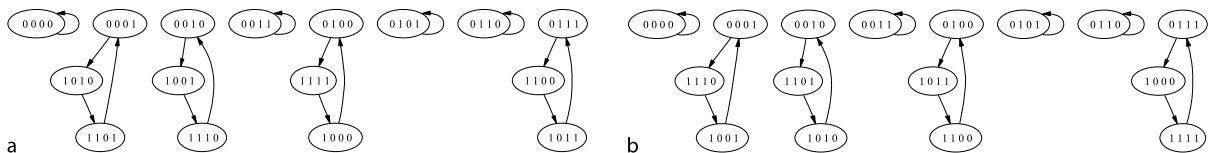
The phase space of Φ_π is the directed graph in Fig. 2a, while the phase space of Φ_γ , where $\gamma = id$ is in Fig. 2b.

Notice that the phase space of any function is a directed graph in which every vertex has out-degree one; this is a characteristic property of deterministic functions.



Agent Based Modeling, Mathematical Formalism for, Figure 1

The phase space of the parallel system Φ (a) and dependency graph of the Boolean functions from Example 2 (b)



Agent Based Modeling, Mathematical Formalism for, Figure 2

The phase spaces from Example 3: Φ_π (a) and Φ_{id} (b)

Making use of Boolean arithmetic is a powerful tool in studying Boolean networks, which is not available in general. In order to have available an enhanced set of tools it is often natural to make an additional assumption regarding X , namely that it is a finite number system, a *finite field* [49]. This amounts to the assumption that there are “addition” and “multiplication” operations defined on X that satisfy the same rules as ordinary addition and multiplication of numbers. Examples include \mathbb{Z}_p , the integers modulo a prime p . This assumption can be thought of as the discrete analog of imposing a coordinate system on an affine space.

When X is a finite field, it is easy to show that for any local function g , there exists a polynomial h such that $g(x_1, \dots, x_n) = h(x_1, \dots, x_n)$ for all $(x_1, \dots, x_n) \in X^n$. To be precise, suppose X is a finite field with q elements. Then

$$g(x_1, \dots, x_n) = \sum_{(c_1, \dots, c_n) \in X^n} g(c_1, \dots, c_n) \prod_{i=1}^n (1 - (x_i - c_i)^{q-1}). \quad (3)$$

This observation has many useful consequences, since polynomial functions have been studied extensively and many analytical tools are available.

Notice that cellular automata and Boolean networks, both parallel and sequential, are special classes of polynomial dynamical systems. In fact, it is straightforward to see that

$$x \wedge y = x \cdot y \quad x \vee y = x + y + xy \quad \text{and} \quad \neg x = x + 1. \quad (4)$$

Therefore, the modeling framework of finite dynamical systems includes that of cellular automata, discussed earlier. Also, since a Hopfield network is a function $X^n \rightarrow X^n$, which can be represented through its local constituent functions, it follows that Hopfield networks also are special cases of finite dynamical systems.

Stochastic Finite Dynamical Systems

The deterministic framework of finite dynamical systems can be made stochastic in several different ways, making one or more of the system’s defining data stochastic. For example, one could use one or both of the following criteria.

- Assume that each variable has a nonempty set of local functions assigned to it, together with a probability distribution on this set, and each time a variable is

updated, one of these local functions is chosen at random to update its state. We call such systems *probabilistic finite dynamical systems* (PFDS), a generalization of probabilistic Boolean networks [81].

- Fix a subset of permutations $T \subseteq S_n$ together with a probability distribution. When it is time for the system to update its state, a permutation $\pi \in T$ is chosen at random and the agents are updated sequentially using π . We call such systems *stochastic finite dynamical systems* (SFDS).

Remark 4 By Remark, each system Φ_π is a parallel system. Hence a SFDS is nothing but a set of parallel dynamical systems $\{\Phi_\pi : \pi \in T\}$, together with a probability distribution. When it is time for the system to update its state, a system Φ_π is chosen at random and used for the next iteration.

To describe the phase space of a stochastic finite dynamical system, a general method is as follows. Let Ω be a finite collection of systems Φ_1, \dots, Φ_t , where $\Phi_i : X^n \rightarrow X^n$ for all i , and consider the probabilities p_1, \dots, p_t which sum to 1. We obtain the stochastic phase space

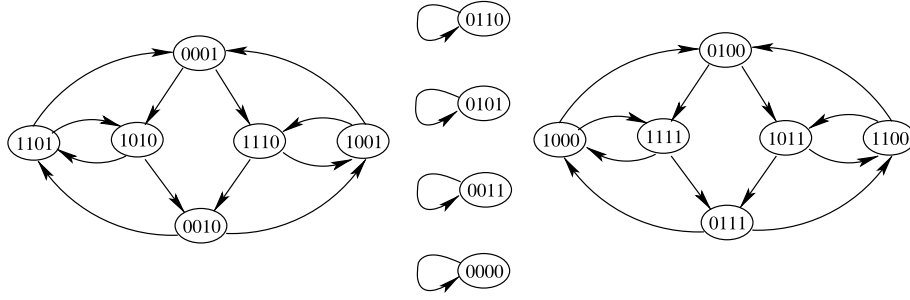
$$\Gamma_\Omega = p_1 \Gamma_1 + p_2 \Gamma_2 + \dots + p_t \Gamma_t, \quad (5)$$

where Γ_i is the phase space of Φ_i . The associated probability space is $\mathcal{F} = (\Omega, 2^\Omega, \mu)$, where the probability measure μ is induced by the probabilities p_i . It is clear that the stochastic phase space can be viewed as a *Markov chain* over the state space X^n . The adjacency matrix of Γ_Ω directly encodes the Markov transition matrix. This is of course not new, and has been done in, e. g., [28,81,83]. But we emphasize the point that even though SFDS give rise to Markov chains *our study of SFDS is greatly facilitated by the rich additional structure* available in these systems. To understand the effect of structural components such as the topology of the dependency graph or the stochastic nature of the update, it is important to study them not as Markov chains but as SFDS.

Example 5 Consider Φ_π and Φ_γ from Example 3 and let Γ_π and Γ_γ be their phases spaces as shown in Fig. 2. Let $p_1 = p_2 = 1/2$. The phase space $(1/2)\Gamma_\pi + (1/2)\Gamma_\gamma$ of the stochastic sequential dynamical system obtained from Φ_π and Φ_γ (with equal probabilities) is presented in Fig. 3.

Agent-Based Simulations as Finite Dynamical Systems

In the following we describe the generic structure of the systems typically modeled and studied through agent-based simulations. The central notion is naturally that of an *agent*.



Agent Based Modeling, Mathematical Formalism for, Figure 3

The stochastic phase space for Example 5 induced by the two deterministic phase spaces of Φ_π and Φ_γ from Fig. 2. For simplicity the weights of the edges have been omitted

Each agent carries a state that may encode its preferences, internal configuration, perception of its environment, and so on. In the case of TRANSIMS, for instance, the agents are the cells making up the road network. The cell state contains information about whether or not the cell is occupied by a vehicle as well as the velocity of the vehicle. One may assume that each cell takes on states from the same set of possible states, which may be chosen to support the structure of a finite field.

The agents interact with each other, but typically an agent only interacts with a small subset of agents, its *neighbors*. Through such an interaction an agent may decide to change its state based on the states (or other aspects) of the agents with which it interacts. We will refer to the process where an agent modifies its state through interaction as an *agent update*. The precise way in which an agent modifies its state is governed by the nature of the particular agent. In TRANSIMS the neighbors of a cell are the adjacent road network cells. From this adjacency relation one obtains a dependency graph of the agents. The local update function for a given agent can be obtained from the rules governing traffic flow between cells.

The updates of all the agents may be scheduled in different ways. Classical approaches include synchronous, asynchronous and event-driven schemes. The choice will depend on system properties or particular considerations about the simulation implementation.

In the case of *CImmSim*, the situation is somewhat more complicated. Here the agents are also the spatial units of the system, each representing a small volume of lymph tissue. The total volume is represented as a 2-dimensional CA, in which every agent has 4 neighbors, so that the dependency graph is a regular 2-dimensional grid. The state of each agent is a collection of counts for the various immune cells and pathogens that are present in this particular agent (volume). Movement between cells is

implemented as diffusion. Immune cells can interact with each other and with pathogens while they reside in the same volume. Thus, the local update function for a given cell of the simulation is made up of the two components of movement between cells and interactions within a cell. For instance, a B cell could interact with the Epstein-Barr virus in a given volume and perform a transition from uninfected to infected by the next time step. Interactions as well as movement are stochastic, resulting in a stochastic finite dynamical system. The update order is parallel.

Example 6 (The voting game (Subsect. “A Voting Game”))

The following scenario is constructed to illustrate how implementation choices for the system components have a clear and direct bearing on the dynamics and simulation outcomes.

Let the voter dependency graph be the star graph on 5 vertices with center vertex a and surrounding vertices b , c , d and e . Furthermore, assume that everybody votes opportunistically using the majority rule: the vote cast by an individual is equal to the preference of the majority of his/her friends with the person’s own preference included. For simplicity, assume candidate 1 is preferred in the case of a tie.

If the initial preference is $x_a = 1$ and $x_b = x_c = x_d = x_e = 0$ then if voter a goes first he/she will vote for candidate 0 since that is the choice of the majority of the neighbor voters. However, if b and c go first they only know a ’s preference. Voter b therefore casts his/her vote for candidate 1 as does c . Note that this is a tie situation with an equal number of preferences for candidate 1 (a) and for candidate 2 (b). If voter a goes next, then the situation has changed: the preference of b and c has already changed to 1. Consequently, voter a picks candidate 1. At the end of the day candidate 1 is the election winner, and the choice of update order has tipped the election!

This example is of course constructed to illustrate our point. However, in real systems it can be much more difficult to detect the presence of such sensitivities and their implications. A solid mathematical framework can be very helpful in detecting such effects.

Finite Dynamical Systems as Theoretical and Computational Tools

If finite dynamical systems are to be useful as a modeling paradigm for agent-based simulations it is necessary that they can serve as a fairly universal model of computation. We discuss here how such dynamical systems can mimic Turing Machines (TMs), a standard universal model for computation. For a more thorough exposition, we refer the reader to the series of papers by Barrett et al. [7,8,10,11,12]. To make the discussion reasonably self-contained, we provide a brief and informal discussion of the TM model. Additional information on TMs can be found in any standard text on the theory of computation (e.g. [82]).

A Computational View of Finite Dynamical Systems: Definitions

In order to understand the relationship of finite dynamical systems to TMs, it is important to view such systems from a computational stand point. Recall that a finite dynamical system $\Phi: X^n \rightarrow X^n$, where X is a finite set, has an underlying dependency graph $Y(V, E)$. From a computational point of view, the nodes of the dependency graph (the agents in the simulation) are thought of as devices that compute appropriate functions. For simplicity, we will assume in this section that the dependency graph is undirected, that is, all dependency relations are symmetric. At any time, the state value of each node $v_i \in V$ is from the specified domain X . The inputs to f_i are the current state of v_i and the states of the neighbors of v_i as specified by Y . The output of f_i , which is also a member of X , becomes the state of v_i at the next time instant. The discussion in this section will focus on sequentially updated systems (SDS), but all results discussed apply to parallel systems as well.

Each step of the computation carried out by an SDS can be thought as consisting of n “mini steps,” in each mini step, the value of the local transition function at a node is computed and the state of the node is changed to the computed value. Given an SDS Φ , a *configuration* C of Φ is a vector $(c_1, c_2, \dots, c_n) \in X^n$. It can be seen that each computational step of an SDS causes a transition from one configuration to another.

Configuration Reachability Problem for SDSs

Based on the computational view, a number of different problems can be defined for SDSs (see for example, [4,10,12]). To illustrate how SDSs can model TM computations, we will focus on one such problem, namely the *configuration reachability* (CR) problem: Given an SDS Φ , an initial configuration C and another configuration C' , will Φ , starting from C , ever reach configuration C' ? The problem can also be expressed in terms of the phase space of Φ . Since configurations such as C and C' are represented by nodes in the phase space, the CR problem boils down to the question of whether there is a directed path in the phase space from C to C' . This abstract problem can be mapped to several problems in the simulation of multi-agent systems. Consider for example the TRANSIMS context. Here, the initial configuration C may represent the state of the system early in the day (when the traffic is very light) and C' may represent an “undesirable” state of the system (such as heavy traffic congestion). Similarly, in the context of modeling an infectious disease, C may represent the initial onset of the disease (when only a small number of people are infected) and C' may represent a situation where a large percentage of the population is infected. The purpose of studying computational problems such as CR is to determine whether one can efficiently predict the occurrence of certain events in the system from a description of the system. If computational analysis shows that the system can indeed reach undesirable configurations as it evolves, then one can try to identify steps needed to deal with such situations.

Turing Machines: A Brief Overview

A Turing machine (TM) is a simple and commonly used model for general purpose computational devices. Since our goal is to point out how SDSs can also serve as computational devices, we will present an informal overview of the TM model. Readers interested in a more formal description may consult [82].

A TM consists of a set Q of states, a one-way infinite input tape and a read/write head that can read and modify symbols on the input tape. The input tape is divided into cells and each cell contains a symbol from a special finite alphabet. An input consisting of n symbols is written on the leftmost n cells of the input tape. (The other cells are assumed to contain a special symbol called *blank*.) One of the states in Q , denoted by q_s , is the designated *start* state. Q also includes two other special states, denoted by q_a (the *accepting* state) and q_r (the *rejecting* state). At any time, the machine is in one of the states in Q . The *transition function* for the TM specifies for each combination of the current

state and the current symbol under the head, a new state, a new symbol for the current cell (which is under the head) and a movement (i. e., left or right by one cell) for the head. The machine starts in state q_s with the head on the first cell of the input tape. Each step of the machine is carried out in accordance with the transition function. If the machine ever reaches either the accepting or the rejecting state, it halts with the corresponding decision; otherwise, the machine runs forever.

A *configuration* of a TM consists of its current state, the current tape contents and the position of the head. Note that the transition function of a TM specifies how a new configuration is obtained from the current configuration.

The above description is for the basic TM model (also called *single tape* TM model). For convenience in describing some computations, several variants of the above basic model have been proposed. For example, in a *multi-tape* TM, there are one or more *work tapes* in addition to the input tape. The work tapes can be used to store intermediate results. Each work tape has its own read/write head and the definitions of configuration and transition function can be suitably modified to accommodate work tapes. While such an enhancement to the basic TM model makes it easier to carry out certain computations, it does not add to the machine's computational power. In other words, any computation that can be carried out using the enhanced model can also be carried out using the basic model.

As in the case of dynamical systems, one can define a *configuration reachability* (CR) problem for TMs: Given a TM M , an initial configuration I_M and another configuration C_M , will the TM starting from I_M ever reach C_M ? We refer to the CR problem in the context of TMs as CR-TM. In fact, it is this problem for TMs that captures the essence of what can be effectively computed. In particular, by choosing the state component of C_M to be one of the halting states (q_a or q_r), the problem of determining whether a function is computable is transformed into an appropriate CR-TM problem. By imposing appropriate restrictions on the resources used by a TM (e.g. the number of steps, the number of cells on the work tapes), one obtains different versions of the CR-TM problem which characterize different computational complexity classes [82].

How SDSs Can Mimic Turing Machines

The above discussion points out an important similarity between SDSs and TMs. Under both of these models, each computational step causes a transition from one configuration to another. It is this similarity that allows one to construct a discrete dynamical system Φ that can simulate

a TM. Typically, each step of a TM is simulated by a short sequence of successive iterations Φ . As part of the construction, one also identifies a suitable mapping between the configurations of the TM being simulated and those of the dynamical system. This is done in such a way that the answer to CR-TM problem is “yes” if and only if the answer to the CR problem for the dynamical system is also “yes.”

To illustrate the basic ideas, we will informally sketch a construction from [10]. For simplicity, this construction produces an SDS Φ that simulates a restricted version of TMs; the restriction being that for any input containing n symbols, the number of work tape cells that the machine may use is bounded by a linear function of n . Such a TM is called a *linear bounded automaton* (LBA) [82]. Let M denote the given LBA and let n denote the length of the input to M . The domain X for the SDS Φ is chosen to be a finite set based on the allowed symbols in the input to the TM. The dependency graph is chosen to be a simple path on n nodes, where each node serves as a representative for a cell on the input tape. The initial and final configurations C and C' for Φ are constructed from the corresponding configurations of M . The local transition function for each node of the SDS is constructed from the given transition function for M in such a way that each step of M corresponds to exactly one step of Φ . Thus, there is a simple bijection between the sequence of configurations that M goes through during its computation and the sequence of states that Φ goes through as it evolves. Using this bijection, it is shown in [10] that the answer to the CR-TM problem is “yes” if and only if Φ reaches C' starting from C . Reference [10] also presents a number of sophisticated constructions where the resulting dynamical system is very simple; for example, it is shown that an LBA can be simulated by an SDS in which X is the Boolean field, the dependency graph is d -regular for some constant d and the local transition functions at all the nodes are *identical*. Such results point out that one does not need complicated dynamical systems to model TM computations.

References [7,8,10] present constructions that show how more general models of TMs can also be simulated by appropriate SDSs. As one would expect, these constructions lead to SDSs with more complex local transition functions.

Barrett et al. [11] have also considered *stochastic* SDS (SSDS), where the local transition function at each node is stochastic. For each combination of inputs, a stochastic local transition function specifies a probability distribution over the domain of state values. It is shown in [11] that SSDSs can effectively simulate computations carried

out by probabilistic TMs (i. e., TMs whose transition functions are stochastic).

TRANSIMS Related Questions

Section “TRANSIMS (Transportation, Analysis, Simulation System)” gave an overview of some aspects of the TRANSIMS model. The micro-simulator module is specified as a functional composition of four cellular automata of the form $\Delta_4 \circ \Delta_3 \circ \Delta_2 \circ \Delta_1$. (We only described Δ_3 which corresponds to velocity updates.) Such a formulation has several advantages. First, it has created an abstraction of the essence of the system in a precise mathematical way without burying it in contextual domain details. An immediate advantage of this specification is that it makes the whole implementation process more straightforward and transparent. While the local update functions for the cells are typically quite simple, for any realistic study of an urban area the problem size would typically require a sequential implementation, raising a number of issues that are best addressed within a mathematical framework like the one considered here.

Mathematical Results on Finite Dynamical Systems

In this section we outline a collection of mathematical results about finite dynamical systems that is representative of the available knowledge. The majority of these results are about deterministic systems, as the theory of stochastic systems of this type is still in its infancy. We will first consider synchronously updated systems.

Throughout this section, we make the assumption that the state set X carries the algebraic structure of a finite field. Accordingly, we use the notation k instead of X . It is a standard result that in this case the number q of elements in k has the form $q = p^t$ for some prime p and $t \geq 1$. The reader may keep the simplest case $k = \{0, 1\}$ in mind, in which case we are effectively considering Boolean networks.

Recall Eq. (3). That is, any function $g: k^n \rightarrow k$ can be represented by a multivariate polynomial with coefficients in k . If we require that the exponent of each variable be less than q , then this representation is unique. In particular Eq. (4) implies that every Boolean function can be represented uniquely as a polynomial function.

Parallel Update Systems

Certain classes of finite dynamical systems have been studied extensively, in particular cellular automata and Boolean networks where the state set is $\{0, 1\}$. Many of these studies have been experimental in nature, however.

Some more general mathematical results about cellular automata can be found in the papers of Wolfram and collaborators [87]. The results there focus primarily on one-dimensional Boolean cellular automata with a particular fixed initial state. Here we collect a sampling of more general results.

We first consider linear and affine systems

$$\Phi = (f_1, \dots, f_n): k^n \rightarrow k^n.$$

That is, we consider systems for which the coordinate functions f_i are linear, resp. affine, polynomials. (In the Boolean case this includes functions constructed from XOR (sum modulo 2) and negation.) When each f_i is a linear polynomial of the form $f_i(x_1, \dots, x_n) = a_{i1}x_1 + \dots + a_{in}x_n$, the map Φ is nothing but a *linear transformation* on k^n over k , and, by using the standard basis, Φ has the matrix representation

$$\Phi \left(\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \right) = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix},$$

where $a_{ij} \in k$ for all i, j .

Linear finite dynamical systems were first studied by Elspas [30]. His motivation came from studying feedback shift register networks and their applications to radar and communication systems and automatic error correction circuits. For linear systems over finite fields of prime cardinality, that is, q is a prime number, Elspas showed that the exact number and length of each limit cycle can be determined from the *elementary divisors* of the matrix A . Recently, Hernandez [41] re-discovered Elspas' results and generalized them to arbitrary finite fields. Furthermore, he showed that the structure of the *tree of transients* at each node of each limit cycle is the same, and can be completely determined from the nilpotent elementary divisors of the form x^a . For *affine* Boolean networks (that is, finite dynamical systems over the Boolean field with two elements, whose local functions are linear polynomials which might have constant terms), a method to analyze their cycle length has been developed in [56]. After embedding the matrix of the transition function, which is of dimension $n \times (n + 1)$, into a square matrix of dimension $n + 1$, the problem is then reduced to the linear case. A fast algorithm based on [41] has been implemented in [44], using the symbolic computation package *Macaulay2*.

It is not surprising that the phase space structure of Φ should depend on invariants of the matrix $A = (a_{ij})$. The rational canonical form of A is a block-diagonal matrix and one can recover the structure of the phase space of A

from that of the blocks in the rational form of A . Each block represents either an invertible or a nilpotent linear transformation. Consider an invertible block B . If $\mu(x)$ is the minimal polynomial of B , then there exists s such that $\mu(x)$ divides $x^s - 1$. Hence $B^s - I = 0$ which implies that $B^s \mathbf{v} = \mathbf{v}$. That is, every state vector \mathbf{v} in the phase space of B is in a cycle whose length is a divisor of s .

Definition 7 For any polynomial $\lambda(x)$ in $k[x]$, the *order* of $\lambda(x)$ is the least integer s such that $\lambda(x)$ divides $x^s - 1$.

The cycle structure of the phase space of Φ can be completely determined from the orders of the irreducible factors of the minimal polynomial of Φ . The computation of these orders involves in particular the factorization of numbers of the form $q^r - 1$, which makes the computation of the order of a polynomial potentially quite costly. The nilpotent blocks in the decomposition of A determine the tree structure at the nodes of the limit cycles. It turns out that all trees at all periodic nodes are identical. This generalizes a result in [55] for additive cellular automata over the field with two elements.

While the fact that the structure of the phase space of a linear system can be determined from the invariants associated with its matrix may not be unexpected, it is a beautiful example of how the right mathematical viewpoint provides powerful tools to completely solve the problem of relating the structure of the local functions to the resulting (or emerging) dynamics. Linear and affine systems have been studied extensively in several different contexts and from several different points of view, in particular the case of cellular automata. For instance, additive cellular automata over more general rings as state sets have been studied, e. g., in [25]. Further results on additive CAs can also be found there. One important focus in [25] is on the problem of finding CAs with limit cycles of maximal length for the purpose of constructing pseudo random number generators.

Unfortunately, the situation is more complicated for nonlinear systems. For the special class of Boolean synchronous systems whose local update functions consist of monomials, there is a polynomial time algorithm that determines whether a given monomial system has only fixed points as periodic points [26]. This question was motivated by applications to the modeling of biochemical networks. The criterion is given in terms of invariants of the dependency graph Y . For a strongly connected directed graph Y (i. e., there is a directed path between any pair of vertices), its *loop number* is the greatest common divisor of all directed loops at a particular vertex. (This number is independent of the vertex chosen.)

Theorem 8 ([26]) *A Boolean monomial system has only fixed points as periodic points if and only if the loop number of every strongly connected component of its dependency graph is equal to 1.*

In [27] it is shown that the problem for general finite fields can be reduced to that of a Boolean system and a linear system over rings of the form $\mathbb{Z}/p^r\mathbb{Z}$, p prime. Boolean monomial systems have been studied before in the cellular automaton context [13].

Sequential Update Systems

The update order in a sequential dynamical system has been studied using combinatorial and algebraic techniques. A natural question to ask here is how the system depends on the update schedule. In [3,5,61,72] this was answered on several levels for the special case where the update schedule is a permutation. We describe these results in some detail. Results about the more general case of update orders described by words on the indices of the local update functions can be found in [34].

Given local update functions $f_i: k^n \rightarrow k$ and permutation update orders σ, π , a natural question is when the two resulting SDS Φ_σ and Φ_π are identical and, more generally, how many different systems one obtains by varying the update order over all permutations. Both questions can be answered in great generality. The answer involves invariants of two graphs, namely the acyclic orientations of the dependency graph Y of the local update functions and the *update graph* of Y . The update graph $U(Y)$ of Y is the graph whose vertex set consists of all permutations of the vertex set of Y [72]. There is an (undirected) edge between two permutations $\sigma = (\sigma_1, \dots, \sigma_n)$ and $\tau = (\tau_1, \dots, \tau_n)$ if they differ by a transposition of two adjacent entries σ_i and σ_{i+1} such that there is no edge in Y between σ_i and σ_{i+1} .

The update graph encodes the fact that one can commute two local update functions f_i and f_j without affecting the end result Φ if i and j are not connected by an edge in Y . That is, $\dots f_i \circ f_j \dots = \dots f_j \circ f_i \dots$ if and only if i and j are not connected by an edge in Y .

All permutations belonging to the same connected component in $U(Y)$ give identical SDS maps. The number of (connected) components in $U(Y)$ is therefore an upper bound for the number of functionally inequivalent SDS that can be generated by just changing the update order. It is convenient to introduce an equivalence relation \sim_Y on S_Y by $\pi \sim_Y \sigma$ if π and σ belong to the same connected component in the graph $U(Y)$. It is then clear that if $\pi \sim_Y \sigma$ then corresponding sequential dynamical systems are identical as maps. This can also be characterized

in terms of acyclic orientations of the graph Y : each component in the update graph induces a unique acyclic orientation of the graph Y . Moreover, we have the following result:

Proposition 9 ([72]) *There is a bijection*

$$F_Y: S_Y / \sim_Y \longrightarrow \text{Acyc}(Y),$$

where S_Y / \sim_Y denotes the set of equivalence classes of \sim_Y and $\text{Acyc}(Y)$ denotes the set of acyclic orientations of Y .

This upper bound on the number of functionally different systems has been shown in [72] to be sharp for Boolean systems, in the sense that for a given Y one constructs this number of different systems, using appropriate combinations of NOR functions.

For two permutations σ and τ it is easy to determine if they give identical SDS maps: one can just compare their induced acyclic orientations. The number of acyclic orientations of the graph Y tells how many functionally different SDS maps one can obtain for a fixed graph and fixed vertex functions. The work of Cartier and Foata [18] on partially commutative monoids studies a similar question, but their work is not concerned with finite dynamical systems.

Note that permutation update orders have been studied sporadically in the context of cellular automata on circle graphs [71] but not in a systematic way, typically using the order $(1, 2, \dots, n)$ or the even-odd/odd-even orders. As a side note, we remark that this work also confirms our findings that switching from a parallel update order to a sequential order turns the complex behavior found in Wolfram's "class III and IV" automata into much more regular or mundane dynamics, see e. g. [79].

The work on functional equivalence was extended to dynamical equivalence (topological conjugation) in [5, 61]. The automorphism group of the graph Y can be made to act on the components of the update graph $U(Y)$ and therefore also on the acyclic orientations of Y . All permutations contained in components of an orbit under $\text{Aut}(Y)$ give rise to dynamically equivalent sequential dynamical systems, that is, to isomorphic phase spaces. However, here one needs some more technical assumptions, i. e., the local functions must be symmetric and induced, see [9]. This of course also leads to a bound for the number of dynamically inequivalent systems that can be generated by varying the update order alone. Again, this was first done for permutation update orders. The theory was extended to words over the vertex set of Y in [34, 75].

The structure of the graph Y influences the dynamics of the system. As an example, graph invariants such as the

independent sets of Y turn out to be in a bijective correspondence with the invariant set of sequential systems over the Boolean field $k = \{0, 1\}$ that have $\text{nor}_t: k^t \rightarrow k$ given by $\text{nor}_t(x_1, \dots, x_t) = (1 + x_1) \cdots (1 + x_t)$ as local functions [73]. This can be extended to other classes such as those with order independent invariant sets as in [39]. We have already seen how the automorphisms of a graph give rise to equivalence [61]. Also, if the graph Y has non-trivial covering maps we can derive simplified or reduced (in an appropriate sense) versions of the original SDS over the image graphs of Y , see e. g. [62, 74].

Parallel and sequential dynamical systems differ when it comes to invertibility. Whereas it is generally computationally intractable to determine if a CA over \mathbf{Z}^d is invertible for $d \geq 2$ [46], it is straightforward to determine this for a sequential dynamical system [61]. For example, it turns out that the only invertible Boolean sequential dynamical systems with symmetric local functions are the ones where the local functions are either the parity function or the logical complement of the parity function [5].

Some classes of sequential dynamical systems such as the ones induced by the nor-function have desirable stability properties [39]. These systems have minimal invariant sets (i. e. periodic states) that do not depend on the update order. Additionally, these invariant sets are stable with respect to configuration perturbations.

If a state \mathbf{c} is perturbed to a state \mathbf{c}' that is not periodic this state will evolve to a periodic state \mathbf{c}'' in one step; that is, the system will quickly return to the invariant set. However, the states \mathbf{c} and \mathbf{c}'' may not necessarily be in the same periodic orbit.

The Category of Sequential Dynamical Systems

As a general principle, in order to study a given class of mathematical objects it is useful to study transformations between them. In order to provide a good basis for a mathematical analysis the objects and transformations together should form a *category*, that is, the class of transformations between two objects should satisfy certain reasonable properties (see, e. g., [53]). Several proposed definitions of a transformation of SDS have been published, notably in [48] and [74]. One possible interpretation of a transformation of SDS from the point of view of agent-based simulation is that the transformation represents the approximation of one simulation by another or the embedding/projection of one simulation into/onto another. These concepts have obvious utility when considering different simulations of the same complex system.

One can take different points of view in defining a transformation of SDS. One approach is to require that

a transformation is compatible with the defining structural elements of an SDS, that is, with the dependency graph, the local update functions, and the update schedule. If this is done properly, then one should expect to be able to prove that the resulting transformation induces a transformation at the level of phase spaces. That is, transformations between SDS should preserve the local and global dynamic behavior. This implies that transformations between SDS lead to transformations between the associated global update functions.

Since the point of view of SDS is that global dynamics emerges from system properties that are defined locally, the notion of SDS transformation should focus on the local structure. This is the point of view taken in [48]. The definition given there is rather technical and the details are beyond the scope of this article. The basic idea is as follows. Let $\Phi_\pi = f_{\pi(n)} \circ \dots \circ f_{\pi(1)}$ and $\Phi_\sigma = g_{\sigma(m)} \circ \dots \circ g_{\sigma(1)}$ with the dependency graphs Y_π and Y_σ , respectively. A transformation $F: \Phi_\pi \rightarrow \Phi_\sigma$ is determined by

- a graph mapping $\varphi: Y_\pi \rightarrow Y_\sigma$ (reverse direction);
- a family of maps $k_{\varphi(v)} \rightarrow k_v$ with $v \in Y_\pi$;
- an order preserving map $\sigma \mapsto \pi$ of update schedules.

These maps are required to satisfy the property that they “locally” assemble to a coherent transformation. Using this definition of transformation, it is shown (Theorem 2.6 in [48]) that the class of SDS forms a category. One of the requirements, for instance, is that the composition of two transformations is again a transformation. Furthermore, it is shown (Theorem 3.2 in [48]) that a transformation of SDS induces a map of directed graphs on the phase spaces of the two systems. That is, a transformation of the local structural elements of SDS induces a transformation of global dynamics. One of the results proven in [48] is that every SDS can be decomposed uniquely into a direct product (in the categorical sense) of indecomposable SDS.

Another possible point of view is that a transformation

$$F: (\Phi_\pi: k^n \rightarrow k^n) \rightarrow (\Phi_\sigma: k^m \rightarrow k^m)$$

is a function $F: k^n \rightarrow k^m$ such that $F \circ \Phi_\pi = \Phi_\sigma \circ F$, without requiring specific structural properties. This is the approach in [74]. This definition also results in a category, and a collection of mathematical results. Whatever definition chosen, much work remains to be done in studying these categories and their properties.

Future Directions

Agent-based computer simulation is an important method for modeling many complex systems, whose global dynamics emerges from the interaction of many local enti-

ties. Sometimes this is the only feasible approach, especially when available information is not enough to construct global dynamic models. The size of many realistic systems, however, leads to computer models that are themselves highly complex, even if they are constructed from simple software entities. As a result it becomes challenging to carry out verification, validation, and analysis of the models, since these consist in essence of complex computer programs. This chapter argues that the appropriate approach is to provide a formal mathematical foundation by introducing a class of mathematical objects to which one can map agent-based simulations. These objects should capture the key features of an agent-based simulation and should be mathematically rich enough to allow the derivation of general results and techniques. The mathematical setting of dynamical systems is a natural choice for this purpose.

The class of finite dynamical systems over a state set X which carries the structure of a finite field satisfies all these criteria. Parallel, sequential, and stochastic versions of these are rich enough to serve as the mathematical basis for models of a broad range of complex systems. While finite dynamical systems have been studied extensively from an experimental point of view, their mathematical theory should be considered to be in its infancy, providing a fruitful area of research at the interface of mathematics, computer science, and complex systems theory.

Bibliography

Primary Literature

1. Bagrodia RL (1998) Parallel languages for discrete-event simulation models. *IEEE Comput Sci Eng* 5(2):27–38
2. Barrett CL, Reidys CM (1999) Elements of a theory of simulation I: Sequential CA over random graphs. *Appl Math Comput* 98:241–259
3. Barrett CL, Mortveit HS, Reidys CM (2000) Elements of a theory of simulation II: Sequential dynamical systems. *Appl Math Comput* 107(2–3):121–136
4. Barrett CL, Hunt III HB, Marathe MV, Ravi SS, Rosenkrantz DJ, Stearns RE, Tosic P (2001) Garden of eden and fixed point configurations in sequential dynamical systems. In: *Proc International Conference on Combinatorics, Computation and Geometry DM-CCG'2001*. pp 95–110
5. Barrett CL, Mortveit HS, Reidys CM (2001) Elements of a theory of simulation III: Equivalence of SDS. *Appl Math Comput* 122:325–340
6. Barrett CL, Marathe MV, Smith JP, Ravi SS (2002) A mobility and traffic generation framework for modeling and simulating ad hoc communication networks. In: *SAC'02 Madrid*, ACM, pp 122–126
7. Barrett CL, Hunt III HB, Marathe MV, Ravi SS, Rosenkrantz DJ, Stearns RE (2003) On some special classes of sequential dynamical systems. *Ann Comb* 7(4):381–408
8. Barrett CL, Hunt III HB, Marathe MV, Ravi SS, Rosenkrantz DJ,

- Stearns RE (2003) Reachability problems for sequential dynamical systems with threshold functions. *Theor Comput Sci* 295(1–3):41–64
9. Barrett CL, Mortveit HS, Reidys CM (2003) Elements of a theory of computer simulation. IV. sequential dynamical systems: fixed points, invertibility and equivalence. *Appl Math Comput* 134(1):153–171
 10. Barrett CL, Hunt III HB, Marathe MV, Ravi SS, Rosenkrantz DJ, Stearns RE (2006) Complexity of reachability problems for finite discrete sequential dynamical systems. *J Comput Syst Sci* 72:1317–1345
 11. Barrett CL, Hunt III HB, Marathe MV, Ravi SS, Rosenkrantz DJ, Stearns RE, Thakur M (2007) Computational aspects of analyzing social network dynamics. In: *Proc International Joint Conference on Artificial Intelligence IJCAI 2007*. pp 2268–2273
 12. Barrett CL, Hunt III HB, Marathe MV, Ravi SS, Rosenkrantz DJ, Stearns RE, Thakur M (2007) Predecessor existence problems for finite discrete dynamical systems. *Theor Comput Sci* 1–2: 3–37
 13. Bartlett R, Garzon M (1993) Monomial cellular automata. *Complex Syst* 7(5):367–388
 14. Bernaschi M, Castiglione F (2002) Selection of escape mutants from immune recognition during hiv infection. *Immunol Cell Biol* 80:307–313
 15. Bernaschi M, Succi S, Castiglione F (2000) Large-scale cellular automata simulations of the immune system response. *Phys Rev E* 61:1851–1854
 16. Booch G, Rumbaugh J, Jacobson I (2005) *Unified Modeling Language User Guide*, 2nd edn. Addison-Wesley, Reading
 17. Brand D, Zafropulo P (1983) On communicating finite-state machines. *J ACM* 30:323–342
 18. Cartier P, Foata D (1969) *Problèmes combinatoires de commutation et réarrangements*. *Lecture Notes in Mathematics*, vol 85. Springer, Berlin
 19. Castiglione F, Agur Z (2003) Analyzing hypersensitivity to chemotherapy in a cellular automata model of the immune system. In: Preziosi L (ed) *Cancer modeling and simulation*. Chapman and Hall/CRC, London
 20. Castiglione F, Bernaschi M, Succi S (1997) Simulating the Immune Response on a Distributed Parallel Computer. *Int J Mod Phys C* 8:527–545; 10.1142/S0129183197000424
 21. Castiglione F, Duca K, Jarrah A, Laubenbacher R, Hochberg D, Thorley-Lawson D (2007) Simulating Epstein–Barr virus infection with C-ImmSim. *Bioinformatics* 23(11):1371–1377
 22. Celada F, Seiden P (1992) A computer model of cellular interactions in the immune system. *Immunol Today* 13(2):56–62
 23. Celada F, Seiden P (1992) A model for simulating cognate recognition and response in the immune system. *J Theor Biol* 158:235–270
 24. Celada F, Seiden P (1996) Affinity maturation and hypermutation in a simulation of the humoral immune response. *Eur J Immunol* 26(6):1350–1358
 25. Chaudhuri PP (1997) *Additive Cellular Automata. Theory and Applications*, vol 1. IEEE Comput Soc Press
 26. Colón-Reyes O, Laubenbacher R, Pareigis B (2004) Boolean monomial dynamical systems. *Ann Comb* 8:425–439
 27. Colón-Reyes O, Jarrah A, Laubenbacher R, Sturfels B (2006) Monomial dynamical systems over finite fields. *Complex Syst* 16(4):333–342
 28. Dawson D (1974) Synchronous and asynchronous reversible Markov systems. *Canad Math Bull* 17(5):633–649
 29. Ebeling W, Schweitzer F (2001) Swarms of particle agents with harmonic interactions. *Theor Biosci* 120–3/4:207–224
 30. Elspas B (1959) The theory of autonomous linear sequential networks. *IRE Trans Circuit Theor*, pp 45–60
 31. Farmer J, Packard N, Perelson A (1986) The immune system, adaptation, and machine learning. *Phys D* 2(1–3):187–204
 32. Frish U, Hasslacher B, Pomeau Y (1986) Lattice-gas automata for the Navier–Stokes equations. *Phys Rev Lett* 56:1505–1508
 33. Fukš H (2004) Probabilistic cellular automata with conserved quantities. *Nonlinearity* 17:159–173
 34. Garcia LD, Jarrah AS, Laubenbacher R (2006) Sequential dynamical systems over words. *Appl Math Comput* 174(1): 500–510
 35. Gardner M (1970) The fantastic combinations of John Conway’s new solitaire game “life”. *Sci Am* 223:120–123
 36. Gouda M, Chang C (1986) Proving liveness for networks of communicating finite-state machines. *ACM Trans Program Lang Syst* 8:154–182
 37. Guo Y, Gong W, Towsley D (2000) Time-stepped hybrid simulation (tshs) for large scale networks. In: *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies*. *Proc. IEEE*, vol 2. pp 441–450
 38. Gupta A, Katiyar V (2005) Analyses of shock waves and jams in traffic flow. *J Phys A* 38:4069–4083
 39. Hansson ÅÅ, Mortveit HS, Reidys CM (2005) On asynchronous cellular automata. *Adv Complex Syst* 8(4):521–538
 40. Hedlund G (1969) Endomorphisms and automorphisms of the shift dynamical system. *Math Syst Theory* 3:320–375
 41. Hernández-Toledo A (2005) Linear finite dynamical systems. *Commun Algebra* 33(9):2977–2989
 42. Hopfield J (1982) Neural networks and physical systems with emergent collective computational properties. *Proc National Academy of Sciences of the USA* 79:2554–2588
 43. Ilachinsky A (2001) *Cellular Automata: A Discrete Universe*. World Scientific, Singapore
 44. Jarrah A, Laubenbacher R, Stillman M, Vera-Licona P (2007) An efficient algorithm for the phase space structure of linear dynamical systems over finite fields. (submitted)
 45. Jefferson DR (1985) Virtual time. *ACM Trans Program Lang Syst* 7(3):404–425
 46. Kari J (2005) Theory of cellular automata: A survey. *Theor Comput Sci* 334:3–33
 47. Keyfitz BL (2004) Hold that light! modeling of traffic flow by differential equations. *Stud Math Libr* 26:127–153
 48. Laubenbacher R, Pareigis B (2003) Decomposition and simulation of sequential dynamical systems. *Adv Appl Math* 30: 655–678
 49. Lidl R, Niederreiter H (1997) *Finite Fields*. Cambridge University Press, Cambridge
 50. Liggett TM (2005) *Interacting particle systems*. *Classics in Mathematics*. Springer, Berlin. Reprint of the 1985 original
 51. Lind DA (1984) Applications of ergodic theory and sofic systems to cellular automata. *Physica D* 10D:36–44
 52. Lindgren K, Moore C, Nordahl M (1998) Complexity of two-dimensional patterns. *J Statistical Phys* 91(5–6):909–951
 53. Mac Lane S (1998) *Category Theory for the Working Mathematician*, 2nd edn. No 5. in GTM. Springer, New York
 54. Macy MW, Kitts JA, Flache A (2003) Polarization in Dynamic Networks: A Hopfield Model of Emergent Structure. In: *Dynamic social network modeling and analysis*. The National Academies Press, Washington D.C., pp 162–173

55. Martin O, Odlyzko A, Wolfram S (1984) Algebraic properties of cellular automata. *Commun Math Phys* 93:219–258
56. Milligan D, Wilson M (1993) The behavior of affine boolean sequential networks. *Connect Sci* 5(2):153–167
57. Minar N, Burkhart R, Langton C, Manor A (1996) The swarm simulation system: A toolkit for building multi-agent simulations. Santa Fe Institute preprint series. <http://www.santafe.edu/research/publications/wpabstract/199606042> Accessed 11 Aug 2008
58. Misra J (1986) Distributed discrete-event simulation. *ACM Comput Surv* 18(1):39–65
59. Moncion T, Hutzler G, Amar P (2006) Verification of biochemical agent-based models using petri nets. In: Robert T (ed) *International Symposium on Agent Based Modeling and Simulation, ABModSim'06*, Austrian Society for Cybernetics Studies, pp 695–700; <http://www.ibisc.univ-evry.fr/pub/basilic/OUT/Publications/2006/MHA06>
60. Morpurgo D, Serenitha R, Seiden P, Celada F (1995) Modelling thymic functions in a cellular automaton. *Int Immunol* 7: 505–516
61. Mortveit HS, Reidys CM (2001) Discrete, sequential dynamical systems. *Discret Math* 226:281–295
62. Mortveit HS, Reidys CM (2004) Reduction of discrete dynamical systems over graphs. *Adv Complex Syst* 7(1):1–20
63. Nagel K, Schreckenberg M (1992) A cellular automaton model for freeway traffic. *J Phys I* 2:2221–2229
64. Nagel K, Wagner P (2006) *Traffic Flow: Approaches to Modelling and Control*. Wiley
65. Nagel K, Schreckenberg M, Schadschneider A, Ito N (1995) Discrete stochastic models for traffic flow. *Phys Rev E* 51: 2939–2949
66. Nagel K, Rickert M, Barrett CL (1997) Large-scale traffic simulation. *Lecture notes in computer science*, vol 1215. Springer, Berlin, pp 380–402
67. Nance RE (1993) A history of discrete event simulation programming languages. *ACM SIGPLAN Notices* 28:149–175
68. North MJ, Collier NT, Vos JR (2006) Experiences creating three implementations of the repast agent modeling toolkit. *ACM Trans Modeling Comput Simulation* 16:1–25
69. Orponen P (1994) Computational complexity of neural networks: A survey. *Nordic J Comput* 1:94–110
70. Orponen P (1996) The computational power of discrete hopfield networks with hidden units. *Neural Comput* 8:403–415
71. Park JK, Steiglitz K, Thruston WP (1986) Soliton-like behavior in automata. *Physica D* 19D:423–432
72. Reidys C (1998) Acyclic Orientations of Random Graphs. *Adv Appl Math* 21:181–192
73. Reidys CM (2001) On acyclic orientations and sequential dynamical systems. *Adv Appl Math* 27:790–804
74. Reidys CM (2005) On Certain Morphisms of Sequential Dynamical Systems. *Discret Math* 296(2–3):245–257
75. Reidys CM (2006) Sequential dynamical systems over words. *Ann Combinatorics* 10(4):481–498
76. Rickert M, Nagel K, Schreckenberg M, Latour A (1996) Two lane traffic simulations using cellular automata. *Physica A* 231: 534–550
77. Rothman DH (1988) Cellular-automaton fluids: A model for flow in porous media. *Geophysics* 53:509–518
78. Russell S, Norwig P (2003) *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Upper Saddle River
79. Schönfisch B, de Roos A (1999) Synchronous and asynchronous updating in cellular automata. *BioSystems* 51: 123–143
80. Shmulevich I, Dougherty ER, Kim S, Zhang W (2002) Probabilistic boolean networks: A rule-based uncertainty model for gene regulatory networks. *Bioinformatics* 18(2):261–274
81. Shmulevich I, Dougherty ER, Zhang W (2002) From boolean to probabilistic boolean networks as models of genetic regulatory networks. *Proc IEEE* 90(11):1778–1792
82. Sipser M (1997) *Introduction to the Theory of Computation*. PWS Publishing Co, Boston
83. Vasershtein L (1969) Markov processes over denumerable products of spaces describing large system of automata. *Probl Peredachi Informatsii* 5(3):64–72
84. von Neumann J, Burks AW (ed) (1966) *Theory of Self-Reproducing Automata*. University of Illinois Press, Champaign
85. Whitham G (1999) *Linear and Nonlinear Waves*, reprint edition edn. *Pure and Applied Mathematics: A Wiley-Interscience Series of Texts, Monographs and Tracts*, Wiley-Interscience, New York
86. Wolfram S (1983) Statistical mechanics of cellular automata. *Rev Mod Phys* 55:601–644
87. Wolfram S (1986) *Theory and Applications of Cellular Automata*, *Advanced Series on Complex Systems*, vol 1. World Scientific Publishing Company, Singapore
88. Wolfram S (2002) *A New Kind of Science*. Wolfram Media, Champaign

Books and Reviews

- Hopcroft JE, Motwani R, Ullman JD (2000) *Automata Theory, Languages and Computation*. Addison Wesley, Reading
- Kozen DC (1997) *Automata and Computability*. Springer, New York
- Wooldridge M (2002) *Introduction to Multiagent Systems*. Wiley, Chichester

Agent Based Modeling and Neoclassical Economics: A Critical Perspective*

SCOTT MOSS

Centre for Policy Modeling, Manchester Metropolitan University Business School, Manchester, UK

Article Outline

Introduction
 Economic Modeling Approaches
 Methodological Issues
 Conditions for Complexity
 Complexity and the Role of Evidence
 Future Directions
 Bibliography

*The remarks about neoclassical economics are drawn from my inaugural lecture [28]

Introduction

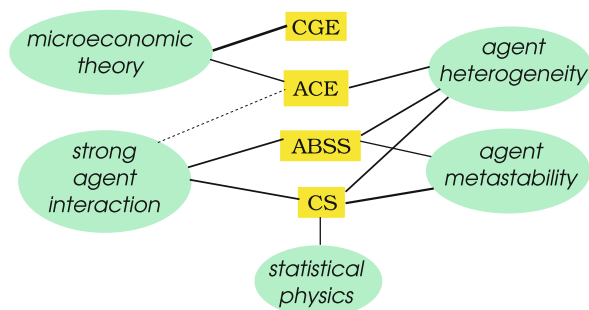
Agent Based Modeling and Neoclassical Economics based modeling naturally generates complexity whereas neoclassical economics is incompatible in principle with complexity. The reasons that preclude complexity in neoclassical economic models also ensure that neoclassical economics cannot describe any society ever observed or that could ever be observed.

The meaning of complexity has been developed, mainly by physicists, to cover unpredictable, episodic volatility and also particular network topologies. In both cases there are nodes representing the components of a system and there are links among the components that can represent interactions amongst those components. Unpredictable, episodic volatility can result from particular forms of behavior by components and the interactions amongst those components. I am not aware of any investigations into relationships between that type of complexity and network topology.

The point to be made here is that the core assumptions *and* the methodology of conventional neoclassical economics preclude the emergence of episodic volatility and render social network topology inconsequential. When elaborated with heterogeneous agents, network topologies might have some effects on the outputs from computational neoclassical economic models – but, again, I am not aware of any systematic investigations into this possibility.

Economic Modeling Approaches

All definitions of complexity take for granted that there will be some specifications of individual components, that in general each component will interact with some other components and there will be some macro level phenomena that could not be described or understood except on the basis of the components and their interactions. The



Agent Based Modeling and Neoclassical Economics: A Critical Perspective, Figure 1
Constraints on model designs

purpose of this section is to categorize the ways in which economists, agent based modelers and complexity scientists approach this micro-macro issue.

There are several strands in the economics and social sciences literatures for building macro analyzes explicitly on micro foundations. The main strands are computable general equilibrium (CGE), agent based computational economics (ACE), agent based social simulation (ABSS) and complexity science (CS) including econophysics and sociophysics. These strands are not all mutually exclusive although there are some conflicting elements among several of them.

Computable General Equilibrium

CGE is the most theoretically constrained of the four strands under consideration. As with general equilibrium theory, it is predicated on the assumptions that households maximize utility and firms maximize profits and that markets clear. The computational load associated with explicit representation of every household and firm leads to the adoption of representative agents intended to capture the behavior of a group such as all households or firms in a particular industrial sector. Some CGE models represent technology with input-output tables; others with marginalist production functions.

Agent Based Computational Economics

An objection to the representative agent device is raised in the ACE literature where the effects of agent heterogeneity are explored. In these models, households can differ in their utility functions (or at least the parameters of those functions) or agents can adopt different game theoretic strategies and firms can employ different production functions. The theoretical core of ACE is not essentially different from that of CGE, both relying on conventional economic theory.

Agent Based Social Simulation

Models reported in the ABSS literature are by and large not driven by traditional theoretical concerns. There is a very wide range in the degree of empirical content: many models being developed to explore “stylized facts”, others driven by qualitative case studies. The latter are often validated against both qualitative micro level data provided by stakeholders and against macro level statistical data.

Complexity Science

Because neoclassical economic theory excludes social embeddedness, the social complexity research that could be

relevant to a consideration of neoclassical economics must be concerned with unpredictable, episodic turbulence. The CS literature on financial markets was seminal and remains well known. The interest in financial markets goes back to Mandelbrot [25] who used financial market data both because it exhibits “outliers” in the relative change series and because of the fineness of the time grain of the price and volume data. A seminal article by Palmer et al. [35] reported a simulation model in which individuals were represented by an early form of software agent and which produced time series marked by the occasional episodes of volatility of the sort observed in real financial market data. However, similar unpredictable episodes of turbulence and volatility have emerged in models of the early post-Soviet Russian economy [31], domestic water consumption [8,15] and models of transactions in inter-mediated markets [29]. Fine grain data exhibiting the same patterns of volatility were found *subsequent to the original publication* of each model.

Relationships Among the Four Approaches

The common thread between CGE and ACE is their common reliance on longstanding economic concepts of utility, continuous production functions, profit maximization, the use of game theory et sic hoc omnes. The common thread between ABSS and complexity science is the importance of social interaction and specifications of individual behavior that are either more ad hoc or based on detailed qualitative evidence for specific cases.

Complex, as distinct from rational, agents’ behavior is “sticky”: it takes non-trivial events or social pressures to make them change. This is the social meaning of metastability. They are also socially embedded in the sense that they interact densely with other agents and are influenced by some of those other agents, generally speaking others who are most like themselves and who they have reason to like and respect [10]. The social difference between influence and imitation is the social equivalent of the physical difference between dissipative and non-dissipative processes. Of course, such influence is meaningless unless the agents differ in some way – they must be heterogeneous.

Methodological Issues

Neoclassical economic theory has no empirically based micro foundation. It has agents of two types: households that maximize utility and firms that maximize profits. Time and expectations are allowed to influence these maximization processes by substituting “expected utility” or “expected profits” for the realized magnitudes. In such models, agents (households or firms) act as if they know

with certainty a population distribution of possible outcomes from their actions. In the terminology introduced by Knight [23], risk pertains when the agent knows the *frequency* distribution of past outcomes that, as in actuarial contexts, are expected with confidence to pertain in the future. When no such frequency distribution is known, then uncertainty prevails. In the sense of Knight (though the terminology gets muddled in the economics literature), the assumption that agents maximize expected utility or expected profits is tenable in conditions of risk but not in conditions of uncertainty. Moreover, it has long been known (with Nobel prizes awarded to Allais [4] and to Daniel Kahneman of Kahneman and Tversky [21] for the demonstrations) that individuals do not act as if they were maximizers of utility or expected utility. Nor is there any evidence that enterprises actually maximize profits. Many economists acknowledge that rationality is bounded and that we lack the cognitive capacity to absorb the required amount of information and then to process that information in order to identify some optimal decision or action. This has given rise to a variety of schools of economic thought such as evolutionary economics (Nelson and Winter [32] is the seminal work here) and Keynesian economics [22] being amongst the best known.

There is evidently a recognizable (and often recognized) divide between the behavioral assumptions of neoclassical economics on the one hand and, on the other hand, common observation, experimental observation (cf. [5]) and a host of business histories (the work of Chandler [12,13] and Penrose [37] being surely the most influential). The evidence shows that the assumptions of neoclassical economics are inaccurate descriptions of the behavior the theories and models are purported to represent. There are two classes of defense for these descriptively inaccurate assumptions. On is the *as-if* defense and the other is the *for-simplicity* defense. These are considered in turn.

The as-if defense was enunciated in several forms by Samuelson [39], Friedman [17] and Alchian [3] in the years around 1950. The details of the differences between Samuelson and Friedman are not germane here. Both argued that their purpose was to model aggregate economic entities such as markets or national economies and descriptively inaccurate assumptions at micro level are permissible provided that the models are descriptively accurate at macro level. Alchian’s contribution was to propose a mechanism. He asserted that, at least in the case of firms, those that were more profitably would be more likely to survive than firms that were less profitable. Consequently, over time, more and more firms would approach more closely to the maximum of profits available to them so that, even if they did not actually seek to maximize profits, the

surviving population of firms would be those that implicitly did actually maximize profits.

The as-if defense is in practice an equilibrium argument. Neoclassical economic models are solved for the simultaneous maximization of utility and profits by all agents – or, at least it is proved that such a solution exists. In such a configuration, no agent has any incentive to change its behavior so the equilibrium presumed to be stable in the small (that is, once reached it is maintained). There is no proof that any general equilibrium model with an arbitrary number of agents is stable in the large (that is, that any feasible solution is an attractor of the system as a whole). The Alchian form of the as-if defense does not take into account any effects of agent interaction or influence of any agent by any other agent. In empirical – that is to say, econometric – testing of neoclassical models, extreme events and observations are dismissed as outliers and removed from the data set being used for the testing or else their effect is encapsulated by specially constructed dummy variables.

The for-simplicity defense rests on the presumption that simpler models are always to be preferred to more complicated models and the achievement of simplicity justifies making assumptions about behavior and environment that are not justified by evidence. The author has for many years now justified this claim by choosing any arbitrary leading economics journal and searching the most recent issue for an assumption made “for simplicity”. On every occasion, the assumption made “for simplicity” has turned out to be an assumption that changed the nature of an empirical problem being addressed so that it conformed to the requirements (such as convexity or absence of externalities) of the mathematical technique being applied to the analysis. Seven of the eleven papers in, at the time of writing, the most recent (November 2007) issue of the *Quarterly Journal of Economics* appealed to the value of simplicity or, in one case, tractability to justify assumptions or specifications that were not justified empirically. The direct quotations are:

Tractability obviously dictated the use of a simple summary statistic of the distribution of legislator ideologies (see p. 1418 in [14]).

The fact, established below, that deriving the properties of the seats-votes relationship requires consideration only of the properties of the univariate distribution of ζ as opposed to those of the bivariate distribution of σ and μ considerably simplifies the analysis (see p. 1480 in [9]).

We make three key assumptions to simplify the analysis. First, we assume that all jobs last indef-

initely once found (i.e., there is no subsequent job destruction). Second, anticipating our empirical findings, we assume that wages are exogenously fixed, eliminating reservation-wage choices. Third, we assume that utility is separable in consumption and search effort (see p. 1516 in [11]).

A more conventional timing assumption in search models without savings is that search in period t leads to a job that begins in period $t + 1$. Assuming that search in period t leads to a job in period t itself simplifies the analytic expressions... (see p. 1517 in [11]).

For simplicity, we'll assume that $\beta_{\text{SAT}}(X_{i,s})$, $\beta_{\text{TEST}}(X_{i,s})$ and $\beta_{\text{OTH}}(X_{i,s})$ are linear in $X_{i,s}$ and can thus be written We will further assume that the random utility component is independent and identically distributed (i. i. d.) from a type 1 extreme value distribution (see p. 1616 in [19]).

For simplicity, all households represent two-earner married couples of the same age (see p. 1683 in [33]).

For simplicity, the model assumes that the highest 35 years of earnings correspond to the ages between 25 and 59 (see p. 1685 in [33]).

We follow Auerbach and Kotlikoff (1987) by measuring efficiency gains from social security privatization using an LSRA that compensates households that would otherwise lose from reform. To be clear, the LSRA is not being proposed as an actual government institution. Instead, it is simply a hypothetical mechanism that allows us to measure the standard Hicksian efficiency gains in general equilibrium associated with privatization (see p. 1687 in [33]).

Assume for simplicity that these batch sizes are fixed for each product class Given these fixed batch sizes for the two classes of product, the firm maximizes profits by deciding how many production runs... [to] undertake... (see pp. 1731–1732 in [7]).

We adopt a number of simplifying assumptions to focus on the main implications of this framework. First, we assume that the relationship between the firm and each manager is short-term. Second, when $x_{i,k} = z_{i,k}$, the manager obtains a private benefit. We assume that managers are credit-constrained and cannot compensate principals for these private benefits and that these private benefits are sufficiently large so that it is not profitable for the principal to utilize incentive contracts to induce managers

to take the right action. These assumptions imply that delegation will lead to the implementation of the action that is preferred by the manager . . . (see p. 1769 in [1]).

All but the first of these quotations are from theoretical papers and the “simplifications” enable the authors to produce equilibrium solutions to their models. No one has ever knowingly observed an equilibrium and in a world where not everything is convex due to (for example) economies of large scale production and where computational capacities limit cognitive abilities, in principle no equilibrium ever will be observed. Indeed, Radner [38] showed that a *necessary* condition for general equilibrium to exist is that all agents have unlimited computational capacities if trading takes place at a sequence of dates. In the core general equilibrium model, all transactions are agreed at a single moment for all time. The “simplifications” required to produce equilibrium models cannot therefore be justified on the basis of relevance to empirical observation. They also ensure that the models cannot capture complexity.

Conditions for Complexity

The social and behavioral conditions for complexity manifest as unpredictable, episodic volatility appears to be the following:

- Individuals behave in routine ways unless some non-trivial event or events or social pressure from other individuals induce them to change their behavior.
- Individuals interact with other individuals.
- Individuals influence but do not generally imitate one another.
- Interactions amongst individuals and individual behavior are not dominated by events that do not arise from that interaction and behavior.

These conditions were first noticed as a general phenomenon in physical models and articulated by Jensen [20] as metastability, dense interaction, dissipation, and coldness of the system, respectively. The phenomenon of unpredictable, clustered volatility in social models had previously been noticed as had its similarity to self organized criticality as described by statistical physicists starting with Bak et al. [6].

Complexity and Social Volatility

Volatility in social statistical time series and power law distributed cross sectional data have long been observed by

statisticians and social scientists. Vilfredo Pareto [36] discovered that the personal distribution of income is power law distributed, a finding which has been replicated widely across countries and time. The same phenomenon is now well known to characterize word use [43] city sizes [44], firm sizes (including market shares) [41], distributions of links between internet sites [2] and a host of other cross sectional distributions. Where firm sizes and market shares are concerned, there have been strands in the industrial economics literature reporting models yielding that result. However, the observed results have not been explained by models in which households maximize utility and firms maximize profits. As Simon and Bonini [41] point out, some variant to Gibrat’s Law (or the law of proportional effect), which states that the growth rate of individuals (say firms) is not correlated with individual size, will generate one highly skewed distribution or another and the particular distribution can be refined by an appropriate choice of the representation of the law.

These desired results also emerged from a series of models based on plausible or empirically based specifications of individual behavior and social interaction in agent based social simulation models. In capturing stakeholders’ perceptions of the behavior and social interactions of relevant classes of individuals and also in relying on well validated propositions from social psychology and cognitive science, models were implemented that produced the sort of skewed distributions that we observe in practice. Episodic volatility followed from the metastability and social embeddedness of agents. In the nature of this process, most changes are relatively small in magnitude but a few changes are very large. This results in fat-tailed distributions of relative changes in variable values at macro level and also in cross sectional data as described by Gibrat’s Law. In practice, the large relative changes tend to be bunched together in unpredictable episodes of volatility.

While these results arise naturally in evidence-driven ABSS models, they are not easily reconciled with neoclassical economic theory. As Krugman [24] (quoted by Eeckhout [16]) had it, “We have to say that the rank-size rule is a major embarrassment for economic theory: one of the strongest statistical relationships we know, lacking any clear basis in theory”.

Complexity and Social Network Topology

Social network topologies can obviously have no meaning in a model comprised by representative agents. In ACE, ABSS and CS models, there will always be a network of social interaction. However, the nature of the interaction can be very different across the different approaches.

In ACE models, it is common to find social interaction taking the form of games – typically the Prisoners’ Dilemma. A nice example of such a model is Tesfatsion’s labor market model using McFadzean and Tesfatsion’s [26,42] Trading Network Game. In Tesfatsion’s labour market model, there is a fixed number of workers and a fixed number of employers identical in their total offers of labour and of employment, respectively. Each worker (resp. employer) ascribes a value of utility to an employment arrangement with any employer (resp. worker). The utility starts out at a some exogenous value and is then increased or reduced depending on the experience at each trading date. The experience is the combination of cooperation and defection by each party to the employment relation at each time step. The social network in this model

... is represented in the form of a directed graph in which the vertices $V(E)$ of the graph represent the work suppliers and employers, the edges of the graph (directed arrows) represent work offers directed from work suppliers to employers, and the edge weight on any edge denotes the number of accepted work offers (contracts) between the work supplier and employer connected by the edge (see p. 431 in [42]).

The topology of this network depends on the outcomes of sequences of prisoners’ dilemma games determining the utilities of workers and employers to one another. Every worker can see every employer and conversely so that the directed links between agents are limited by the number of work contracts into which each agent can engage. After some arbitrary number of time steps, the strategies of the agents are represented as genes and a genetic algorithm is applied so that, over a whole simulation, the elements of the most successful defect/cooperate strategies become more dominant. Since these strategies determine the outcomes of the prisoners’ dilemma games, the social network continues to evolve with utility enhancing strategies becoming more dominant.

In a recent (at the time of writing) issue of The Journal of Economic Dynamics and Control, Page and Tassier [34] modeled the development of chain stores across markets. A firm was defined by its product. Each product was assigned an “intrinsic quality” represented by an integer drawn at random from a distribution $\theta(q)$, and a set of I “hedonic attributes” represented by I positive integers in a range from 0 to some arbitrary, user selected number A . Consumers are represented by utility functions that are positively related to “quality” and negatively related

to the difference between some desired set of hedonic attributes and the hedonic attributes of the product. There are a number (set by the model user) of discrete markets. Page and Tassier then run a variety of simulations that allow for firms to replicate themselves across markets or, through lack of demand, to leave markets.

These two models seem to be representative of a wide class of ACE models. In the first place, agents are defined by utility functions or game theoretic strategies so that the behavior of any individual agent is either fixed or responds smoothly to infinitesimal changes in prices, incomes or whatever other arguments might populate its utility function. In either event, agents cannot be metastable and follow behavioral routines until (but only until) some significant stimulus causes them to change their behavioral responses. In the second place, agents’ preferences and responses are not influenced by the preferences or actions of any other agents like themselves. That is, their behavior as determined by their utility functions or game theoretic strategies will respond to market signals or the actions of the other agent in their game but not to communications with or observations of any other agents. These agents are not, in the words of Granovetter [18], socially embedded especially since it is rare in a neoclassical model for there to be more than two players in a game and unheard-of for there to be more than three (cf. [27]).

Whilst we cannot state with authority that the conditions of metastability, social influence and the consistency principle are necessary for complexity to emerge at macro level from micro level behavior, these conditions have characterized the social simulation models that have produced the episodic and unpredictable volatility associated with complexity. The absence of social embeddedness in the neoclassical ACE models must also explain their lack of any representation of social (as distinct from merely economic) networks.

Complexity and the Role of Evidence

An interesting and fairly typical feature of papers reporting neoclassical models – both theoretical and computational with agents – is that they motivate the modeling exercise by appeal to some empirical, macro level economic phenomenon and then ignore evidence about the micro level behavior that might bring about such phenomena. This practice can be seen in both of the ACE examples described in Sect. “Conditions for Complexity”.

Tesfatsion [42] motivates her model on more theoretical grounds than do Page and Tassier [34]. She wrote:

Understanding the relationship between market structure, market behavior, and market power in

markets with multiple agents engaged in repeated strategic interactions has been a major focus of analytical, empirical, and human-subject experimental researchers in industrial organization since the early 1970s. To date, however, definitive conclusions have been difficult to obtain.

She goes on to cite “a unified theoretical treatment of oligopoly decision-making”, an article on empirical findings with respect to market power that looks only at industry level statistical measures, and some work with experimental subjects. No references are made, either by Tesfatsion or those she cites, to any case studies of the “repeated strategic interactions” in which the “multiple agents” engage.

Page and Tassier give more historical detail. Their motivation turns on:

Chain stores and franchises dominate the American economic landscape. A drive through any moderately sized city reveals remarkable conformity in restaurants, stores, and service centers. Anyone who so desires can eat at Applebee’s, shop at Wal-Mart, and grab a Starbuck’s latte grande while getting her car brakes done at Midas (see p. 3428 in [34]).

For example, in many markets, Lowe’s and Home Depot capture a significant portion of the home improvement market. These big box stores drove many small independent hardware stores and lumber yards out of business. The residual demand resides in niches that can be filled by hardware chains specializing in upscale home furnishings like Restoration Hardware. ... Often, when Lowe’s enters a market, it creates a niche for Restoration Hardware as well. And, as both Lowe’s and Restoration Hardware enter more and more markets, they in turn create additional common niches that can be filled by even more chains. Thus, chains beget chains (see p. 3429 in [34]).

So this article claims a clear and direct historical basis. And yet

... To capture the increasing correlation in niches formally, we introduce two new concepts, the niche landscape and the differential niche landscape. The former plots the quality required to enter a market at a given set of hedonic attributes. The latter plots the differences in two niche landscapes. In the presence of chains, differential niche landscapes become flat, i. e. the niche landscapes become correlated across markets (see p. 3429 in [34]).

The representation of the actors in this framework has been discussed above. At no stage is the agent design discussed in relation to any empirical accounts of the behavior and motivations of the managers of Wal-Mart, Starbucks, Lowes, Restoration Hardware or any other enterprise or any consumer.

This is, of course, the way of neoclassical economics and it has extended to ACE research as well. What is perhaps more unsettling is that it has also extended to the bastions of complexity science – the econophysicists.

There is a long literature now on complexity and financial markets and also on complexity and the formation of opinions – opinion dynamics. There are at least two reasons for the popularity amongst physicists of financial market modeling. First, there are long series of very fine grain data. Second, the data exhibits the unpredictable, episodic volatility associated with complexity. The popularity of opinion dynamics cannot be based on the quality of the data – even at macro level – because that quality is much more coarse grain and inconsistent over time than financial market data. Nonetheless, the two literatures are marked by the heavy presence and influence of physicists and by the lack of connection between their agent designs and any available evidence about the behavior of traders in financial markets or voters or others acting on or expressing their opinions.

A good example from the opinion dynamics literature – chosen at random from *The European Physical Journal B* – is by Schweitzer and Hoyst [40], “Modeling collective opinion formation by means of active Brownian particles”. The motivation for their article is

The formation of public opinion is among the challenging problems in social science, because it reveals a complex dynamics, which may depend on different internal and external influences. We mention the influence of political leaders, the biasing effect of mass media, as well as individual features, such as persuasion or support for other opinions.

We immediately have complexity and social relevance to motivate an article on social dynamics in a physics journal. However, there is no empirical justification for modeling individuals who form opinions as active Brownian particles. The apparent complexity in the outcomes of social processes of opinion formation can be produced by the non-linear feedbacks of fields of active Brownian particles. Whether individuals actually behave in this way is not addressed by Schweitzer and Hoyst or, as far as I know, by any contributor to the opinion dynamics literature.

Much the same can be said of the econophysics literature on financial markets. The clustered volatility associated with complexity is readily produced by physical models with characteristics of metastability, dissipation and dense patterns of interaction. What the econophysicists fail to address is the question of whether their particular formulations – and active Brownian particle is just one of many examples – are *descriptively accurate* representations of the individual actors whose behavior they are seeking to analyze.

In this regard, the econophysicists are not better scientists than neoclassical economists. It can be said in favor of neoclassical (including ACE) economists that they are at least following in a long tradition when they ignore the relationship between what people actually do and how agents are modeled. In the long history of the physical sciences, however, observation and evidence at micro and macro level and all levels in between has dominated theory (cf. [30]). There are some at least in the ABSS research community who would prefer our colleagues with backgrounds in the physical scientists to follow their own methodological tradition in this regard and not that of the neoclassical economists.

Future Directions

Complexity science is not a niche research interest in the social sciences. Societies are complex and all social science should be complexity science. However, any social science that excludes social interaction and inertia or routine necessarily suppresses complexity. As noted here, the adoption of utility theory and representative agents by neoclassical economists (and other social scientists influenced by them) amounts to the exclusion of behavioral inertia and social interaction, respectively. To drop both utility and representative agents and to build analyzes bottom up from a sound basis in evidence would produce a better – very likely, a good – body of economic analysis. But the transition from present convention would be enormous – a transition that experience shows to be beyond the capacity of current and previous generations of mainstream economists. Not only would they have to abandon theories that drive and constrain their research but also their whole epistemological and wider methodological stance. They would have to accept that prediction and forecasting cannot be core methodological objectives and that theories are built by abstracting from detailed evidence based social simulation models the designs and outputs from which have been validated by stakeholders in a range of contexts. This would be a future direction guided by good science.

Bibliography

1. Acemoglu D, Aghion P, Lelarge C, Van Reenen J, Zilibotti F (2007) Technology, information, and the decentralization of the firm. *Q J Econ* 122(4):1759–1799. <http://www.mitpressjournals.org/doi/abs/10.1162/qjec.2007.122.4.1759>
2. Adamic LA, Huberman BA (1999) Power-law distribution of the World Wide Web. *Science* 287(5461):2115
3. Alchian AA (1950) Uncertainty, evolution and economic theory. *J Political Econ* 58(2):211–221
4. Allais M (1953) Le comportement de l'homme rationnel devant le risque: Critiques des postulats et axiomes de l'école américaine. *Econometrica* 21(4):503–546
5. Anderson JR (1993) Rules of the mind. Lawrence Erlbaum Associates, Hillsdale
6. Bak P, Tang C, Wiesenfeld K (1987) Self organized criticality: An explanation of $1/f$ noise. *Phys Rev Lett* 59(4):381–384
7. Bartel A, Ichniowski C, Shaw K (2007) How does information technology affect productivity? Plant-level comparisons of product innovation, process improvement, and worker skills. *Q J Econ* 122(4):1721–1758. <http://www.mitpressjournals.org/doi/abs/10.1162/qjec.2007.122.4.1721>
8. Barthelemy O (2006) Untangling scenario components with agent based modeling: An example of social simulations of water demand forecasts. Ph D thesis, Manchester Metropolitan University
9. Besley T, Preston I (2007) Electoral bias and policy choice: Theory and evidence. *Q J Econ* 122(4):1473–1510. <http://www.mitpressjournals.org/doi/abs/10.1162/qjec.2007.122.4.1473>
10. Brown R (1965) Social psychology. The Free Press, New York
11. Card D, Chetty R, Weber A (2007) Cash-on-hand and competing models of intertemporal behavior: New evidence from the labor market. *Q J Econ* 122(4):1511–1560. <http://www.mitpressjournals.org/doi/abs/10.1162/qjec.2007.122.4.1511>
12. Chandler AD (1962) Strategy and structure: Chapters in the history of the american industrial enterprise. MIT Press, Cambridge
13. Chandler AD (1977) The visible hand: The managerial revolution in american business. Harvard University Press
14. Coate S, Brian K (2007) Socially optimal districting: A theoretical and empirical exploration. *Q J Econ* 122(4):1409–1471
15. Downing TE, Moss S, Pahl Wostl C (2000) Understanding climate policy using participatory agent based social simulation. In: Moss S, Davidsson P (eds) Multi agent based social simulation. Lecture Notes in Artificial Intelligence, vol 1979. Springer, Berlin, pp 198–213
16. Eeckhout J (2004) Gibrat's law for (all) cities. *Am Econ Rev* 94(5):1429–1451
17. Friedman M (1953) The methodology of positive economics. In: Essays on Positive Economics. University of Chicago Press, Chicago
18. Granovetter M (1985) Economic action and social structure: The problem of embeddedness. *Am J Sociol* 91(3):481–510
19. Jacob BA, Lefgren L (2007) What do parents value in education? An empirical investigation of parents' revealed preferences for teachers. *Q J Econ* 122(4):1603–1637. <http://www.mitpressjournals.org/doi/abs/10.1162/qjec.2007.122.4.1603>
20. Jensen H (1998) Self-organized criticality: Emergent complex behavior in physical and biological systems. Cambridge University Press, Cambridge

21. Kahneman D, Tversky A (1979) Prospect theory: An analysis of decision under risk. *Econometrica* 47(2):263–292
22. Keynes JM (1935) The general theory of employment, interest and money. Macmillan, London
23. Knight FH (1921) Risk, uncertainty and profit. Houghton-Mifflin
24. Krugman P (1995) Development, geography and economic theory. MIT Press, Cambridge
25. Mandelbrot B (1963) The variation of certain speculative prices. *J Bus* 36(4):394–419
26. McFadzean D, Tesfatsion L (1999) A c++ platform for the evolution of trade networks. *Comput Econ* 14:109–134
27. Moss S (2001) Game theory: Limitations and an alternative. *J Artif Soc and Soc Simul* 4(2)
28. Moss S (1999) Relevance, realism and rigour: A third way for social and economic research. Technical Report 99-56, Centre for Policy Modeling, Manchester Metropolitan University
29. Moss S (2002) Policy analysis from first principles. *Proc US Nat Acad Sci* 99(Suppl. 3):7267–7274
30. Moss S, Edmonds B (2005) Towards good social science. *J Artif Soc and Soc Simul* 8(4). ISSN 1460-7425. <http://jasss.soc.surrey.ac.uk/8/4/13.html>
31. Moss S, Kuznetsova O (1996) Modeling the process of market emergence. In: Owsinski JW, Nahorski Z (eds) Modeling and analyzing economies in transition. MODEST, Warsaw, pp 125–138
32. Nelson RR, Winter SG (1982) An evolutionary theory of economic change. Harvard University Press, Cambridge
33. Nishiyama S, Smetters K (2007) Does social security privatization produce efficiency gains? *Q J Econ* 122(4):1677–1719. <http://www.mitpressjournals.org/doi/abs/10.1162/qjec.2007.122.4.1677>
34. Page SE, Tassier T (2007) Why chains beget chains: An ecological model of firm entry and exit and the evolution of market similarity. *J Econ Dyn Control* 31(10):3427–3458
35. Palmer R, Arthur WB, Holland JH, LeBaron B, Taylor P (1993) Artificial economic life: A simple model for a stock market. *Physica D* 75:264–274
36. Pareto V (1896–1897) Cours d'économie politique professé à l'Université de Lausanne. Rouge, Lausanne
37. Penrose ET (1959) The theory of the growth of the firm. Wiley, New York
38. Radner R (1968) Competitive equilibrium under uncertainty. *Econometrica* 36(1):31–58
39. Samuelson PA (1949) Foundations of economic analysis. Harvard University Press
40. Schweitzer F, Hoyst JA (2000) Modeling collective opinion formation by means of active Brownian particles. *Europ Phys J B – Condens Matter Complex Syst* 15(4):723–732
41. Simon HA, Bonini CP (1958) The size distribution of business firms. *Am Econ Rev* 48(4):607–617. ISSN 0002-8282 <http://links.jstor.org/sici?sici=0002-8282%28195809%2948%3A4%3C607%3ATS%DOBF%3E2.0.CO%3B2-3>
42. Tesfatsion L (2001) Structure, behavior, and market power in an evolutionary labor market with adaptive search. *J Econ Dyn Control* 25(3–4):419–457
43. Zipf GK (1935) The psycho-biology of language. Houghton Mifflin, Boston
44. Zipf GK (1949) Human behavior and the principle of least effort. Addison-Wesley, Cambridge

Agent Based Modeling and Simulation

STEFANIA BANDINI, SARA MANZONI,

GIUSEPPE VIZZARI

Complex Systems and Artificial Intelligence Research Center, University of Milan-Bicocca, Milan, Italy

Article Outline

Glossary

Definition of the Subject

Introduction

Agent-Based Models for Simulation

Platforms for Agent-Based Simulation

Future Directions

Bibliography

Glossary

Agent The definition of the term agent is controversial even inside the restricted community of computer scientists dealing with research on agent models and technologies [25]. A weak definition, that could be suited to describe the extremely heterogeneous approaches in the agent-based simulation context, is “an autonomous entity, having the ability to decide the actions to be carried out in the environment and interactions to be established with other agents, according to its perceptions and internal state”.

Agent architecture The term agent *architecture* [53] refers to the internal structure that is responsible of effectively selecting the actions to be carried out, according to the perceptions and internal state of an agent. Different architectures have been proposed in order to obtain specific agent behaviors and they are generally classified into *deliberative* and *reactive* (respectively, *hysteretic* and *tropistic*, according to the classification reported in [29]).

Autonomy The term autonomy has different meanings, for it represents (in addition to the control of an agent over its own internal state) different aspects of the possibility of an agent to decide about its own actions. For instance, it may represent the possibility of an agent to decide (i) about the timing of an action, (ii) whether or not to fulfill a request, (iii) to act without the need of an external trigger event (also called pro-activeness or proactivity) or even (iv) basing on its personal experience instead of hard-wired knowledge [53]. It must be noted that different agent models do not generally embody all the above notions of autonomy.

Interaction “An interaction occurs when two or more agents are brought into a dynamic relationship through a set of reciprocal actions” [22].

Environment “The environment is a first-class abstraction that provides the surrounding conditions for agents to exist and that mediates both the interaction among agents and the access to resources” [66].

Platform for agent-based simulation a software framework specifically aimed at supporting the realization of agent-based simulation systems; this kind of framework often provides abstractions and mechanisms for the definition of agents and their environments, to support their interaction, but also additional functionalities like the management of the simulation (e. g. set-up, configuration, turn management), its visualization, monitoring and the acquisition of data about the simulated dynamics.

Definition of the Subject

Agent-Based Modeling and Simulation – an approach to the modeling and simulation of a system in which the overall behavior is determined by the local action and interaction of a set of agents situated in an environment. Every agent chooses the action to be carried out on the basis of its own behavioral specification, internal state and perception of the environment. The environment, besides enabling perceptions, can regulate agents’ interactions and constraint their actions.

Introduction

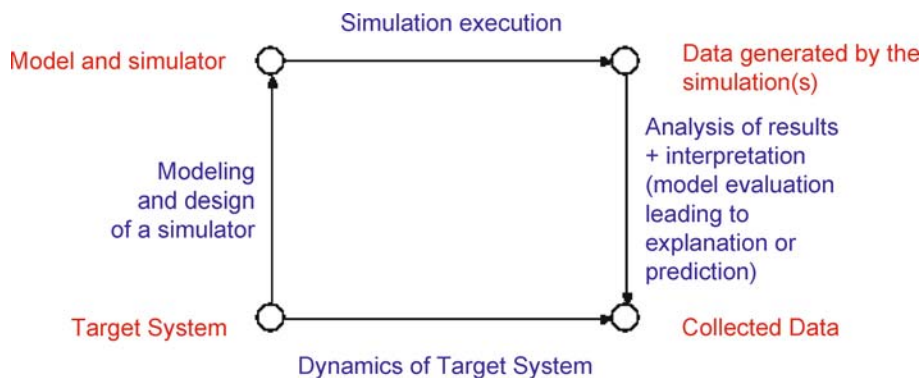
Computer simulation represents a way to exploit a computational model to evaluate designs and plans without actually bringing them into existence in the real world (e. g. architectural designs, road networks and traffic lights), but

also to evaluate theories and models of complex systems (e. g. biological or social systems) by envisioning the effect of the modeling choices, with the aim of gaining insight of their functioning. The use of these “synthetic environments” is sometimes necessary because the simulated system cannot actually be observed, since it is actually being designed or also for ethical or practical reasons. A general schema (based on several elaborations, such as those described in [19,31]) describing the role of simulation as a predictive or explanatory instrument is shown in Fig. 1.

Several situations are characterized by the presence of autonomous entities whose actions and interactions determine (in a non-trivial way) the evolution of the overall system. Agent-based models are particularly suited to represent these situations, and to support the study and analysis of topics like decentralized decision making, local-global interactions, self-organization, emergence, effects of heterogeneity in the simulated system. The interest in this relatively recent approach to modeling and simulation is demonstrated by the number of scientific events focused in this topic (see, to make some examples rooted in the computer science context, the Multi Agent Based Simulation workshop series [17,35,45,55,56,57], the IMA workshop on agent-based modeling¹ and the Agent-Based Modeling and Simulation symposium [7]). Agent-based models and multi-agent systems (MAS) have been adopted to simulate complex systems in very different contexts, ranging from social and economical simulation (see, e. g., [18]) to logistics optimization (see, e. g., [64]), from biological systems (see, e. g., [8]) to traffic (see, e. g., [4,12,60]) and crowd simulation (see, e. g., [11]).

This heterogeneity in the application domains also reflects the fact that, especially in this context of agent focused research, influences come from most different re-

¹<http://www.ima.umn.edu/complex/fall/agent.html>



Agent Based Modeling and Simulation, Figure 1

A general schema describing the usage of simulation as a predictive or explanatory instrument

search areas. Several *traffic and crowd* agent models, to make a relevant example, are deeply influenced by *physics*, and the related models provide agents that are modeled as particles subject to forces generated by the environment as well as by other agents (i.e. *active walker* models, such as [37]). Other approaches to crowd modeling and simulation build on experiences with *Cellular Automata* (CA) approaches (see, e.g., [54]) but provide a more clear separation between the environment and the entities that inhabit, act and interact in it (see, e.g., [6,38]). This line of research leads to the definition of models for situated MASs, a type of model that was also defined and successfully applied in the context of (reactive) robotics and control systems [61,62]. Models and simulators defined and developed in the context of social sciences [31] and economy [50] are instead based on different theories (often non-classical ones) of *human behavior* in order to gain further insight on it and help building and validating new theories.

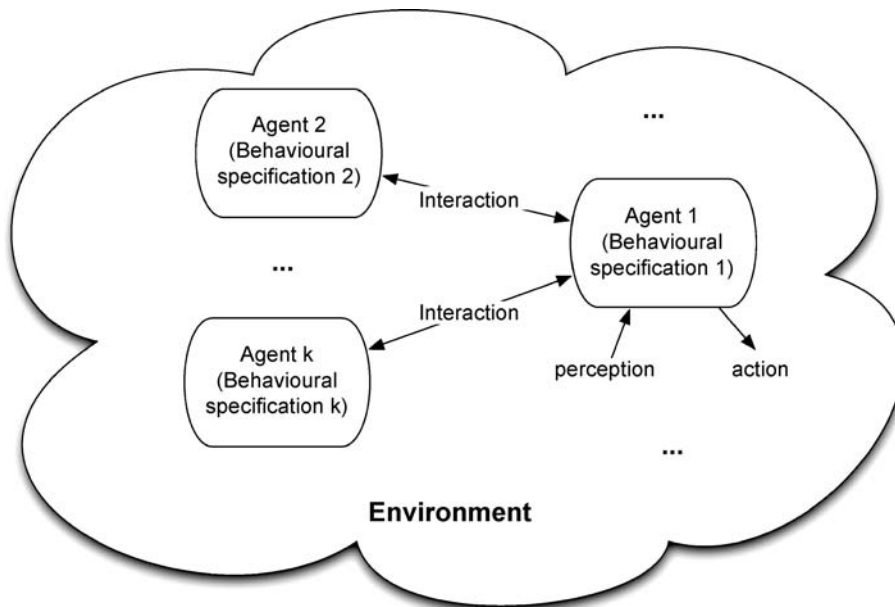
The common standpoint of all the above-mentioned approaches and of many other ones that describe themselves as agent-based is the fact that the analytical unit of the system is represented by the individual agent, acting and interacting with other entities in a shared environment: the overall system dynamic is not defined in terms of a global function, but rather the result of individuals' ac-

tions and interactions. On the other hand, it must also be noted that in most of the introduced application domains, the environment plays a prominent role because:

- It deeply influences the behaviors of the simulated entities, in terms of perceptions and allowed actions for the agents.
- The aim of the simulation is to observe some aggregate level behavior (e.g. the density of a certain type of agent in an area of the environment, the average length of a given path for mobile agents, the generation of clusters of agents), that can actually only be observed in the environment.

Besides these common elements, the above introduced approaches often dramatically differ in the way agents are described, both in terms of properties and behavior. A similar consideration can be done for their environment.

Considering the above introduced considerations, the aim of this article is not to present a specific technical contribution but rather to introduce an abstract reference model that can be applied to analyze, describe and discuss different models, concrete simulation experiences, platforms that, from different points of view, legitimately claim to adopt an agent-based approach. The reference model is illustrated in Fig. 2.



Agent Based Modeling and Simulation, Figure 2

An abstract reference model to analyze, describe and discuss different models, concrete simulation experiences, platforms legitimately claiming to adopt an agent-based approach

In particular, the following are the main elements of this reference model:

- *Agents* encompassing a possibly heterogeneous behavioral specification;
- Their *environment*, supplying agents their perceptions and enabling their actions;
- Mechanisms of *interaction* among agents.

Agent interaction, in fact, can be considered as a specific type of action having a central role in agent-based models. In the following Section, these elements will be analyzed, with reference to the literature on agent-based models and technologies but also considering the concrete experiences and results obtained by researches in other areas, from biology, to urban planning, to social sciences, in an attempt to enhance the mutual awareness of the points of contact of the respective efforts. Section “[Platforms for Agent-Based Simulation](#)” briefly discusses the available platforms supporting the rapid prototyping or development of agent-based simulations. A discussion on the future directions for this multi-disciplinary research area will end the article.

Agent-Based Models for Simulation

A model is an abstract and simplified representation of a given part of the world, either existing or planned (a target system, in the terminology adopted for Fig. 1). Models are commonly defined in order to study and explain observed phenomena or to forecast future phenomena. Agent-based models for simulation, as previously mentioned, are characterized by the presence of agents performing some kind of behavior in a shared environment. The notion of agent, however, is controversial even inside the restricted community of computer scientists dealing with research on agent models and technologies [25]. The most commonly adopted definition of agent [68] specifies a set of properties that must characterize an entity to effectively call it an agent, and in particular *autonomy* (the possibility to operate without intervention by humans, and a certain degree of control over its own state), *social ability* (the possibility to interact employing some kind of agent communication language, a notion that will be analyzed more in details in Subsect. “[Agent Interaction](#)”), *reactivity* (the possibility to perceive an environment in which it is situated and respond to perceived changes) and *pro-activeness* (the possibility to take the initiative, starting some activity according to internal goals rather than as a reaction to an external stimulus). This definition, considered by the authors a *weak* definition of agency, is generally already too restrictive to describe as agents most of the

entities populating agent-based models for simulation in different fields. Even if the distance between the context of research on *intelligent* agents and agent-based simulation, that is often more focused on the resulting behavior of the local action and interaction of relatively simple agents, cannot be neglected, the aim of this section is to present some relevant results of research on agent models and technologies in computer science and put them in relation with current research on agent-based simulation in other research areas.

As previously suggested, agent-based models (ABMs) can be considered models of complex systems and the ABM approach considers that simple and complex phenomena can be the result of interactions between autonomous and independent entities (i.e. agents) which operate within communities in accordance with different modes of interaction. Thus, agents and ABMs should not be considered simply as a technology [43,69] but also a modeling approach that can be exploited to represent some system properties that are not simply described as properties or functionalities of single system components but sometimes *emerge from collective behaviors* [22]. The study of such emerging behaviors is a typical activity in complex systems modeling [10] and agent-based models are growingly employed for the study of complex systems (see, e.g., [3,36,65] satellite workshops of the 2007 edition of the European Conference on Complex Systems).

Agent Behavior Specification

This section will first of all discuss the possible notions of *actions* available to an agent, then it will discuss the way an action actually chooses the actions to be carried out, introducing the notion of *agent architecture*.

Agent Actions – Actions are the elements at the basis of agent behavior. Agent actions can cause modifications in their environment or in other agents that constitutes the ABM. Different modeling solutions can be provided in order to describe agent actions: as *transformation of a global state*, as *response to influences*, as *computing processes*, as *local modification*, as *physical displacement*, and as *command* (more details about the reported methods to represent agent actions can be found in [22]).

- *Functional transformation of states* is based on concepts of states and state transformation and constitutes the most classical approach of artificial intelligence to action representation (mainly used in the planning and multi-agent planning contexts). According to this approach, agent actions are defined as operators whose effect is a change in the state of the world [24,29,30].

- Modeling *actions as local modification* provides an approach opposite to the one based on transformation of a global state. Agents perceive their local environment (the only part of the environment that they can access) and according to their perceptions, they modify their internal state. Automata networks [32] and cellular automata [67] are examples of this type of models. They are dynamical systems whose behavior is defined in terms of local relationships and local transformations; in turn, these changes influence the overall system state. Within the ABMs context cellular automata are often exploited to represent the dynamic behavior of agent environment [58] or to simulate population dynamics [21] in artificial life [1,41].
- Modeling *actions as response to influences* [23] extends the previous approach introducing elements to consider the effects of agent interactions and the simultaneous execution of actions in an ABM. Agent actions are conditioned and represent a reaction to other agent actions or to environment modifications.
- Agents can also be considered *as computational processes* (in the vein of the actor model [2]). A computing system, and thus a ABM, can be considered as a set of activities (i. e. processes) that are executed sequentially or in parallel. This modeling approach in particular focuses on single computing entities, their behaviors and interactions. The most studied and well known methods to represent processes are finite state automata, Petri nets and their variants [46].
- *Approaches derived from physics* can also be found in order to represent agent actions. In these cases, actions mainly concern movements and their applications are in the robotics contexts (i. e. reactive and situated agents) [42]. One of the most used notion derived from physics is the one of *field* (e. g. gravitational, electrical, magnetic). Agents are attracted or repelled by given objects or environment areas that emit fields to indicate their position.
- According to cybernetics and control system theory, actions can be represented *as system commands* that regulate and control agent behavior. In this way, actions are complex tasks that the agent executes in order to fulfill given goals and that take into account the environment reactions and correctives to previous actions.

Agent Architecture – The term *architecture* [53] refers to the model of agent internal structure that is responsible of effectively selecting the actions to be carried out, according to the perceptions and internal state of an agent. Different architectures have been proposed in order to obtain specific agent behaviors and they are generally classified

into *deliberative* and *reactive* (respectively, *hysteretic* and *tropistic* according to the classification reported in [29]).

Reactive agents are elementary (and often memory-less) agents with a defined position in the environment. Reactive agents perform their actions as a consequence of the perception of stimuli coming either from other agents or from the environment; generally, the behavioral specification of this kind of agent is a set of condition-action rules, with the addition of a selection strategy for choosing an action to be carried out whenever more rules could be activated. In this case, the motivation for an action derives from a triggering event detected in the environment; these agents cannot be pro-active.

Deliberative or *cognitive* agents, instead, are characterized by a more complex action selection mechanism, and their behavior is based on so called mental states, on facts representing agent knowledge about the environment and, possibly, also on memories of past experiences. Deliberative agents, for every possible sequence of perceptions, try to select a sequence of actions, allowing them to achieve a given goal. Deliberative models, usually defined within the planning context, provide a symbolic and explicit representation of the world within agents and their decisions are based on logic reasoning and symbol manipulation. The BDI model (belief, desire, intention [51,52]) is perhaps the most widespread model for deliberative agents. The internal state of agents is composed of three “data structures” concerning agent beliefs, desires and intentions. Beliefs represent agent information about its surrounding world, desires are the agent goals, while intentions represent the desire an agent has effectively selected, that it has to some extent committed.

Hybrid architectures can also be defined by combining the previous ones. Agents can have a layered architecture, where deliberative layers are based on a symbolic representation of the surrounding world, generate plans and take decisions, while reactive layers perform actions as effect of the perception of external stimuli. Both vertical and horizontal architectures have been proposed in order to structure layers [13]. In horizontal architecture no priorities are associated to layers and the results of the different layers must be combined to obtain agent’s behavior. When layers are instead arranged in a vertical structure, reactive layers have higher priority over deliberative ones, that are activated only when no reactive behavior is triggered by the perception of an external stimulus.

A MAS can be composed of cognitive agents (generally a relatively low number of deliberative agents), each one possessing its own knowledge-model determining its behavior and its interactions with other agents and the environment. By contrast, there could be MAS made only by

reactive agents. This type of system is based on the idea that it is not necessary for a single agent to be individually intelligent for the system to demonstrate complex (intelligent) behaviors. Systems of reactive agents are usually more robust and fault tolerant than other agent-based systems (e.g. an agent may be lost without any catastrophic effect for the system). Other benefits of reactive MAS include flexibility and adaptability in contrast to the inflexibility that sometimes characterizes systems of deliberative agents [14]. Finally, a system might also present an *heterogeneous* composition of reactive and deliberative agents.

Environment

Weyns et al. in [66] provide a definition of the notion of environment for MASs (and thus of an environment for an ABM), and also discuss the core responsibilities that can be ascribed to it. In particular, in the specific context of simulation the environment is typically responsible for the following:

- Reflecting/reifying/managing the structure of the physical/social arrangement of the overall system;
- Embedding, supporting regulated access to objects and parts of the system that are not modeled as agents;
- Supporting agent perception and situated action (it must be noted that agent interaction should be considered a particular kind of action);
- Maintain internal dynamics (e.g. spontaneous growth of resources, dissipation signals emitted by agents);
- Define/enforce rules.

In order to exemplify this schema, we will now consider agent-based models and simulators that are based on a physical approach; the latter generally consider agents as particles subject to and generating forces. In this case, the environment comprises laws regulating these influences and relevant elements of the simulated system that are not agents (e.g. point of reference that generate attraction/repulsion forces). It is the environment that determines the overall dynamics, combining the effects that influence each agent and applying them generally in discrete time steps. In this cycle, it captures all the above introduced responsibilities, and the role of agents is minimal (according to some definitions they should not be called agents at all), and running a simulation is essentially reduced to computing iteratively a set equations (see, e.g., [4,37]). In situated ABM approaches agents have a higher degree of autonomy and control over their actions, since they evaluate their perceptions and choose their actions according to their behavioral specification. The environment retains a very relevant role, since it pro-

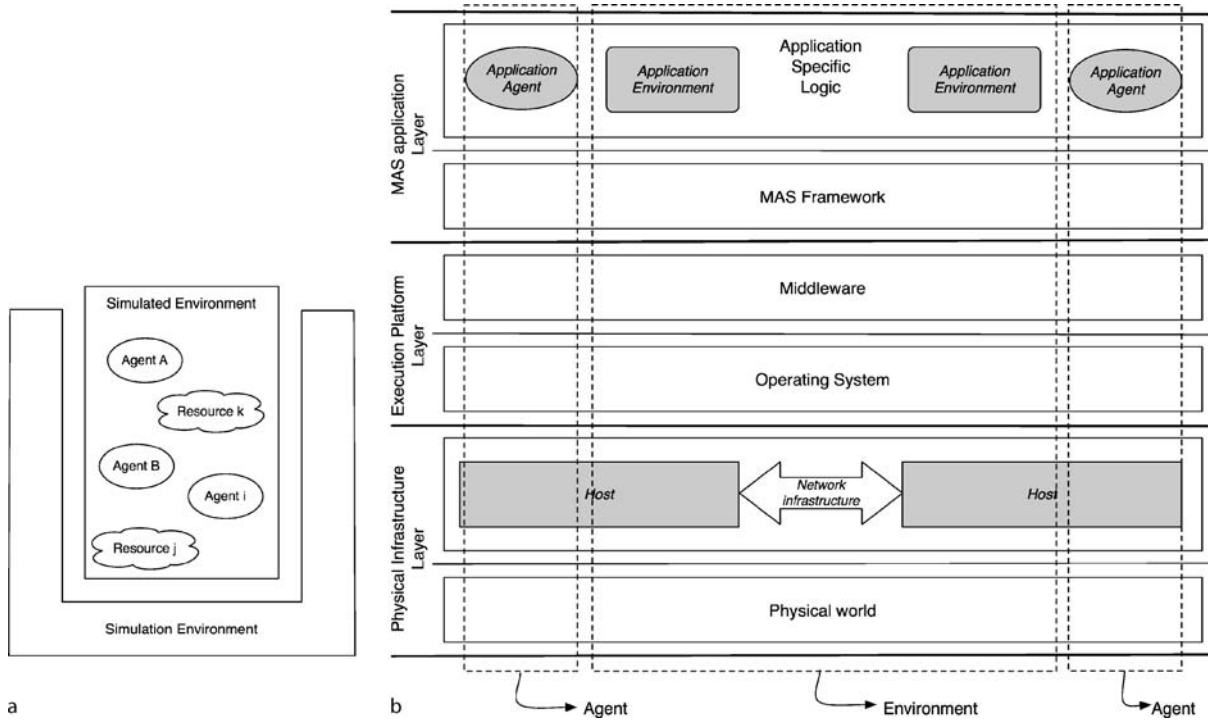
vides agents with their perceptions that are generated according to the current structure of the system and to the arrangement of agents situated in it. Socioeconomic models and simulations provide various approaches to the representation of the simulated system, but are generally similar to situated ABMs.

It is now necessary to make a clarification on how the notion of environment in the context of MAS-based simulation can be turned into a software architecture. Klügl et al. [40] argue that the notion of environment in multi-agent simulation is actually made up of two conceptually different elements: the simulated environment and the simulation environment. The former is a part of the computational model that represents the reality or the abstraction that is the object of the simulation activity. The simulation environment, on the other hand, is a software infrastructure for executing the simulation. In this framework, to make an explicit decoupling between these levels is a prerequisite for good engineering practice. It must be noted that also a different work [33], non-specifically developed in the context of agent-based simulation, provided a model for the deployment environment, that is the specific part of the software infrastructure that manages the interactions among agents.

Another recent work is focused on clarifying the notion of ABM environment and describes a three layered model for situated ABM environments [63]. This work argues that environmental abstractions (as well as those related to agents) crosscut all the system levels, from application specific ones, to the execution platform, to the physical infrastructure. There are thus application specific aspects of agents' environment that must be supported by the software infrastructure supporting the execution of the ABM, and in particular the ABM framework (MAS framework in the figure). Figure 3 compares the two above-described schemas.

The fact that the environment actually crosscuts all system levels in a deployment model represents a problem making difficult the separation between simulated environment and simulation infrastructure. In fact, the modeling choices can have a deep influence on the design of the underlying ABM framework and, vice versa, design choices on the simulation infrastructure make it suitable for some ABM and environment models but not usable for other ones. As a result, general ABM framework supporting simulation actually exist, but they cannot offer a specific form of support to the modeler, although they can offer basic mechanisms and abstractions.

SeSAm, for instance, offers a general simulation infrastructure but relies on plugins [40]. Those plugins, for example, could be used to define and manage the spatial



Agent Based Modeling and Simulation, Figure 3

A schema introduced in [40] to show differences and relationships between simulated and simulation environment (a), and a three layer deployment model for situated MAS introduced in [63] highlighting the crosscutting abstractions *agent* and *environment* (b)

features of the simulated environment, including the associated basic functions supporting agent movement and perception in that kind of environment. With reference to Fig. 3b, such a plugin would be associated to the application environment module, in the ABM application layer. However, these aspects represent just some of the features of the simulated environment, that can actually comprise rules and laws that extend their influence over the agents and the outcomes of their attempts to act in the environment.

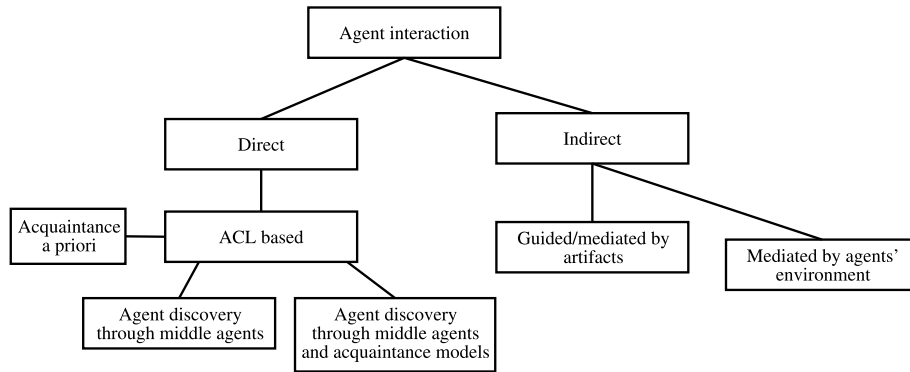
Agent Interaction

Interaction is a key aspect in ABM. There is a plethora of definitions for the concept of agent and most of them emphasize the fact that this kind of entity should be able to interact with their environment and with other entities in order to solve problems or simply reach their goals according to coordination, cooperation or competition schemes. The essence of a ABM is the fact that the global system dynamics emerges from the local behaviors and interactions among its composing parts. Strictly speaking, for some kind of ABM the global dynamics is just the sum of lo-

cal behaviors and interactions, so we cannot always speak of emergent behavior when we talk about a ABM. However the assumptions that underlie the design of an interaction model (or the choice of an existing one for the design and implementation of a specific application) are so important that they have a deep impact on the definition of agent themselves (e. g. an interpreter of a specific language, a perceiver of signals). Therefore it is almost an obvious consequence that interaction mechanisms have a huge impact on the modeling, design and development of applications based on a specific kind of ABM, which, in turn, is based on a particular interaction model. It is thus not a surprise that a significative part of the research that was carried out in the agent area was focused on this aspect.

This section presents a conceptual taxonomy of currently known/available agent interaction models, trying to define advantages and issues related to them, both from a conceptual and a technical point of view.

There are many possible dimensions and aspects of agent interaction models that can be chosen and adopted in order to define a possible taxonomy. The first aspect that is here considered to classify agent interaction models is related to the fact that agents communicate di-



Agent Based Modeling and Simulation, Figure 4
The proposed taxonomy of agent interaction models

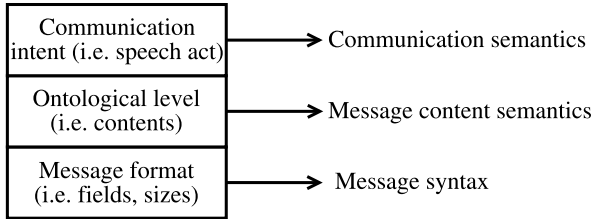
rectly (for instance exchanging messages), and thus the model does not include an abstraction of the actual communication channel, or there are some media interposed among the communication partners which is explicitly included in the interaction model. While the former approach, with specific reference to Agent Communication Language (ACL)-based models, is the most widely adopted in the agent area it has its drawbacks, and most of them are related to the issue of agents acquaintance. The way ACL-based agent interaction models deal with this issue is the subject of another dimension of the taxonomy, providing direct a priori acquaintance among agents, the adoption of middle agents for information discovery and the development of more complex acquaintance models to tackle issues related to the representation and maintenance of acquaintance information but also to robustness and scalability of the agent infrastructure. However there are other agent interaction models providing an indirect communication among them. Some of these approaches provide the creation and exploitation of artifacts that represent a medium for agents' interaction. Other indirect interaction models are more focused on modeling agent environment as the place where agent interactions take place, thus influencing interactions and agent behavior.

Direct Interaction Models The first and most widely adopted kind of agent interaction model provide a direct information exchange between communication partners. This approach ignores the details related to the communication channel that allows the interaction, and does not include it as an element of the abstract interaction model. Generally the related mechanisms provide a point-to-point message-passing protocol regulating the exchange of messages among agents. There are various aspects of the communicative act that must be modeled (ranging from

low-level technical considerations on message format to conceptual issues related to the formal semantics of messages and conversations), but generally this approach provides the definition of suitable languages to cover these aspects. While this approach is generally well-understood and can be implemented in a very effective way (especially as it is substantially based on the vast experience of computer networks protocols), in the agent context, it requires specific architectural and conceptual solutions to tackle issues related to the agent acquaintance/discovery and ontological issues.

Intuitively an Agent Communication Language (ACL) provides agents with a means of exchanging information and knowledge. This vague definition inherently includes the point of view on the conception of the term agent, which assumes that an agent is an intelligent autonomous entity whose features include some sort of *social ability* [68]. According to some approaches this kind of feature is the one that ultimately defines the essence of agency [28]. Leaving aside the discussion on the definition and conception of agency, this section will focus on what the expression “social ability” effectively means. To do so we will briefly compare these ACL share with those approaches that allow the exchange of information among distributed components (e.g. in legacy systems²) some basic issues: in particular, the definition of a communication channel allowing the reliable exchange of messages over a computer network (i.e. the lower level aspects of the communication). What distinguishes ACLs from such systems are the objects of discourse and their semantic complexity; in particular there are two aspects which distributed computing protocols and architectures do not have to deal with:

²With this expression we mean pieces of software which are not designed to interact with agents and agent based systems.



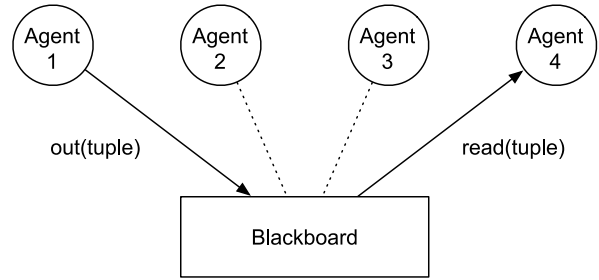
Agent Based Modeling and Simulation, Figure 5
Layers and concerns of an agent communication language

- *Autonomy* of interacting components: modern systems' components (even though they can be quite complex and can be considered as self-sufficient with reference to supplying a specific service) have a lower degree of autonomy than the one that is generally associated to agents.
- The information conveyed in messages does not generally require a comprehensive *ontological approach*, as structures and categories can be considered to be shared by system components.

Regarding the autonomy, while traditional software components offer services and generally perform the required actions as a reaction to the external requests, agents may decide not to carry out a task that was required by some other system entity. Moreover generally agents are considered temporally continuous and proactive, while this is not generally true for common software components.

For what concerns the second point, generally components have specific interfaces which assume an agreement on a set of shared data structures. The semantics of the related information, and the semantics of messages/method invocation/service requests, is generally given on some kind of (more or less formally specified) modeling language, but is tightly related to component implementation. For agent interaction a more explicit and comprehensive view on domain concepts must be specified. In order to be able to effectively exchange knowledge, agents must share an *ontology* (see, e.g., [34]), that is a representation of a set of categories of objects, concepts, entities, properties and relations among them. In other words, the same concept, object or entity must have a uniform meaning and set of properties across the whole system.

Indirect Interaction Models From a strictly technical point of view, agent communication is generally indirect even in direct agent interaction models. In fact most of these approaches adopt some kind of communication infrastructure supplying a reliable end-to-end message passing mechanism. Nonetheless, the adoption of a *conceptu-*



Agent Based Modeling and Simulation, Figure 6

A conceptual diagram for a typical blackboard architecture, including two sample primitives of the Linda coordination model, that is, the output of a tuple into the blackboard (the *out* operation) and the non-destructive input of an agent from the blackboard (the *read* operation)

ally direct agent interaction model brings specific issues that were previously introduced. The remaining of this section will focus on models providing the presence of an intermediate entity mediating (allowing and regulating) agent interaction. This communication abstraction is not merely a low-level implementation detail, but a first-class concept of the model.

Agent interaction models which provide indirect mechanisms of communication will be classified into *artifact mediated* and *spatially grounded* models. The distinction is based on the inspiration and metaphor on which these models are rooted. The former provide the design and implementation of an artifact which emulates concrete objects of agents' environment whose goal is the communication of autonomous entities. Spatially grounded agent interaction models bring the metaphor of modeling agent environment to the extreme, recognizing that there are situations in which spatial features and information represent a key factor and cannot be neglected in analyzing and modeling a system.

Both of these approaches provide interaction mechanisms that are deeply different from point-to-point message exchange among entities. In fact, the media which enable the interaction intrinsically represent a *context* influencing agent communication.

In the real world a number of physical agents interact sharing resources, by having a competitive access to them (e.g. cars in streets and crossroads), but also collaborating in order to perform tasks which could not be carried out by single entities alone, due to insufficient competencies or abilities (e.g. people that carry a very heavy burden together). Very often, in order to regulate the interactions related to these resources, we build concrete artifacts, such as traffic lights on the streets, or neatly placed

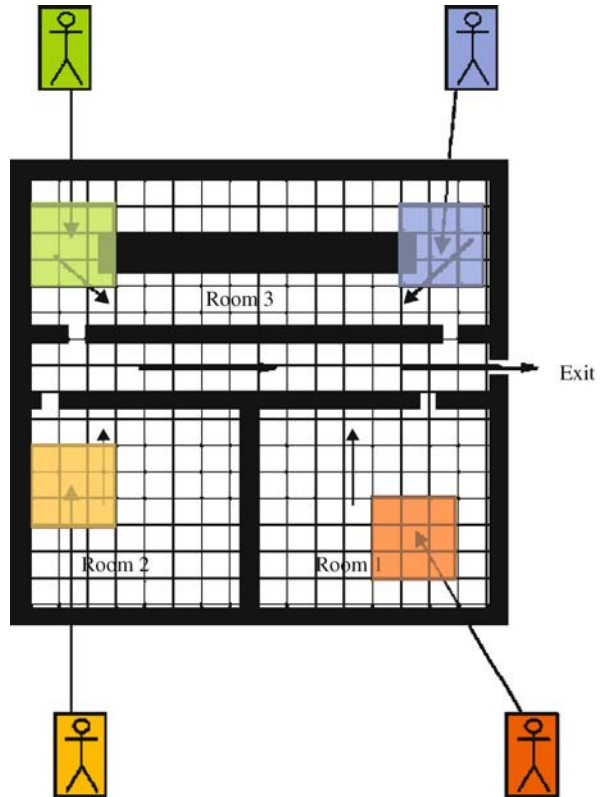
handles on large heavy boxes. Exploiting this metaphor, some approaches to agent interaction tend to model and implement abstractions allowing the cooperation of entities through a shared resource, whose access is regulated according to precisely defined rules. *Blackboard-based architectures* are the first examples of this kind of models. A blackboard is a shared data repository that enables co-operating software modules to communicate in an indirect and anonymous way [20]. In particular the concept of *tuple space*, first introduced in Linda [26], represents a pervasive modification to the basic blackboard model.

Linda [26] coordination language probably represents the most relevant blackboard-based model. It is based on the concept of tuple space, that is an associative blackboard allowing agents to share and retrieve data (i.e. tuples) through some data-matching mechanism (such as pattern-matching or unification) integrated within the blackboard. Linda also defines a very simple language defining mechanisms for accessing the tuple space.

The rationale of this approach is to keep separated computation and coordination contexts as much as possible [27], by providing specific abstractions for agent interaction. With respect to direct interaction models, part of the burden of coordination is in fact moved from the agent to the infrastructure. The evolution of this approach has basically followed two directions: the extension of the coordination language and infrastructure in order to increase its expressiveness or usability, and the modeling and implementation of distributed tuple spaces [16,48,49].

While the previously described indirect approaches define artifacts for agent interaction taking inspiration from actual concrete object of the real world, other approaches bring the metaphor of agent environment to the extreme by taking into account its spatial features.

In these approaches agents are situated in an environment whose spatial features are represented possibly in an explicit way and have an influence on their perception, interaction and thus on their behavior. The concept of perception, which is really abstract and metaphoric in direct interaction models and has little to do with the physical world (agents essentially perceive their state of mind, which includes the effect of received messages, like new facts in their knowledge base), here is related to a more direct modeling of what is often referred to as “local point of view”. In fact these approaches provide the implementation of an infrastructure for agent communication which allows them to perceive the state of the environment in their position (and possibly in nearby locations). They can also cause local modifications to the state of the environment, generally through the emission of signals, emulating some kind of physical phenomenon



Agent Based Modeling and Simulation, Figure 7

A sample schema exemplifying an environment mediated form of interaction in which the spatial structure of the environment has a central role in determining agents' perceptions and their possibility to interact

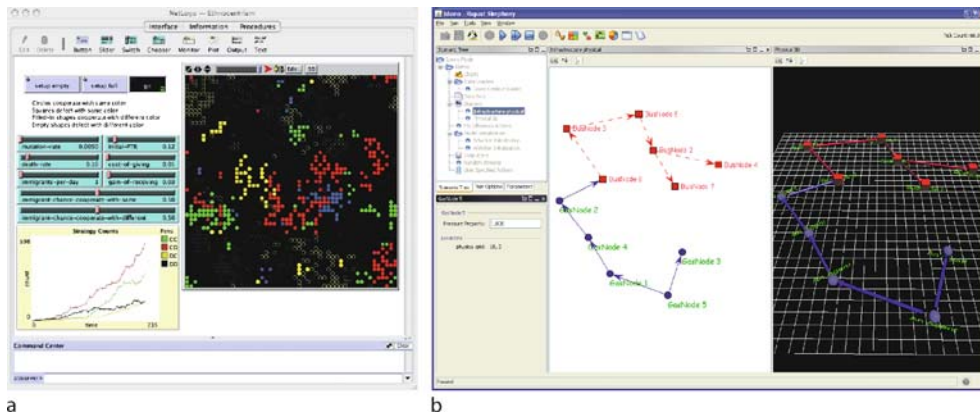
(e.g. pheromones [15], or fields [5,44]) or also by simply observing the actions of other agents and reacting to this perception, in a “behavioral implicit communication” schema [59].

In all these cases, however, the structuring function of the environment is central, since it actually defines what can be perceived by an agent in its current position and how it can actually modify the environment, to which extent its actions can be noted by other agents and thus interact with them.

Platforms for Agent-Based Simulation

Considering the pervasive diffusion and adoption of agent-based approaches to modeling and simulation, it is not surprising the fact that there is a growing interest in software frameworks specifically aimed at supporting the realization of agent-based simulation systems.

This kind of framework often provides abstractions and mechanisms for the definition of agents and their en-



Agent Based Modeling and Simulation, Figure 8

A screenshot of a NetLogo simulation applet, (a), and a Repast simulation model, (b)

vironments, to support their interaction, but also additional functionalities like the management of the simulation (e.g. set-up, configuration, turn management), its visualization, monitoring and the acquisition of data about the simulated dynamics. It is not the aim of this article to provide a detailed review of the current state of the art in this sector, but rather sketch some classes of instruments that have been used to support the realization of agent-based simulations and provide a set of references to relevant examples of platforms facilitating the development of agent-based simulations.

A first category of these platforms instruments provides *general purpose frameworks* in which agents mainly represent *passive abstractions*, sort of data structures that are manipulated by an overall simulation process. A relevant example of such tools is **NetLogo**³, a dialect of the Logo language specifically aimed at modeling phenomena characterized by a decentralized, interconnected nature. NetLogo does not even adopt the term agent to denote individuals, but it rather calls them *turtles*; a typical simulation consists in a cycle choosing and performing an action for every turtle, considering its current situation and state. It must be noted that, considering some of the previously mentioned definitions of autonomous agent a turtle should not be considered an agent, due to the almost absent autonomy of these entities. The choice of a very simple programming language that does not require a background on informatics, the possibility to deploy in a very simple way simulations as Java applets, and the availability of simple yet effective visualization tools, made NetLogo extremely popular.

³<http://ccl.northwestern.edu/netlogo/>

A second category of platforms provides frameworks that are developed with a similar rationale, providing for very similar support tools, but these instruments are based on *general purpose programming languages* (generally object oriented). **Repast**⁴ [47] represents a successful representative of this category, being a widely employed agent-based simulation platform based on the Java language. The object-oriented nature of the underlying programming language supports the definition of computational elements that make these agents more autonomous, closer to the common definitions of agents, supporting the encapsulation of state (and state manipulation mechanisms), actions and action choice mechanism in agent's class. The choice of adopting a general purpose programming language, on one hand, makes the adoption of these instruments harder for modelers without a background in informatics but, on the other, it simplifies the integration with external and existing libraries. Repast, in its current version, can be easily connected to instruments for statistical analysis, data visualization, reporting and also geographic information systems.

While the above-mentioned functionalities are surely important in simplifying the development of an effective simulator, and even if in principle it is possible to adapt frameworks belonging to the previously described categories, it must be noted that their neutrality with respect to the specific adopted agent model leads to a necessary preliminary phase of adaptation of the platform to the specific features of the model that is being implemented. If the latter defines specific abstractions and mechanisms for agents, their decision-making activities, their environ-

⁴<http://repast.sourceforge.net/>

ment and the way they interact, then the modeler must in general develop proper computational supports to be able to fruitfully employ the platform. These platforms, in fact, are not endowed with specific supports to the realization of agent deliberation mechanisms or infrastructures for interaction models, either direct or indirect (even if it must be noted that all the above platforms generally provide some form of support to agent environment definition, such as grid-like or graph structures).

A third category of platforms represent an attempt to provide a *higher level linguistic support* trying to reduce the distance between agent-based models and their implementations. The latest version of Repast, for instance, is characterized by the presence of a high level interface for “point-and-click” definition of agent’s behaviors, that is based on a set of primitives for the specification of agent’s actions. **SimSesam**⁵ [39] defines a set of *primitive functions* as basic elements for describing agents’ behaviors, and it also provides visual tools supporting model implementation. At the extreme border of this category, we can mention efforts that are specifically aimed at supporting the development of simulations based on a precise agent model, approach and sometimes even for a specific area of application, such as the one described in [9,64].

Future Directions

Agent-based modeling and simulation is a relatively young yet already widely diffused approach to the analysis, modeling and simulation of complex systems. The heterogeneity of the approaches, modeling styles and applications that legitimately claim to be “agent-based”, as well as the fact that different disciplines are involved in the related research efforts, they are all factors that hindered the definition of a generally recognized view of the field. A higher level framework of this kind of activity would be desirable in order to relate different efforts by means of a shared schema. Moreover it could represent the first step in effectively facing some of the epistemological issues related to this approach to the modeling and analysis of complex systems. The future directions in this broad research area are thus naturally aimed at obtaining vertical analytical results on specific application domains, but they must also include efforts aimed at “building bridges” between the single disciplinary results in the attempt to reach a more general and shared understanding of how these bottom-up modeling approaches can be effectively employed to study, explain and (maybe) predict the overall behavior of complex systems.

Bibliography

1. Adami C (1998) Introduction to Artificial Life. Springer, New York
2. Agha G (1986) Actors: A Model of Concurrent Computation in Distributed Systems. MIT press, Cambridge
3. Alfi V, Galla T, Marsili M, Pietronero L (eds) (2007) Interacting Agents, Complexity and Inter-Disciplinary Applications (IACIA)
4. Balmer M, Nagel K (2006) Shape morphing of intersection layouts using curb side oriented driver simulation. In: van Leeuwen JP, Timmermans HJ (eds) Innovations in Design & Decision Support Systems in Architecture and Urban Planning. Springer, Dordrecht, pp 167–183
5. Bandini S, Manzoni S, Simone C (2002) Heterogeneous agents situated in heterogeneous spaces. Appl Artif Intell 16:831–852
6. Bandini S, Manzoni S, Vizzari G (2004) Situated cellular agents: a model to simulate crowding dynamics. IEICE Transactions on Information and Systems: Special Issues on Cellular Automata E87-D, pp 669–676
7. Bandini S, Petta P, Vizzari G (eds) (2006) International Symposium on Agent Based Modeling and Simulation (ABModSim 2006), vol Cybernetics and Systems. Austrian Society for Cybernetic Studies (2006) 18th European Meeting on Cybernetics and Systems Research (EMCSR 2006)
8. Bandini S, Celada F, Manzoni S, Puzone R, Vizzari G (2006) Modelling the immune system with situated agents. In: Apolloni B, Marinaro M, Nicosia G, Tagliaferri R (eds) Proceedings of WIRN/NAIS 2005. Lecture Notes in Computer Science, vol 3931. Springer, Berlin, pp 231–243
9. Bandini S, Federici ML, Vizzari G (2007) Situated cellular agents approach to crowd modeling and simulation. Cybernet Syst 38:729–753
10. Bar-Yam Y (1997) Dynamics of Complex Systems. Addison-Wesley, Reading
11. Batty M (2001) Agent based pedestrian modeling. Env Plan B: Plan Des 28:321–326
12. Bazzan ALC, Wahle J, Klügl F (1999) Agents in traffic modelling – from reactive to social behaviour. In: Burgard W, Christaller T, Cremers AB (eds) KI-99: Advances in Artificial Intelligence, 23rd Annual German Conference on Artificial Intelligence, Bonn, Germany, 13–15 September 1999. Lecture Notes in Computer Science, vol 1701. Springer, Berlin, pp 303–306
13. Brooks RA (1986) A robust layered control system for a mobile robot. IEEE J Robot Autom 2:14–23
14. Brooks RA (1990) Elephants don’t play chess. Robot Autonom Syst 6:3–15
15. Brueckner S (2000) An analytic approach to pheromone-based coordination. In: ICMAS IEEE Comp Soc, pp 369–370
16. Cabri G, Leonardi L, Zambonelli F (2000) MARS: a programmable coordination architecture for mobile agents. IEEE Inter Comp 4:26–35
17. Davidsson P, Logan B, Takadama K (eds) (2005) Multi-Agent and Multi-Agent-Based Simulation, Joint Workshop MABS (2004), New York, 19 July 2004, Revised Selected Papers. In: Davidsson P, Logan B, Takadama K (eds) MABS. Lecture Notes in Computer Science, vol 3415. Springer, Berlin
18. Dosi G, Fagiolo G, Roventini A (2006) An evolutionary model of endogenous business cycles. Comput Econ 27:3–34
19. Edmonds B (2001) The use of models – making MABS more informative. In: Multi-Agent-Based Simulation, Second Interna-

⁵<http://www.simsesam.de/>

- tional Workshop MABS (2000), Boston MA, USA, July (2000), Revised and Additional Papers. Lecture Notes in Computer Science, vol 1979. Springer, Berlin, pp 15–32
20. Englemore RS, Morgan T (eds) (1988) Blackboard Systems. Addison-Wesley, Reading
 21. Epstein JM, Axtell R (1996) Growing Artificial Societies. MIT Press, Boston
 22. Ferber J (1999) Multi-Agent Systems. Addison-Wesley, London
 23. Ferber J, Muller J (1996) Influences and reaction: A model of situated multiagent systems. In: Proceedings of the 2nd International Conference on Multiagent Systems
 24. Fikes RE, Nilsson NJ (1971) STRIPS: a new approach to the application of theorem proving to problem solving. *Artif Intell* 2:189–208
 25. Franklin S, Graesser A (1997) Is it an agent, or just a program?: A taxonomy for autonomous agents. In: Müller JP, Wooldridge M, Jennings NR (eds) Intelligent Agents III, Agent Theories, Architectures, and Languages ECAI '96 Workshop (ATAL), Budapest, 12–13 August 1996. Lecture Notes in Computer Science, vol 1193. Springer, Berlin, pp 21–36
 26. Gelernter D (1985) Generative communication in Linda. *ACM Trans Program Lang Syst* 7:80–112
 27. Gelernter D, Carriero N (1992) Coordination languages and their significance. *Commun ACM* 35:97–107
 28. Genesereth MR, Ketchpel SP (1994) Software agents. *Commun ACM* 37(7):48–53
 29. Genesereth MR, Nilsson N (1987) Logical Foundations of Artificial Intelligence. Morgan Kaufmann, San Mateo
 30. Georgeff M (1984) A theory of action in multi-agent planning. In: Proceedings of the AAAI84, pp 121–125
 31. Gilbert N, Troitzsch KG (2005) Simulation for the Social Scientist 2nd edn. Open University Press, Maidenhead
 32. Goles E, Martinez S (1990) Neural and automata networks, dynamical behavior and applications. Kluwer, Norwell
 33. Gouaich A, Michel F, Guiraud Y (2005) MIC: A deployment environment for autonomous agents. In: Environments for Multi-Agent Systems, First International Workshop (E4MAS 2004). Lecture Notes in Computer Science, vol 3374. Springer, Berlin, pp 109–126
 34. Gruber TR (1995) Toward principles for the design of ontologies used for knowledge sharing. *Int J Hum Comp Stud* 43:907–928
 35. Hales D, Edmonds B, Norling E, Rouchier J (eds) (2003) Multi-Agent-Based Simulation III, 4th International Workshop MABS (2003), Melbourne, 14 July 2003. Revised Papers. In: Hales D, Edmonds B, Norling E, Rouchier J (eds) MABS. Lecture Notes in Computer Science, vol 2927. Springer, Berlin
 36. Hassas S, Serugendo GDM, Phan D (eds) (2007) Multi-Agent for modelling Complex Systems (MA4CS). <http://bat710.univ-lyon1.fr/~farmetta/MA4CS07>
 37. Helbing D, Schweitzer F, Keltsch J, Molnár P (1997) Active walker model for the formation of human and animal trail systems. *Phys Rev E* 56:2527–2539
 38. Henein CM, White T (2005) Agent-based modelling of forces in crowds. In: Davidsson P, Logan B, Takadama K (eds) Multi-Agent and Multi-Agent-Based Simulation, Joint Workshop MABS (2004), New York 19 July 2004. Revised Selected Papers. Lecture Notes in Computer Science, vol 3415. Springer, Berlin, pp 173–184
 39. Klügl F, Herrler R, Oechslein C (2003) From simulated to real environments: How to use sesam for software development. In: Schillo M, Klusch M, Müller JP, Tianfield H (eds) MATES. Lecture Notes in Computer Science, vol 2831. Springer, Berlin, pp 13–24
 40. Klügl F, Fehler M, Herrler R (2005) About the role of the environment in multi-agent simulations. In: Weyns D, Parunak HVD, Michel F (eds) Environments for Multi-Agent Systems, First International Workshop E4MAS (2004), New York 19 July 2004. Revised Selected Papers. vol 3374, pp 127–149
 41. Langton C (1995) Artificial life: An overview. MIT Press, Cambridge
 42. Latombe JC (1991) Robot Motion Planning. Kluwer, Boston
 43. Luck M, McBurney P, Sheory O, Willmott S (eds) (2005) Agent Technology: Computing as Interaction. University of Southampton, Southampton
 44. Mamei M, Zambonelli F, Leonardi L (2002) Co-fields: towards a unifying approach to the engineering of swarm intelligent systems. In: Engineering Societies in the Agents World III: Third International Workshop (ESAW 2002). Lecture Notes in Artificial Intelligence, vol 2577. Springer, Berlin, pp 68–81
 45. Moss S, Davidsson P (eds) (2001) Multi-Agent-Based Simulation, Second International Workshop MABS (2000), Boston, July, (2000), Revised and Additional Papers. Lecture Notes in Computer Science, vol 1979. Springer, Berlin
 46. Murata T (1989) Petri nets: properties, analysis and applications. *Proc IEEE* 77:541–580
 47. North MJ, Collier NT, Vos JR (2006) Experiences creating three implementations of the repast agent modeling toolkit. *ACM Trans Model Comp Sim* 16:1–25
 48. Omicini A, Zambonelli F (1999) Coordination for Internet application development. *Autono Agents Multi-Agent Syst* 2:251–269 Special Issue: Coordination Mechanisms for Web Agents
 49. Picco GP, Murphy AL, Roman GC (1999) Lime: Linda meets mobility. In: Proceedings of the 21st International Conference on Software Engineering (ICSE 99) ACM Press, Los Angeles, pp 368–377
 50. Pyka A, Fagiolo G (2007) Agent-Based Modelling: A Methodology for Neo-Schumpeterian Economics. In: Hanusch H, Pyka A (eds) Elgar Companion to Neo-Schumpeterian Economics. Edward Elgar Publishing, pp 467–487
 51. Rao A, Georgeff M (1991) Modeling rational agents within a BDI-architecture. In: Proc Knowledge Representation and Reasoning (KR&R 1991)
 52. Rao A, Georgeff M (1995) BDI agents: from theory to practice. In: Proceedings of the International Conference on Multi-Agent Systems
 53. Russel S, Norvig P (1995) Artificial Intelligence: A Modern Approach. Prentice Hall, Upper Saddle River
 54. Schadschneider A, Kirchner A, Nishinari K (2002) CA approach to collective phenomena in pedestrian dynamics. In: Bandini S, Chopard B, Tomassini M (eds) Cellular Automata, 5th International Conference on Cellular Automata for Research and Industry ACRI 2002. Lecture Notes in Computer Science, vol 2493. Springer, Berlin, pp 239–248
 55. Sichman JS, Antunes L (eds) (2006) Multi-Agent-Based Simulation VI, International Workshop MABS (2005), Utrecht, The Netherlands, 25 July 2005, Revised and Invited Papers. In: Sichman JS, Antunes L (eds) MABS. Lecture Notes in Computer Science, vol 3891. Springer, Berlin
 56. Sichman JS, Conte R, Gilbert N (eds) (1998) Multi-Agent Systems and Agent-Based Simulation, First International Work-

- shop MABS '98, Paris, France, 4–6 July 1998. Proceedings. In: Sichman JS, Conte R, Gilbert N (eds): MABS. Lecture Notes in Computer Science, vol 1534. Springer, Berlin
57. Sichman JS, Bousquet F, Davidsson P (eds) (2003) Multi-Agent-Based Simulation, Third International Workshop MABS (2002), Bologna, Italy, 15–16 July 2002, Revised Papers. In: Sichman JS, Bousquet F, Davidsson P (eds) MABS. Lecture Notes in Computer Science, vol 2581. Springer, Berlin
 58. Torrens P (2002) Cellular automata and multi-agent systems as planning support tools. In: Geertman S, Stillwell J (eds) Planning Support Systems in Practice. Springer, London, pp 205–222
 59. Tummlolini L, Castelfranchi C, Ricci A, Viroli M, Omicini A (2004) “Exhibitionists” and “voyeurs” do it better: A shared environment approach for flexible coordination with tacit messages. In: Weyns D, Parunak HVD, Michel F (eds) 1st International Workshop on Environments for MultiAgent Systems (E4MAS 2004), pp 97–111
 60. Wahle J, Schreckenberg M (2001) A multi-agent system for on-line simulations based on real-world traffic data. In: Annual Hawaii International Conference on System Sciences (HICSS-34), IEEE Computer Society
 61. Weyns D, Holvoet T (2006) From reactive robots to situated multi-agent systems: a historical perspective on the role of environment in multi-agent systems. In: Dikenelli O, Gleizes MP, Ricci A (eds) Engineering Societies in the Agents World VI, 6th International Workshop ESAW (2005). Lecture Notes in Computer Science, vol 3963. Springer, Berlin, pp 63–88
 62. Weyns D, Schelfhout K, Holvoet T, Lefever T (2005) Decentralized control of E'GV transportation systems. In: AAMAS Industrial Applications. ACM Press, Utrecht, pp 67–74
 63. Weyns D, Vizzari G, Holvoet T (2006) Environments for situated multi-agent systems: beyond infrastructure. In: Weyns D, Parunak HVD, Michel F (eds) Environments for Multi-Agent Systems II, Second International Workshop E4MAS (2005), Utrecht, 25 July 2005. Selected Revised and Invited Papers. Lecture Notes in Computer Science, vol 3830. Springer, Berlin, pp 1–17
 64. Weyns D, Boucké N, Holvoet T (2006) Gradient field-based task assignment in an AGV transportation system. In: AAMAS '06: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems. ACM Press, Hakodate, pp 842–849
 65. Weyns D, Brueckner SA, Demazeau Y (eds) (2008) Engineering Environment-Mediated Multi-Agent Systems: International Workshop, EEMMAS 2007, Dresden, Germany, October 2007. Selected Revised and Invited Papers. Lecture Notes in Computer Science, vol 5049. Springer, Berlin
 66. Weyns D, Omicini A, Odell J (2007) Environment as a first class abstraction in multiagent systems. *Auton Agents Multi-Agent Syst* 14:5–30
 67. Wolfram S (1986) Theory and Applications of Cellular Automata. World Press, Singapore
 68. Wooldridge MJ, Jennings NR (1995) Intelligent agents: theory and practice. *Knowl Eng Rev* 10:115–152
 69. Zambonelli F, Parunak HVD (2003) Signs of a revolution in computer science and software engineering. In: Petta P, Tolksdorf R, Zambonelli F (eds) Engineering Societies in the Agents World III, Third International Workshop, ESAW 2002, Madrid, Spain, September 2002, Revised Papers. Lecture Notes in Computer Science, vol 2577. Springer, Berlin, pp 13–28

Agent Based Modeling and Simulation, Introduction to

FILIPPO CASTIGLIONE

Institute for Computing Applications (IAC) –
National Research Council (CNR), Rome, Italy

Agent-based modeling (ABM) is a relatively new computational modeling paradigm that is markedly useful in studying complex systems composed of a large number of interacting entities with many degrees of freedom. Other names for ABM are individual-based modeling (IBM) or multi-agent systems (MAS). Physicists often use the term micro-simulation or interaction-based computing.

The basic idea of ABM is to construct the computational counterpart of a conceptual model of a system under study on the basis of discrete entities (agents) with some properties and behavioral rules, and then to simulate them in a computer to mimic the real phenomena.

The definition of agent is somewhat controversial as witnessed by the fact that the models found in the literature adopt an extremely heterogeneous rationale. The agent is an autonomous entity having its own internal state reflecting its perception of the environment and interacting with other entities according to more or less sophisticated rules. In practice, the term agent is used to indicate entities ranging all the way from simple pieces of software to “conscious” entities with learning capabilities. For example, there are “helper” agents for web retrieval, robotic agents to explore inhospitable environments, buyer/seller agents in an economy, and so on. Roughly speaking, an entity is an “agent” if it has some degree of autonomy, that is, if it is distinguishable from its environment by some kind of spatial, temporal, or functional attribute: an agent must be identifiable. Moreover, it is usually required that an agent must have some autonomy of action and that it must be able to engage in tasks in an environment without direct external control.

From simple agents, which interact locally with simple rules of behavior, merely responding befittingly to environmental cues, and not necessarily striving for an overall goal, we observe a synergy which leads to a higher-level whole with much more intricate behavior than the component agents (*holism*, meaning all, entire, total). Agents can be identified on the basis of a set of properties that must characterize an entity, and in particular, *autonomy* (the capability of operating without intervention by humans, and a certain degree of control over its own state); *social ability* (the capability of interacting by employing some kind of agent communication language); *reactivity* (the ability

to perceive an environment in which it is situated and respond to perceived changes); and *pro-activeness* (the ability to take the initiative, starting some activity according to internal goals rather than as a reaction to an external stimulus). Moreover, it is also conceptually important to define what the agent “environment” in an ABM is.

In general, given the relative immaturity of this modeling paradigm and the broad spectrum of disciplines in which it is applied, a clear cut and widely accepted definition of high level concepts of agents, environment, interactions and so on, is still lacking. Therefore a real ABM ontology is needed to address the epistemological issues related to the agent-based paradigm of modeling of complex systems in order to attempt to reach a more general comprehension of emergent properties which, though ascribed to the definition of a specific application domain, are also universal (see ► [Agent Based Modeling and Simulation](#)).

Historically, the first simple conceptual form of agent-based models was developed in the late 1940s, and it took the advent of the computer to show its modeling power. This is the Von Neumann machine, a theoretical machine capable of reproduction. The device von Neumann proposed would follow precisely detailed instructions to produce an identical copy of itself. The concept was then improved by Stanislaw Ulam. He suggested that the machine be built on paper, as a collection of cells on a grid. This idea inspired von Neumann to create the first of the models later termed cellular automata (CA). John Conway then constructed the well-known “Game of Life”. Unlike the von Neumann’s machine, Conway’s *Game of Life* operated by simple rules in a virtual world in the form of a two-dimensional checkerboard.

Conway’s Game of Life has become a paradigmatic example of models concerned with the emergence of order in nature. How do systems self-organize themselves and spontaneously achieve a higher-ordered state? These and other questions have been deeply addressed in the first workshop on Artificial Life held in the late 1980s in Santa Fe. This workshop shaped the ALife field of research. Agent-based modeling is historically connected to ALife because it has become a distinctive form of modeling and simulation in this field. In fact, the essential features of ALife models are translated into computational algorithms through agent-based modeling (see ► [Agent Based Modeling and Artificial Life](#)).

Agent-based models can be seen as the natural extension of the CA-like models, which have been very successful in the past decades in shedding light on various physical phenomena. One important characteristic of ABMs which distinguishes them from cellular automata, is the

potential asynchrony of the interactions among agents and between agents and their environments. In ABMs, agents typically do not simultaneously perform actions at constant time-steps, as in CAs or boolean networks. Rather, their actions follow discrete-event cues or a sequential schedule of interactions. The discrete-event setup allows for the cohabitation of agents with different environmental experiences. Also ABMs are not necessarily grid-based nor do agents “tile” the environment.

Physics investigation is based on building models of reality. It is a common experience that, even using simple “building blocks”, one usually obtains systems whose behavior is quite complex. This is the reason why CA-like, and therefore agent-based models, have been used extensively among physicists to investigate experimentally (that is, on a computer) the essential ingredients of a complex phenomenon. Rather than being derived from some fundamental law of physics, these essential ingredients constitute artificial worlds. Therefore, there exists a pathway from Newton’s laws to CA and ABM simulations in classical physics that has not yet expressed all its potential (see ► [Interaction Based Computing in Physics](#)).

CA-like models also proved very successful in theoretical biology to describe the aggregation of cells or microorganisms in normal or pathological conditions (see ► [Cellular Automaton Modeling of Tumor Invasion](#)).

Returning to the concept of agent in the ABM paradigm, an agent may represent a particle, a financial trader, a cell in a living organism, a predator in a complex ecosystem, a power plant, an atom belonging to a certain material, a buyer in a closed economy, customers in a market model, forest trees, cars in large traffic vehicle system, etc. Once the level of description of the system under study has been defined, the identification of such entities is quite straightforward. For example, if one looks at the world economy, then the correct choice of agents are nations, rather than individual companies. On the other hand, if one is interested in looking at the dynamics of a stock, then the entities determining the price evolution are the buyers and sellers.

This example points to a field where ABM provides a very interesting and valuable instrument of research. Indeed, mainstream economic models typically make the assumption that an entire group of agents, for example, “investors”, can be modeled with a single “rational representative agent”. While this assumption has proven extremely useful in advancing the science of economics by yielding analytically tractable models, it is clear that the assumption is not realistic: people differ in their tastes, beliefs, and sophistication, and as many psychological studies have shown, they often deviate from rationality in

systematic ways. Agent-based computational economics (ACE) is a framework allowing economics to expand beyond the realm of the “rational representative agent”. By modeling and simulating the behavior of each agent and interactions among agents, agent-based simulation allows us to investigate the dynamics of complex economic systems with many heterogeneous (and not necessarily fully rational) agents. Agent-based computational economics complements the traditional analytical approach and is gradually becoming a standard tool in economic analysis (see ► [Agent Based Computational Economics](#)).

Because the paradigm of agent-based modeling and simulation can handle richness of detail in the agent’s description and behavior, this methodology is very appealing for the study and simulation of *social systems*, where the behavior and the heterogeneity of the interacting components are not safely reducible to some stylized or simple mechanism. Social phenomena simulation in the area of agent-based modeling and simulation, concerns the emulation of the individual behavior of a group of social entities, typically including their cognition, actions and interaction. This field of research aims at “growing” artificial society following a bottom-up approach.

Historically, the birth of the agent-based model as a model for social systems can be primarily attributed to a computer scientist, Craig Reynolds. He tried to model the reality of lively biological agents, known as artificial life, a term coined by Christopher Langton. In 1996 Joshua M. Epstein and Robert Axtell developed the first large scale agent model, the Sugarscape, to simulate and explore the role of social phenomenon such as seasonal migrations, pollution, sexual reproduction, combat, transmission of disease and even culture (see ► [Social Phenomena Simulation](#)).

In the field of artificial intelligence, the collective behavior of agents that without central control, collectively carry out tasks normally requiring some form of “intelligence”, constitutes the central concept in the field of *swarm intelligence*. The term “swarm intelligence” first appeared in 1989. As the use of the term swarm intelligence has increased, its meaning has broadened to a point in which it is often understood to encompass almost any type of collective behavior. Technologically, the importance of “swarms” is mainly based on potential advantages over centralized systems. The potential advantages are: economy (the swarm units are simple, hence, in principle, mass producible, *modularizable*, interchangeable, and disposable; reliability (due to the redundancy of the components; destruction/death of some units has negligible effect on the accomplishment of the task, as the swarm adapts to the loss of few units); ability to perform tasks beyond those of

centralized systems, for example, escaping enemy detection. From this initial perspective on potential advantages, the actual application of swarm intelligence has extended to many areas, and inspired potential future applications in defense and space technologies, (for example, control of groups of unmanned vehicles in land, water, or air), flexible manufacturing systems, and advanced computer technologies (bio-computing), medical technologies, and telecommunications (see ► [Swarm Intelligence](#)).

Similarly, robotics has adopted the ABM paradigm to study, by means of simulation, the crucial features of adaptation and cooperation in the pursuit of a global goal. Adaptive behavior concerns the study of how organisms develop their behavioral and cognitive skills through a synthetic methodology, consisting of designing artificial agents which are able to adapt to their environment autonomously. These studies are important both from a modeling point of view (that is, for better understanding intelligence and adaptation in natural beings) and from an engineering point of view (that is, for developing artifacts displaying effective behavioral and cognitive skills) (see ► [Embodied and Situated Agents, Adaptive Behavior in](#)).

What makes ABM a novel and interesting paradigm of modeling is the idea that agents are individually represented and “monitored” in the computer’s memory. One can, at any time during the simulation, ask a question such as “what is the age distribution of the agents?”, or “how many stocks have accumulated buyers following that specific strategy?”, or “what is the average velocity of the particles?”. “Large scale” simulations in the context of agent-based modeling are not only simulations that are large in terms of size (number of agents simulated), but are also complex. Complexity is inherent in agent-based models, as they are usually composed of dynamic, heterogeneous, interacting agents. Large-scale agent-based models have also been referred to as “Massively Multi-agent Systems (MMAS)”. MMAS is defined as “beyond resource limitation”: the number of agents exceeds local computer resources, or the situations are too complex to design and program given human cognitive resource limits. Therefore, for agent-based modeling, “large scale” is not simply a size problem, it is also a problem of managing complexity to ensure scalability of the agent model. Agent-based models increase in scale as the modeler requires many agents to investigate whole system behavior, or as the modeler wishes to fully examine the response of a single agent in a realistic context. There are two key problems that have to be tackled as the scale of a multi-agent system increases: computational resources limit the simulation time and/or data storage capacity; and agent model analysis may be-

come more difficult. Difficulty in analyzing the model may be due to the model system having a large number of complex components or due to memory for model output storage being restricted by computer resources (see ► [Agent Based Modeling, Large Scale Simulations](#)).

For implementation of agent-based models, both domain-specific and general-purpose languages are routinely used. Domain-specific languages include business-oriented languages (for example, spreadsheet programming tools); science and engineering languages (such as Mathematica); and dedicated agent-based modeling languages (for example, NetLogo). General-purpose languages can be used directly (as in the case of Java programming) or within agent-based modeling toolkits (for example, Repast). The choice that is most appropriate for a given modeling project depends on both the requirements of that project and the resources available to implement it (see ► [Agent Based Modeling and Computer Languages](#)).

Interestingly, ABM is not being used exclusively in science. In fact, the entertainment industry has promoted its own interest in the ABM technology. As graphics technology has improved in recent years, more and more importance has been placed on the behavior of virtual characters in applications set in virtual worlds in areas such as games, movies and simulations. The behavior of virtual characters should be believable in order to create the illusion that these virtual worlds are populated with living characters. This has led to the application of agent-based modeling to the control of these virtual characters. There are a number of advantages of using agent-based modeling techniques which include the fact that they remove the requirement for hand controlling all agents in a virtual environment, and allow agents in games to respond to unexpected actions by players (see ► [Computer Graphics and Games, Agent Based Modeling in](#)).

Since it is difficult to formally analyze complex multi-agent systems, they are mainly studied through computer simulations. While computer simulations can be very useful, results obtained through simulations do not formally validate the observed behavior. It is widely recognized that building a sound and widely applicable theory for ABM systems will require an inter-disciplinary approach and the development of new mathematical and computational concepts. In other words, there is a compelling need for a mathematical framework which one can use to represent ABM systems and formally establish their properties. Fortunately, some known mathematical frameworks already exist that can be used to formally describe multi-agent systems, for example, that of finite *dynamical systems* (both deterministic and stochastic). A sampling of the results from this field of mathematics shows that they can be used

to carry out rigorous studies of the properties of multi-agent systems and, in general, that they can also serve as universal models for computation. Moreover, special cases of dynamical systems (sequential dynamical systems) can be structured in accordance with the theory of categories and therefore provide the basis for a formal theory to describe ABM behavior (see ► [Agent Based Modeling, Mathematical Formalism for](#)).

On the same line of thought, agents and interaction can be studied from the perspective of logic and computer science. In particular, ideas about *logical dynamics*, *games semantics* and *geometry of interaction*, which have been developed over the past two decades, lead towards a structural theory of agents and interaction. This provides a basis for powerful logical methods such as compositionality, types and high-order calculi, which have proved so fruitful in computer science, to be applied in the domain of ABM and simulation (see ► [Logic and Geometry of Agents in Agent-Based Modeling](#)).

The appeal of ABM methodology in science increases manifestly with advances in computational power of modern computers. However, it is important to bear in mind that increasing the complexity of a model does not necessarily bring more understanding of the fundamental laws governing the overall dynamics. Actually, beyond a certain level of model complexity, the model loses its ability to explain or predict reality, thus reducing model building to a mere surrogate of reality where things may happen with a surprising adherence to reality, but we are unable to explain *why* this happens. Therefore, model construction must proceed incrementally, step by step, possibly validating the model at each stage of development before adding more details. ABM technology is very powerful but, if badly used, could reduce science to a mere exercise consisting of mimicking reality.

Agent Based Models in Economics and Complexity

MAURO GALLEGATI¹, MATTEO G. RICHARDI^{1,2}

¹ Università Politecnica delle Marche, Ancona, Italy

² Collegio Carlo Alberto – LABORatorio R. Revelli, Moncalieri, Italy

Article Outline

[Glossary](#)

[Definition of the Subject](#)

[Introduction](#)

Some Limits of the Mainstream Approach
 The Economics of Complexity
 Additional Features of Agent-Based Models
 An Ante Litteram Agent-Based Model:
 Thomas Schelling's Segregation Model
 The Development of Agent-Based Modeling
 A Recursive System Representation
 of Agent-Based Models
 Analysis of Model Behavior
 Validation and Estimation
 The Role of Economic Policy
 Future Directions
 Bibliography

Glossary

Abduction also called *inference to the best explanation*, abduction is a method of reasoning in which one looks for the hypothesis that would best explain the relevant evidence.

Agents entities of a model that (i) are perceived as a unit from the outside, (ii) have the ability to *act*, and possibly to *react* to external stimuli and *interact* with the environment and other agents.

Agent-based computational economics (ACE) is the computational study of economic processes modeled as dynamic systems of interacting agent.

Agent-based models (ABM) are models where (i) there is a multitude of objects that interact with each other and with the environment; (ii) the objects are autonomous, i. e. there is no central, or top-down control over their behavior; and (iii) the outcome of their interaction is numerically computed.

Complexity there are more than 45 existing definitions of complexity (Seth Lloyd, as reported on p. 303 in [97]). However, they can be grouped in just two broad classes: a *computational* view and a *descriptive* view. Computational (or algorithmic) complexity is a measure of the amount of information necessary to compute a system; descriptive complexity refers to the amount of information necessary to describe a system. We refer to this second view, and define complex systems as systems characterized by emergent properties (see *emergence*).

Deduction the logical derivation of conclusions from given premises.

Economics is the science about the intended and unintended consequences of individual actions, in an environment characterized by scarce resources that both requires and forces to interaction.

Emergence the spontaneous formation of self-organized structures at different layers of a hierarchical system configuration.

Evolution in biology, is a change in the inherited traits of a population from one generation to the next. In social sciences it is intended as an endogenous change over time in the behavior of the population, originated by competitive pressure and/or learning.

Heterogeneity non-degenerate distribution of characteristics in a population of agents.

Induction the intuition of general patterns from the observation of statistical regularities.

Interaction a situation when the actions or the supposed actions of one agent may affect those of other agents within a reference group.

Out-of-equilibrium a situation when the behavior of a system, in terms of individual strategies or aggregate outcomes, is not stable.

Definition of the Subject

A crucial aspect of the complexity approach is how interacting elements produce aggregate patterns that those elements in turn react to. This leads to the emergence of aggregate properties and structures that cannot be guessed by looking only at individual behavior.

It has been argued [144] that complexity is ubiquitous in economic problems (although this is rarely acknowledged in economic modeling), since (i) the economy is inherently characterized by the interaction of individuals, and (ii) these individuals have cognitive abilities, e. g. they form expectations on aggregate outcomes and base their behavior upon them: “Imagine how hard physics would be if electrons could think”, is how the Nobel prize winner Murray Gell-Mann, a physicist, has put it (as reported by Page [131]).

Explicitly considering how heterogeneous elements dynamically develop their behavior through interaction is a hard task analytically, the equilibrium analysis of mainstream (neoclassical) economics being a shortcut that in many cases is at risk of throwing the baby out with the bath water, so to speak. On the other hand, numerical computation of the dynamics of the process started to be a feasible alternative only when computer power became widely accessible. The computational study of heterogeneous interaction agents is called agent-based modeling (ABM). Interestingly, among its first applications a prominent role was given to economic models [4], although it was quickly found of value in other disciplines too (from sociology to ecology, from biology to medicine). The goal of this chapter is to motivate the use of the complexity approach

and agent-based modeling in economics, by discussing the weaknesses of the traditional paradigm of mainstream economics, and then explain what ABM is and which research and policy questions it can help to analyze.

Introduction

Economics is in troubled waters. Although there exists a mainstream approach, its internal coherence and ability to explain the empirical evidence are increasingly questioned. The causes of the present state of affairs go back to the middle of the eighteenth century, when some of the Western economies were transformed by the technological progress which led to the industrial revolution. This was one century after the Newtonian revolution in physics: from the small apple to the enormous planets, all objects seemed to obey the simple *natural* law of gravitation. It was therefore natural for a new figure of social scientist, the economist, to borrow the method (*mathematics*) of the most successful hard science, physics, allowing for the mutation of *political economy* into *economics*. It was (and still is) the mechanical physics of the seventeenth century, which ruled economics. In the final chapter of his *General Theory*, Keynes wrote of politicians as slaves of late economists: in their turn, they are slaves of late physicists of the seventeenth century (see also [125]).

From then on, economics lived its own evolution based on the classical physics assumptions (reductionism, determinism and mechanicism). Quite remarkably, the approach of statistical physics, which deeply affected physical science at the turn of the nineteenth century by emphasizing the difference between micro and macro, was adopted by Keynes around the mid 1930s. However, after decades of extraordinary success it was rejected by the neoclassical school around the mid 1970s, which framed the discipline into the old approach and ignored, by definition, any interdependencies among agents and difference between individual and aggregate behavior (being agents, electrons, nations or planets).

The ideas of natural laws and equilibrium have been transplanted into economics *sic et simpliciter*. As a consequence of the adoption of the classical mechanics paradigm, the difference between micro and macro was analyzed under a reductionist approach. In such a setting, aggregation is simply the process of summing up market outcomes of individual entities to obtain economy-wide totals. This means that there is no difference between micro and macro: the dynamics of the whole is nothing but a summation of the dynamics of its components (in terms of physics, the motion of a planet can be described by the dynamics of the atoms composing it). This approach does

not take into consideration that there might be two-way interdependencies between the agents and the aggregate properties of the system: interacting elements produce aggregate patterns that those elements in turn react to. What macroeconomists typically fail to realize is that the correct procedure of aggregation is not a sum: this is when emergence enters the drama. With the term emergence we mean the arising of complex structures from simple individual rules [147,153,171]. Empirical evidence, as well as experimental tests, shows that aggregation generates regularities, i. e. simple individual rules, when aggregated, produce statistical regularities or well-shaped aggregate functions: regularities emerge from individual chaos [106]. The concept of equilibrium is quite a dramatic example. In many economic models equilibrium is described as a state in which (individual and aggregate) demand equals supply. The notion of *statistical equilibrium*, in which the aggregate equilibrium is compatible with individual disequilibrium, is outside the box of tools of the mainstream economist. The same is true for the notion of *evolutionary equilibrium* (at an aggregate level) developed in biology. The equilibrium of a system no longer requires that every single element be in equilibrium by itself, but rather that the statistical distributions describing aggregate phenomena be stable, i. e. in “[...] a state of macroscopic equilibrium maintained by a large number of transitions in opposite directions” (p. 356 in [64]).

According to this view, an individual organism is in equilibrium *only when it is dead*. A consequence of the idea that macroscopic phenomena can emerge is that reductionism is wrong.

Ironically, since it can be argued, as we will do in the section below, that economics strongly needs this methodological twist [144], ABM has received less attention in economics than in other sciences ([110]; but [82] is a counter-example). The aim of this chapter is not to provide a review of applications of the complexity theory to economics (the interested reader is referred to [15,26,60,124,140,142]), but rather to describe the development of the *Agent-Based Modeling* (ABM) approach to complexity.

The chapter is structured as follows: after reviewing some limits of mainstream economics (Sect. “[Additional Features of Agent-Based Models](#)”), Sects. “[The Economics of Complexity](#)” and “[Additional Features of Agent-Based Models](#)” describe how the complexity perspective differs from the traditional one, and how many problems of the mainstream approach can be overcome by ABM. As an example, we present a prototypical example of ABM, based on the work of Thomas Schelling on the dynamics of segregation. After dedicating some sections to, respectively,

a skeleton history of ABM, a recursive system representation of these models, a discussion on how ABM can be interpreted, estimated and validated, we finally discuss how the complexity approach can be used to guide policy intervention and analysis. A final section discusses the achievements of the ABM agenda.

Some Limits of the Mainstream Approach

The research program launched by the neoclassical school states that macroeconomics should be explicitly grounded on microfoundations. This is how Robert Lucas put it: “The most interesting recent developments in macroeconomic theory seem to me describable as the reincorporation of aggregative problems [...] within the general framework of ‘microeconomic’ theory. If these developments succeed, the term ‘macroeconomic’ will be simply disappear from use and the modifier ‘micro’ will become superfluous. We will simply speak, as did Smith, Marshall and Walras, of *economic theory*” (pp. 107–108 in [115]). According to the mainstream, this implies that economic phenomena at a macroscopic level should be explained as a summation of the activities undertaken by individual decision makers. This procedure of microfoundation is very different from that now used in physics. The latter starts from the micro-dynamics of the single particle, as expressed by the Liouville equation and, through the master equation, ends up with the macroscopic equations. In the aggregation process, the dynamics of the agents lose their degree of freedom and behave coherently in the aggregate. In mainstream economics, while the procedure is formally the same (from micro to macro), it is assumed that the dynamics of the agents are those of the aggregate. The reduction of the degree of freedom, which is characteristic of the aggregation problem in physics, is therefore ruled out: a rational agent with complete information chooses to implement the individually optimal behavior, without additional constraints. There are three main pillars of this approach: (i) the precepts of the rational choice-theoretic tradition; (ii) the equilibrium concept of the Walrasian analysis; and (iii) the reductionist approach of classical physics. In the following, we will show that assumptions (i)–(ii), which constitute the necessary conditions for reducing macro to micro, are logically flawed (and empirically unfounded), while rejection of (iii) opens the road to complexity.

Mainstream economics is axiomatic and based on unrealistic (or unverifiable) assumptions. According to the supporters of this view, such an abstraction is necessary since the real world is complicated: rather than compromising the epistemic worth of economics, such assump-

tions are essential for economic knowledge. However, this argument does not invalidate the criticism of unrealistic assumptions [136]. While it requires internal coherence, so that theorems can be logically deduced from a set of assumptions, it abstracts from external coherence between theoretical statements and empirical evidence. Of course, this implies an important epistemological detachment from falsifiable sciences like physics. In setting the methodological stage for the dynamic stochastic general equilibrium (DSGE) macroeconomic theory, Lucas and Sargent declared:

“An economy following a multivariate stochastic process is now routinely described as being in equilibrium, by which is meant nothing more than that at each point in time (a) markets clear and (b) agents act in their own self-interest. This development, which stemmed mainly from the work of Arrow [...] and Debreu [...], implies that simply to look at any economic time series and conclude that it is a disequilibrium phenomenon is a meaningless observation. [...] The key elements of these models are that agents are rational, reacting to policy changes in a way which is in their best interests privately, and that the impulses which trigger business fluctuations are mainly unanticipated shocks.” (p. 7 in [116]).

The self-regulating order of Adam Smith [153] is transformed into a competitive general equilibrium (GE) in the form elaborated in the 1870s by Walras, that is a configuration of (fully flexible) prices and plans of action such that, at those prices, all agents can carry out their chosen plans and, consequently, markets clear. In a continuous effort of generalization and analytical sophistication, modern (neoclassical) economists interested in building microfoundations for macroeconomics soon recurred to the refinement proposed in the 1950s by Arrow and Debreu [14], who showed that also individual intertemporal (on an infinite horizon) optimization yields a GE, as soon as the economy is equipped with perfect price foresight for each future state of nature and a complete set of Arrow-securities markets [11], all open at time zero and closed simultaneously. Whenever these conditions hold true, the GE is an allocation that maximizes a properly defined social welfare function, or the equilibrium is Pareto-efficient (*First Welfare Theorem*).

The literature has pointed out several logical inconsistencies of the mainstream approach. Davis [44] identifies three impossibility results, which determine the breakdown of the mainstream, i.e. neoclassical, economics:

(i) Arrow's 1951 theorem showing that neoclassical theory is unable to explain social choice [10]; (ii) the Cambridge capital debate pointing out that mainstream is contradictory with respect to the concept of aggregate capital [40]; and (iii) the Sonnenschein–Mantel–Debreu results showing that the standard comparative static reasoning is inapplicable in general equilibrium models. In particular, a few points are worth remembering here.

1. The GE is neither unique nor locally stable under general conditions. This negative result, which refers to the work of Sonnenschein [155], Debreu [46] and Mantel [119], can be summarized along the following lines. Let the aggregate excess demand function $F(p)$ – obtained from aggregating among individual excess demands $f(p)$ – be a mapping from the price simplex Π to the commodity space P^N . A GE is defined as a price vector p such that $F(p^*) = 0$. It turns out that the only conditions that $F(\cdot)$ inherits from $f(\cdot)$ are continuity, homogeneity of degree zero and the Walras' law (i. e., the total value of excess demand is zero). These assure the existence, but neither the uniqueness nor the local stability of p^* , unless preferences generating individual demand functions are restricted to very implausible cases.
2. The existence of a GE is proved *via* the Brouwer's fix point theorem, i. e. by finding a continuous function $g(\cdot): \Pi \rightarrow \Pi$ so that any fixed point for $g(\cdot)$ is also an equilibrium price vector $F(p^*) = 0$. Suppose that we are interested in finding an algorithm which, starting from an arbitrary price vector p , chooses price sequences to check for p^* and halts when it finds it. In other terms, to find the GE price vector $F(p^*) = 0$ means that halting configurations are decidable. As this violates the undecidability of the halting problem for Turing machines, from a recursion theoretic viewpoint the GE solution is incomputable [138,167]. Notice that the same problem applies, in spite of its name, to the class of computable GE models [169].
3. By construction, in a GE all transactions are undertaken at the same equilibrium price vector. Economic theory has worked out two mechanisms capable of reaching this outcome. First, one can assume that buyers and sellers adjust, costless, their optimal supplies and demands to prices called out by a (explicit or implicit) fictitious auctioneer, who continues to do his job until he finds a price vector which clears all markets. Only then transactions take place (Walras' assumption). Alternatively, buyers and sellers sign provisional contracts and are allowed to freely (i. e., without any cost) recontract until a price vector is found which makes individual plans fully compatible. Once again, transactions occur only after the equilibrium price vector has been established (Edgeworth's assumption). Regardless of the mechanism one adopts, the GE model is one in which the formation of prices precedes the process of exchange, instead of being the result of it, through a *tatonnement* process occurring in a meta-time. Real markets work the other way round and operates in real time, so that the GE model cannot be considered a scientific explanation of real economic phenomena [9].
4. It has been widely recognized since Debreu [45], that integrating money in the theory of value represented by the GE model is at best problematic. No economic agent can individually decide to monetize alone; monetary trade should be the equilibrium outcome of market interactions among optimizing agents. The use of money – that is, a common medium of exchange and a store of value – implies that one party to a transaction gives up something valuable (for instance, his endowment or production) for something inherently useless (a fiduciary token for which he has no immediate use) in the hope of advantageously re-trading it in the future. Given that in a GE model actual transactions take place only after a price vector coordinating all trading plans has been freely found, money can be consistently introduced into the picture only if the logical keystone of the absence of transaction costs is abandoned. By the same token, since credit makes sense only if agents can sign contracts in which one side promises future delivery of goods or services to the other side, in equilibrium markets for debt are meaningless, and bankruptcy can be safely ignored. Finally, as the very notion of a GE implies that all transactions occur only when individual plans are mutually compatible, and this has to be true also in the labor market, the empirically observed phenomenon of involuntary unemployment and the microfoundation program put forth by Lucas and Sargent are logically inconsistent.
5. The very absence of money and credit is a consequence of the fact that in GE there is no time. The only role assigned to time in a GE model is, in fact, that of dating commodities. Products, technologies and preferences are exogenously given and fixed from the outset. The convenient implication of banning out-of-equilibrium transactions is simply that of getting rid of any disturbing influence of intermediary modifications of endowments – and therefore of individual excess demands – on the final equilibrium outcome. The introduction of non-Walrasian elements into the GE microfoundations program – such as fixed or sticky prices, imperfect competition and incomplete markets leading to temporary

equilibrium models – yields interesting Keynesian features such as the breaking of the Say's law and scope for a monetary theory of production, a rationale for financial institutions and a more persuasive treatment of informational frictions. As argued in Vriend [165], however, all these approaches preserve a Walrasian perspective in that models are invariably closed by a GE solution concept which, implicitly or (more often) not, implies the existence of a fictitious auctioneer who processes information, calculates equilibrium prices and quantities, and regulates transactions. As a result, if the Walrasian auctioneer is removed the decentralized economy becomes dynamically incomplete, as we are not left with any mechanism determining how quantities and prices are set and how exchanges occur.

The flaws of the solution adopted by mainstream macroeconomists to overcome the problems of uniqueness and stability of equilibrium on the one hand, and of analytical-tractability on the other one – i. e. the usage of a representative agent (RA) whose choices summarize those of the whole population of agents – are so pervasive that we discuss them hereafter.

6. Although the RA framework has a long history, it is standard to build the microfoundation procedure on it only after Lucas' critique paper [114]. Mainstream models are characterized by an explicitly stated optimization problem of the RA, while the derived individual demand or supply curves are used to obtain the aggregate demand or supply curves. Even when the models allow for heterogeneity, interaction is generally absent (the so-called *weak interaction hypothesis* [139]). The use of RA models should allow one to avoid the Lucas critique, to provide microfoundations to macroeconomics, and, *ça va sans dire*, to build Walrasian general equilibrium models. Since models with many heterogeneous interacting agents are complicated and no closed form solution is often available (aggregation of heterogeneous interacting agents is analyzed in [5,6,7,53,78]), economists assume the existence of an RA: a simplification that makes it easier to solve for the competitive equilibrium allocation, since direct interaction is ruled out by definitions. Unfortunately, as Hildenbrand and Kirman [95] noted:

“There are no assumptions on isolated individuals, which will give us the properties of aggregate behavior. We are reduced to making assumptions at the aggregate level, which cannot be justified, by the usual individualistic assumptions.

This problem is usually avoided in the macroeconomic literature by assuming that the economy behaves like an individual. Such an assumption cannot be justified in the context of the standard model”.

The equilibria of general equilibrium models with a RA are characterized by a complete absence of trade and exchange, which is a counterfactual idea. Kirman [99], Gallegati [76] and Caballero [36] show that RA models ignore valid aggregation concerns, by neglecting interaction and emergence, hence committing fallacy of composition (what in philosophy is called fallacy of division, i. e. to attribute properties to a different level than where the property is observed: game theory offers a good case in point with the concept of Nash equilibrium, by assuming that social regularities come from the agent level equilibrium). Those authors provide examples in which the RA does not represent the individuals in the economy so that the reduction of a group of heterogeneous agents to an RA is not just an analytical convenience, but it is both unjustified and leads to conclusions which are usually misleading and often wrong ([99]; see also [98]). A further result, which is a proof of the logical fallacy in bridging the micro to the macro is the *impossibility theorem* of Arrow: it shows that an ensemble of people, which has to collectively take a decision, cannot show the same rationality of an individual [123]. Moreover, the standard econometric tools are based upon the assumption of an RA. If the economic system is populated by heterogeneous (not necessarily interacting) agents, then the problem of the microfoundation of macroeconometrics becomes a central topic, since some issues (e. g., *co-integration*, *Granger-causality*, *impulse-response function of structural VAR*) lose their significance [69].

All in all, we might say that the failure of the RA framework, points out the *vacuum* of the mainstream microfoundation literature, which ignores interactions: no box of tools is available to connect the micro and the macro levels, beside the RA whose existence is at odds with the empirical evidence [30,158] and the equilibrium theory as well [99].

The Economics of Complexity

According to the mainstream approach there is no direct interaction among economic units (for a pioneeristic and neglected contribution see [68]; see also [101]). In the most extreme case, any individual strategy is excluded (*principle of excluded strategy*, according to Schumpeter [149]) and agents are homogeneous. Small departures from the

perfect information hypothesis are incoherent with the Arrow–Debreu general equilibrium model, as shown by Grossman and Stiglitz [88], since they open the chance of having direct links among agents [156]. In particular, if prices convey information about the quality there cannot be an equilibrium price as determined by the demand–supply schedule, since demand curves depend on the probability distribution of the supply (p. 98 in [87]).

What characterizes a complex system is the notion of emergence, that is the spontaneous formation of self-organized structures at different layers of a hierarchical system configuration [43]. Rather, mainstream economics conceptualizes economic systems as consisting of several identical and isolated components, each one being a copy of a RA. The aggregate solution can thus be obtained by means of a simple summation of the choices made by each optimizing agent. The RA device, of course, is a way of avoiding the problem of aggregation by eliminating heterogeneity. But heterogeneity is still there. If the macroeconomist takes it seriously, he/she has to derive aggregate quantities and their relationships from the analysis of the micro-behavior of different agents. This is exactly the key point of the *aggregation problem*: starting from the *micro-equations* describing/representing the (optimal) choices of the economic units, what can we say about the *macro-equations*? Do they have the same functional form of the micro-equations (the *analogy principle*)? If not, how is the macro-theory derived?

The complexity approach to economics discards the GE approach to the microfoundation program, as well as its RA shorthand version. Instead of asking to deductively prove the existence of an equilibrium price vector p^* such that $F(p^*) = 0$, it aims at explicitly constructing it by means of an algorithm or a rule. From an epistemological perspective, this implies a shift from the realm of classical to that of constructive theorizing [168]. Clearly, the act of computationally constructing a coordinated state – instead of imposing it via the Walrasian auctioneer – for a decentralized economic system requires complete description of goal-directed economic agents and their interaction structure.

Agent-based modeling represents an effective implementation of this research agenda ([60,124], see also [24, 67,81,175]). ABM is a methodology that allows one to construct, based on simple rules of behavior and interaction, models with heterogeneous agents, where the resulting aggregate dynamics and empirical regularities are not known a priori and are not deducible from individual behavior. It is characterized by three main tenets: (i) there is a multitude of objects that interact with each other and with the environment; (ii) the objects are autonomous, i. e.

there is no central, or top-down control over their behavior; and (iii) the outcome of their interaction is numerically computed. Since the objects are autonomous, they are called agents ([3,4]; see also the repository of ACE-related material maintained by Leigh Tesfatsion at <http://www.econ.iastate.edu/tesfatsi/ace.htm>): “Agent-based Computational Economics is the computational study of economic processes modeled as dynamic systems of interacting agent” [161].

Agents can be anything from cells to biological entities, from individuals to social groups like families or firms. Agents can be composed by other agents: the only requirement being that they are perceived as a unit from the outside, and that they do something, i. e. they have the ability to *act*, and possibly to *react* to external stimuli and *interact* with the environment and other agents. The environment, which may include physical entities (infrastructures, geographical locations, etc.) and institutions (markets, regulatory systems, etc.), can also be modeled in terms of agents (e. g. a central bank, the order book of a stock exchange, etc.), whenever the conditions outlined above are met. When not, it should be thought of simply as a set of variables (say, temperature or business confidence).

The methodological issues are the real *litmus paper* of the competing approaches. According to one of the most quoted economic papers, Friedman [71], the ultimate goal of a positive science is to develop hypotheses that yield valid and meaningful *predictions* about actual phenomena. Not a word on predictions at the *meso*-level or on the realism of the hypotheses. Even the Occam rule is systematically ignored: e. g. to get a downward sloping aggregate demand curve, mainstream economics has to assume indifference curves which are: (i) defined only in the positive quadrant of commodity-bundle quantities; (ii) negatively sloped; (iii) complete; (iv) transitive, and (v) strictly convex, while ABM has to assume only the existence of reservation prices. Moreover, to properly aggregate from microbehavior, i. e. to get a well shaped aggregate demand from the individual ones, it has to be assumed that the propensity to consume out of income has to be homogeneous for all the agents (*homothetic* Engel curves) and that distribution is independent from relative prices. This methodology resembles the scientific procedure of the aruspexes, who predicted the future by reading the animals’ bowels. The ABM methodology is bottom-up and focuses on the interaction between many heterogeneous interacting agents, which might produce a statistical equilibrium, rather than a *natural* one as the mainstream approach assumes. The bottom-up approach models individual behavior according to simple behavioral rules; agents are allowed to have *local interaction* and to change

the *individual rule (through adaptation)* as well as the *interaction nodes*. By aggregating, some *statistical regularity* emerges, which cannot be inferred from individual behavior (*self emerging regularities*): this *emergent behavior* feeds back to the individual level (*downward causation*) thus establishing a macrofoundation of micro. As a consequence, each and every proposition may be falsified at *micro*, *meso* and *macro* levels. This approach opposes the axiomatic theory of economics, where the optimization procedure is the standard for a scientific, i. e. not ad-hoc, modeling procedure.

The agent-based methodology can also be viewed as a way to reconcile the two opposing philosophical perspectives of *methodological individualism* and *holism*. Having agents as the unit of analysis, ABM is deeply rooted in methodological individualism, a philosophical method aimed at explaining and understanding broad society-wide developments as the aggregation of decisions by individuals [13,172]. Methodological individualism suggests – in its most extreme (and erroneous) version – that a system can be understood by analyzing *separately* its constituents, the reductionist approach that the whole is nothing but the sum of its parts [51,127]. However, the ability to reduce everything to simple fundamental objects and laws does not imply the ability to start from those objects and laws and reconstruct the universe. In other terms, reductionism does not imply constructionism [2].

The Austrian school of economics championed the use of methodological individualism in economics in the twentieth century, of which Friederich von Hayek has been one of the main exponents. The legacy of Hayek to ABM and the complex system approach has been recognized [166]. Methodological individualism is also considered an essential part of modern neoclassical economics, with its analysis of collective action in terms of rational, utility-maximizing individuals: should the microfoundations in terms of individual rational behavior be abandoned, the Lucas Critique [114] would kick in. However, it is hard to recognize the imprinting of methodological individualism in the RA paradigm, which claims that the whole society can be analyzed in terms of the behavior of a single, representative, individual and forgets to apply to it the Lucas critique. On the other hand, focusing on aggregate phenomena arising from the bottom up [61] from the interaction of many different agents, ABM also adopts a holistic approach when it claims that these phenomena cannot be studied without looking at the entire context in which they are embedded. Indeed, holism is the idea that all the properties of a given system cannot be determined or explained by the sum of its component parts alone. Instead, the system as a whole determines in

an important way that the parts behave. The general principle of holism was concisely summarized by Aristotle in his *Metaphysics*: “The whole is more than the sum of its parts”, a *manifesto* of the complexity approach. However, ABM (and more in general complexity theory) should not be confused with general systems theory, an holistic approach developed in the 1950s and 1960s that in its most radical form argued that everything affects everything else: according to systems theory, phenomena that appear to have simple causes, such as unemployment, actually have a variety of complex causes – complex in the sense that the causes are interrelated, nonlinear, and difficult to determine [133]. Conversely, the complexity approach looks for *simple* rules that underpin complexity, an agenda that has been entirely transferred to ABM.

Also, ABM can be thought of as a bridge between methodological individualism and methodological holism. In agent-based models aggregate outcomes (the whole, e. g. the unemployment rate) are computed as the sum of individual characteristics (its parts, e. g. individual employment status). However, aggregate behavior can often be recognized as distinct from the behavior of the comprising agents, leading to the discovery of emergent properties. In this sense, the whole is more than – and different from – the sum of its parts. It might even be the case that the whole appears to act as if it followed a distinct logic, with its own goals and means, as in the example of a cartel of firms that act in order to influence the market price of a good. From the outside, the whole appears no different from a new agent type (e. g. a family, a firm). A new entity is born; the computational experiment has been successful in growing artificial societies from the bottom up [61].

This *bottom-up* approach to complexity consists in deducing the macroscopic objects (*macros*) and their phenomenological complex *ad-hoc* laws in terms of a multitude of elementary microscopic objects (*micros*) interacting by simple fundamental laws [154], and ABM provides a technique that allows one to systematically follow the birth of these complex macroscopic phenomenology. The *macros* at a specific scale can become the *micros* at the next scale.

Depending on the scope of the analysis, it is generally convenient to stop at some scale in the way down to reconstruct aggregate, top-level dynamics from the bottom up. When applied to economics, only a few levels (e. g. a *micro*, a *meso* and a *macro* level) are in general sufficient to provide a thorough understanding of the system. Defining the elementary units of analysis amounts to fixing the limits for the reductionist approach, which is not aprioristically discarded but rather integrated in the analysis. These units

are in fact characterized by an inner structure that does not depend on the environment in which they are embedded. They can thus be analyzed separately.

The need for the ABM approach at any given scale is often linked to the existence of some underlying autocatalytic process at a lower level. *Autocatalytic processes* are dynamic processes with positive feedbacks, where the growth of some quantity is to some extent self-perpetuating, as in the case when it is proportional to its initial value. The importance of positive feedbacks has been recognized in the literature on increasing returns, in particular with respect to the possibility of multiple equilibria [151], since the time of Marshall. However, the traditional analysis is static, and does not address how an equilibrium out of several might be selected. Looking at the problem from a dynamic stochastic process perspective, selection is explained in terms of one set of small historical events magnified by increasing returns.

Moreover, the existence of an autocatalytic process implies that looking at the average, or most probable, behavior of the constituent units is non representative of the dynamics of the system: autocatalyticity insures that the behavior of the entire system is dominated by the elements with the highest auto-catalytic growth rate rather than by the typical or average element [154]. In presence of autocatalytic processes, even a small amount of individual heterogeneity invalidates any description of the behavior of the system in terms of its average element: the real world is controlled as much by the *tails* of distributions as by means or averages. We need to free ourselves from *average* thinking [3].

The fact that autocatalytic dynamics are scale invariant (i. e. after a transformation that multiplies all the variables by a common factor) is a key to understanding the emergence of scale invariant distributions of these variables (e. g. power laws), at an aggregate level. The relevance of scale free distributions in economics (e. g. of firm size, wealth, income, etc.) is now extensively recognized (Brock, 1999), and has been the subject of thorough investigation in the econophysics literature [120].

Additional Features of Agent-Based Models

We have so far introduced the three fundamental characteristics of ABM: there are agents that play the role of actors, there is no script or *Deus ex-machina* and the story is played live, i. e. it is computed. Following Epstein [58,59,60], we can further characterize the methodology, by enumerating a number of features that, although not necessary to define an agent-based model, are often present. These are:

Heterogeneity

While in analytical models there is a big advantage in reducing the ways in which individuals differ, the computational burden of ABM does not change at all if different values of the parameters (e. g. preferences, endowments, location, social contacts, abilities etc.) are specified for different individuals. Normally, a distribution for each relevant parameter is chosen, and this simply implies that a few parameters (those governing the distribution) are added to the model.

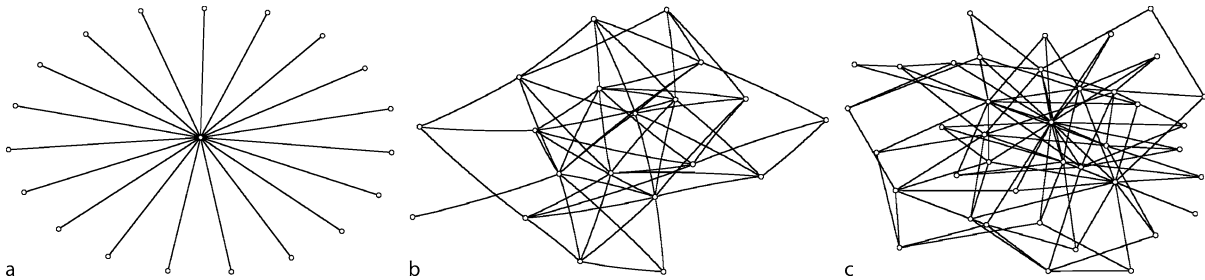
Explicit Space

This can be seen as specification of the previous point: individuals often differ in the physical place where they are located, and/or in the neighbors with whom they can or have to interact (which define the network structure of the model, see below).

Local Interaction

Again, this can be seen as a specification of the network structure connecting the agents. Analytical models often assume either global interaction (as in Walrasian markets), or very simple local interaction. ABM allow for much richer specifications. No direct interaction (only through prices) is allowed in the GE, while direct interaction (*local and stochastic*, usually [101]) is the rule for the complexity approach: figures 1a-c give a graphical representation of Walrasian, random and scale-free interaction respectively. Note that the empirical evidence supports the third case: hubs and power laws are the rule in the real world [38,52].

Actually, some neoclassical economists asked for an analysis of how social relations affect the allocation of resources (e. g., [12,107,134]). They went almost completely unheard, however, until the upsurge in the early 1990s of a brand new body of work aimed at understanding and modeling the social context of economic decisions, usually labeled *new social economics* or *social interaction economics* [56]. Models of social interactions (Manski [118] offers an operational classification of the channels through which the actions of one agent may affect those of other agents within a reference group) are generally able to produce several properties, such as *multiple equilibria* [34]; *non-ergodicity* and *phase transition* [54]; *equilibrium stratification* in social and/or spatial dimension [27,83]; the existence of a *social multiplier* of behaviors [84]. The key idea consists in recognizing that the social relationships in which individual economic agents are embedded can have a large impact on economic decisions. In fact, the social context impacts on individual economic decisions



Agent Based Models in Economics and Complexity, Figure 1

a a Walrasian GE representation; b a random graph; c a scale free graph in which several hubs can be identified

through several mechanisms. First, social norms, cultural processes and economic institutions may influence motivations, values, and tastes and, ultimately, make preferences endogenous [31]. Second, even if we admit that individuals are endowed with exogenously given preferences, the pervasiveness of information asymmetries in real-world economies implies that economic agents voluntarily share values, notions of acceptable behavior and socially based enforcement mechanisms in order to reduce uncertainty and favor coordination [50]. Third, the welfare of individuals may depend on some social characteristics like honor, popularity, stigma or status [41]. Finally, interactions not mediated by enforceable contracts may occur because of pure technological externalities in network industries [152] or indirect effects transmitted through prices (pecuniary externalities) in non-competitive markets [28], which may lead to coordination failures due to strategic complementarities [42].

Bounded Rationality

Interestingly, while in analytical models it is generally easier to implement some form of optimal behavior rather than solving models where individuals follow “reasonable” rules of thumb, or learn either by looking at what happened to others or what happened to them in the past, for ABM the opposite is true. However, it can be argued that *real* individuals also face the same difficulties in determining and following the optimal behavior, and are characterized by some sort of bounded rationality: “There are two components of this: bounded information and bounded computing power. Agents have neither global information nor infinite computational capacity. Although they are typically purposive, they are not global optimizers; they use simple rules based on local information” (p. 1588 in [59]).

The requirement on full rationality is indeed very strong, since it requires an infinite computational capacity (the ability of processing tons of data in a infinitesimal

amount of time) and all the information. Moreover, according to the mainstream approach, information is complete and free for all the agents. Note that one of the assumptions in the Walrasian approach is that each agent has only private information: this is equivalent to say that *strategic behavior* about information collection and dissemination is ruled out and the collection of the whole set of the information is left to the market *via* the auctioneer (or a benevolent dictator [25]). Indeed, one could read the rational expectation “revolution” as the tentative to decentralize the price setting procedure by defenestrating the auctioneer. Limited information is taken into account, but the constraints have to affect every agent in the same way (the so-called Lucas’ islands hypothesis) and the Greenwald–Stiglitz theorem [86] states that in this case the equilibrium is not even Pareto-constrained. If information is asymmetric or private, agents have to be heterogeneous and direct interaction has to be considered: this destroys the mainstream model and generates coordination failures.

On the contrary, agent-based models are build upon the hypothesis that agents have limited information. Once again, the ABM approach is much more parsimonious, since it only requires that the agents do not commit systematic errors. Moreover, given the limited information setting, the economic environment might change affecting, and being affected by, agents’ behavior: individuals learn through experience and by interacting with other agents.

Non-equilibrium Dynamics

As we will explain in more details below, ABM are recursive models, in which the state of the system at time $t + 1$ is computed starting from the state at time t . Hence, they allow the investigation of what happens all along the route, not only at the start and at the end of the journey. This point is, we believe, the most important. Brian Arthur (p. 1552 in [16]) offers an effective statement of

its relevance for economic theory: “Standard neoclassical economics asks what agents’ actions, strategies, or expectations are in equilibrium with (consistent with) the outcome or pattern these behaviors aggregatively create. Agent-based computational economics enables us to ask a wider question: how agents’ actions, strategies or expectations might react to – might endogenously change with – the pattern they create. [...] This out-of-equilibrium approach is not a minor adjunct to standard economic theory; it is economics done in a more general way. [...] The static equilibrium approach suffers two characteristic indeterminacies: it cannot easily resolve among multiple equilibria; nor can it easily model individuals’ choices of expectations. Both problems are ones of formation (of an equilibrium and of an ‘ecology’ of expectations, respectively), and when analyzed in formation – that is, out of equilibrium – these anomalies disappear”.

As we have seen, continuous market clearing is *assumed* by the mainstream. It is a necessary condition to obtain “efficiency and optimality” and it is quite curious to read of a theory assuming the *explanandum*. In such a way, every out of equilibrium dynamics or path dependency is ruled out and initial conditions do not matter. The GE model assumes that transactions happen only after the vector of the equilibrium prices has been reached: instead of being the result of the exchange, it foresees it *par tatonnement* in a logical, fictitious time. Because the real markets operate in real, historical, time and the exchange process determines prices, the GE model is not able to describe any real economy [9]. Clower [39] suggested (resemblance Edgeworth, [57]) that exchange might happen out of equilibrium (*at false prices*). In such a case, agents will be quantity-rationed in their supply of-demand for: because of it, the intertemporal maximization problem has to be quantity-constraints (the so-called *Clower constraint*) and if the economy would reach equilibrium, it will be non-optimal and inefficient.

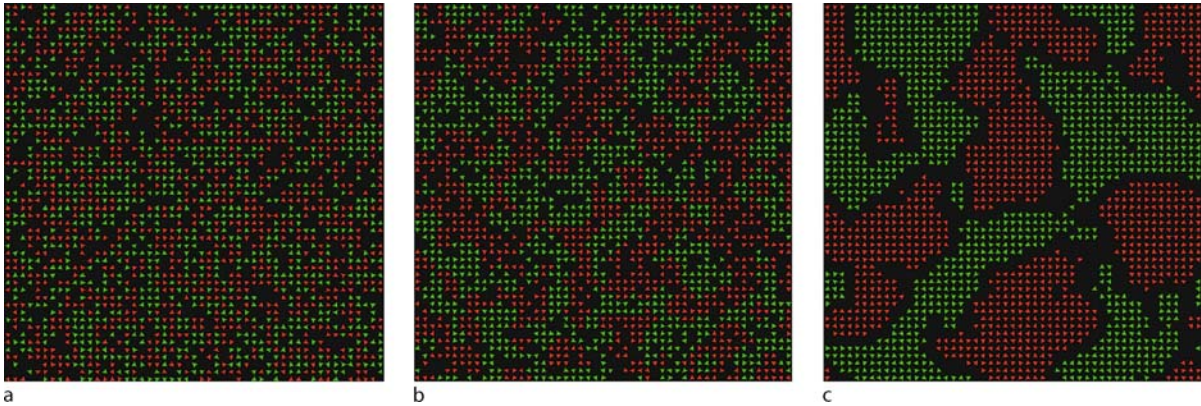
The requirement on rationality is also very strong, since it requires an infinite computational capacity (the ability of processing tons of data in a infinitesimal amount of time) and all the information. In fact, if information is limited, the outcome of a rational choice may be non-optimal. Once again, all the ABM approach is much more parsimonious, since it requires that the agents do not commit systematic errors. Moreover, given the limited information setting, the economic environment might change affecting, and being affected by, agents’ behavior: learning and adaptive behavior are therefore contemplated.

Finally, according to Beinhocker [26], the approaches differ also as regard *dynamics* (*Complex Systems* are open, dynamic, non-linear systems, far from equilibrium; *Main-*

stream economics are closed, static, linear systems in equilibrium) and *evolution* (*Complex Systems* have an evolutionary process of differentiation, selection and amplification which provides the system with novelty and is responsible for its growth in order and complexity, while *Mainstream* has no mechanism for endogenously creating novelty, or growth in order and complexity).

An Ante Litteram Agent-Based Model: Thomas Schelling’s Segregation Model

One of the early and most well known examples of an agent-based model is the segregation model proposed by Thomas Schelling [145,146], who in 2005 received the Nobel prize for his studies in game theory (surveys of more recent applications of ABM to economics can be found in [159,160,161,163]). To correctly assess the importance of the model, it must be evaluated against the social and historical background of the time. Up to the end of the 1960s racial segregation was institutionalized in the United States. Racial laws required that public schools, public places and public transportation, like trains and buses, had separate facilities for whites and blacks. Residential segregation was also prescribed in some States, although it is now widely recognized that it mainly came about through organized, mostly private efforts to ghettoize blacks in the early twentieth century – particularly the years between the world wars [63,126]. But if the social attitude was the strongest force in producing residential segregation, the Civil Rights movement of the 1960s greatly contributed to a change of climate, with the white population exhibiting increasing levels of tolerance. Eventually, the movement gained such strength to achieve its main objective, the abolition of the racial laws: this was sealed in the Civil Rights Act of 1968 which, among many other things, outlawed a wide range of discriminatory conduct in housing markets. Hence, both the general public attitude and the law changed dramatically during the 1960s. As a consequence, many observers predicted a rapid decline in housing segregation. The decline, however, was almost imperceptible. The question then was why this happened. Schelling’s segregation model brought an answer, suggesting that small differences in tolerance level or initial location could trigger high level of segregation even without formal (i. e. legal) constraints, and even for decent levels of overall tolerance. In the model, whites and blacks are (randomly) located over a grid, each individual occupying one cell. As a consequence, each individual has at most eight neighbors (Moore neighborhood), located on adjacent cells. Preferences over residential patterns are represented as the maximum quota of racially different



Agent Based Models in Economics and Complexity, Figure 2

NETLOGO implementation of Schelling's segregation model. **a** Initial (random) pattern. The average share of racially similar neighbors is roughly 50%. With a tolerance level of 70% (40%), less than 20% (more than 80%) of the individuals are not satisfied. **b** Final pattern. The average share of racially similar neighbors is 72.1%. Everyone is satisfied. **c** Final pattern. The average share of racially similar neighbors is 99.7%. Everyone is satisfied

neighbors that an individual tolerates. For simplicity, we can assume that preferences are identical: a unique number defines the level of tolerance in the population. For example, if the tolerance level is 50% and an individual has only five neighbors, he would be satisfied if no more than two of his neighbors are racially different. If an individual is not satisfied by his current location, he tries to move to a different location where he is satisfied.

The mechanism that generates segregation is the following. Since individuals are initially located randomly on the grid, by chance there will be someone who is not satisfied. His decision to move creates two externalities: one in the location of origin and the other in the location of destination. For example, suppose a white individual decides to move because there are too many black people around. As he leaves, the ethnic composition of his neighborhood is affected (there is one white less). This increases the possibility that another white individual, who was previously satisfied, becomes eager to move. A similar situation occurs in the area of destination. The arrival of a white individual affects the ethnic composition of the neighborhood, possibly causing some black individual to become unsatisfied. Thus, a small non-homogeneity in the initial residential pattern triggers a chain effect that eventually leads to high levels of segregation. This mechanism is reinforced when preferences are not homogeneous in the population.

Figure 2, which shows the NETLOGO implementation of the Schelling model, exemplifies [173]. The left panel depicts the initial residential pattern, for a population of 2000 individuals, evenly divided between green and red, living on a 51×51 cells torus (hence the population density is 76.9%). Two values for the tolerance threshold are

tested: in the first configuration, tolerance is extremely high (70%), while in the second it is significantly lower (30%), although at a level that would still be considered decent by many commentators. The initial residential pattern (obviously) shows no levels of segregation: every individual has on average 50% of neighbors of a different race. However, after just a few periods the equilibrium configurations of the middle (for a tolerance level of 70%) and right (for tolerance level of 30%) panels are obtained. The level of segregation is high: more than three quarters of neighbors are on average of the same racial group, even in case (b), when individuals are actually happy to live in a neighborhood dominated by a different racial group! Moreover, most people live in perfectly homogeneous clusters, with different ethnic clusters being often physically separated from each other by a no man's land. Only the relative mix brought by confining clusters keeps down the measure of overall segregation. Should the overall composition of the population be biased in favor of one ethnic group, we would clearly recognize the formation of ghettos.

Note that the formation of racially homogeneous ethnic clusters and ghettos is an emergent property of the system, which could hardly be deduced by looking at individual behavior alone, without considering the effects of interaction. Moreover, the clusters themselves could be considered as the elementary unit of analysis at a different, more aggregate level, and their behavior, e.g. whether they shrink, expand, merge or vanish, studied with respect to some exogenous changes in the environment. Not only a *property*, i.e. a statistical regularity, has emerged, but also a whole new *entity* can be recognized. However, this

new entity is nothing else but a subjective interpretation by some external observer of an emergent property of the system.

The Development of Agent-Based Modeling

The early example of the segregation model notwithstanding, the development of agent-based computational economics is closely linked with the work conducted at the Santa Fe Institute for the study of complexity, a private, non-profit, independent research and education center founded in 1984 in Santa Fe, New Mexico. The purpose of the institute has been, since its foundation, to foster multi-disciplinary collaboration in pursuit of understanding the common themes that arise in natural, artificial, and social systems. This unified view is the dominant theme of what has been called the *new science of complexity*.

The outcomes of this research program are well depicted in three books, all bearing the title *The economy as an evolving complex system* [4,17,29]. The following quotation, from the preface of the 1997 volume, summarizes very accurately the approach:

“In September 1987 twenty people came together at the Santa Fe Institute to talk about ‘the economy as a evolving, complex system’. Ten were theoretical economists, invited by Kenneth J. Arrow, and ten were physicists, biologists and computer scientists, invited by Philip W. Anderson. The meeting was motivated by the hope that new ideas bubbling in the natural sciences, loosely tied together under the rubric of ‘the sciences of complexity’, might stimulate new ways of thinking about economic problems. For ten days, economists and natural scientists took turns talking about their respective worlds and methodologies. While physicists grappled with general equilibrium analysis and non-cooperative game theory, economists tried to make sense of spin glass models, Boolean networks, and genetic algorithms. The meeting left two legacies. The first was the 1988 volume of essays; the other was the founding, in 1988, of the Economics Program at the Santa Fe Institute, the Institute’s first resident research program. The Program’s mission was to encourage the understanding of economic phenomena from a complexity perspective, which involved the development of theory as well as tools for modeling and for empirical analysis. [...] But just what is the complexity perspective in economics? That is not an easy question to answer. [...] Looking back over the developments in the past decade, and of the papers produced by the program, we believe that a coher-

ent perspective – sometimes called the ‘Santa Fe approach’ – has emerged within economics.”

The work carried out at the Santa Fe Institute greatly contributed to popularize the complexity approach to economics, although a similar line of research was initiated in Europe by chemists and physicists concerned with emergent structures and disequilibrium dynamics (more precisely, in Brussels by the group of the Nobel prize winner physical chemist Ilya Prigogine ([128]) and in Stuttgart by the group of the theoretical physicist Hermann Haken [91], as discussed in length by Rosser [141]).

Two main reasons can help explaining why the Santa Fe approach gained some visibility outside the restricted group of people interested in the complexity theory (perhaps contributing in this way to mount what Horgan [96,97], called an intellectual fad). Together, they offered an appealing suggestion of both what to do and how to do it. The first reason was the ability to present the complexity paradigm as a unitary perspective. This unitary vision stressed in particular the existence of feedbacks between *functionalities* and *objectives*: individual objectives determine to some extent the use and modification of existing functionalities, but functionalities direct to some extent the choice of individual objectives. It is this analytical focus that proved to be valuable in disciplines as diverse as the social sciences, the biological sciences and even architecture [70]. The second reason has to do with the creation of a specific simulation platform that allowed relatively inexperienced researchers to build their own toy models that, thanks to the enormous and sustained increase in commonly available computing power, could run quickly even on small PCs. This simulation platform was called SWARM [18], and consisted in a series of libraries that implemented many of the functionalities and technicalities needed to build an agent-based simulation, e. g. the schedule of the events, the passing of time and graphical widgets to monitor the simulation. In addition to offering a practical tool to write agent-based simulations, the SWARM approach proposed a protocol in simulation design, which the SWARM libraries exemplified.

The principles at the basis of the SWARM protocol are:

- (i) The use of an *object-oriented programming* language (SWARM was first written in OBJECTIVE C, and later translated into JAVA), with different objects (and object types) being a natural counterpart for different agents (and agent types);
- (ii) A *separate implementation* of the model and the tools used for monitoring and conducting experiments on the model (the so-called observer);

- (iii) An architecture that allows nesting models one into another, in order to build a *hierarchy* of swarms – a swarm being a group of objects and a schedule of actions that the objects execute. One swarm can thus contain lower-level swarms whose schedules are integrated into the higher-level schedule.

A number of different simulation platforms that adhered to the SWARM protocol for simulation design have been proposed since, the most widespread being REPAST ([129]; see also [135]). However, other alternative approaches to writing agent-based models exist. Some rely on general-purpose mathematical software, like MATHEMATICA, MATLAB or MATCAD. Others, exemplified by the STARLOGO/NETLOGO experience [137], are based on the idea of an agent-based specific language.

Finally, despite the fact that ABM are most often computer models, and that the methodology could not develop in the absence of cheap and easy-to-handle personal computers, it is beneficial to remember that one of the most well-known agent-based models, the segregation model we have already described, abstracted altogether from the use of computers. As Schelling recalls, he had the original idea while seated on plane, and investigated it with paper and pencil. When he arrived home, he explained the rules of the game to his son and got him to move zins and coppers from the child's own collection on a checkerboard, looking for the results: The dynamics were sufficiently intriguing to keep my twelve-year-old engaged p. 1643 in [148].

A Recursive System Representation of Agent-Based Models

Although the complexity theory is, above all, a mathematical concept, a rather common misunderstanding about agent-based simulations is that they are not as sound as mathematical models. In an often-quoted article, Thomas Ostrom [130] argued that *computer simulation* is a third symbol system in its own right, aside *verbal description* and *mathematics*: simulation is no mathematics at all (see [79]). An intermediate level of abstraction, according to this view, characterizes computer simulations: they are more abstract than verbal descriptions, but less abstract than pure mathematics. Ostrom (p. 384 in [130]) also argued that any theory that can be expressed in either of the first two symbol systems can also be expressed in the third symbol system. This implies that there might be verbal theories, which cannot be adequately expressed in the second symbol system of mathematics, but can be in the third [79].

This view has become increasingly popular among social simulators themselves, apparently because it offers a shield to the perplexity of the mathematicians, while hinting at a sort of superiority of computer simulations. Our opinion is that both statements are simply and plainly wrong. Agent-based modeling – and more in general simulation – *is* mathematics, as we argue in this paragraph. Moreover, the conjecture that any theory can be expressed via simulation is easily contradicted: think for instance of simulating Hegel's philosophical system.

Actually, agent-based simulations are nothing else but recursive systems [59,110], where the variables s that describe at time t the state of each individual unit are determined, possibly in a stochastic way, as a function of the past states s and some parameters a :

$$s_{i,t} = f_i(s_{i,t-1}, s_{-i,t-1}; a_i, a_{-i}; t) \quad (1)$$

The individual state variables could include the memory of past values, as in the case when an unemployed person is characterized not only by the fact that he is unemployed, but also by when he last had a job. The function f_i and the parameters a_i determine individual behavior. They can possibly change over time, either in a random way or depending on some lagged variable or on higher-order parameters (as in the *Environment-Rule-Agent* framework of Gilbert and Terna [80]); when this is the case, their expression can simply be substituted for in Eq. (1). Equation (1) allows the recursive computation of the system: at any moment in time the state of each unit can be expressed as a (possibly stochastic) function of the initial values X_0 only, where X_0 includes the initial states and parameters of all the individual units:

$$s_{i,t} = g_i(X_0; t) \quad (2)$$

The aggregate state of the system is simply defined as

$$S_t = \sum_i s_{i,t} \quad (3)$$

Equilibrium in this system is described as a situation where the aggregate state S , or some other aggregate statistics Y computed on the individual states or the individual parameters are stationary.

Notice that this formalization describes both traditional dynamic micro models and agent-based simulations. In principle an agent-based model, not differently from traditional dynamic micro models, *could* be solved analytically. The problem is that the expressions involved quickly become unbearable, as (i) the level of heterogeneity, as measured by the distribution of the parame-

ters a_i and functional forms f_i , increases; (ii) the amount of interaction, as measured by the dependency of $s_{i,t}$ on $s_{-i,t-1}$, increases; (iii) the functional forms f become more complicated, e.g. with the introduction of *if-else* conditions, etc.

Hence, the resort to numerical simulation. Traditional analytical models on the other hand must take great care that the system can be solved analytically, i.e. by symbolic manipulation. Hence the use of simple functions as the omnipresent Cobb–Douglas, the assumption of homogeneous units (that can then be replaced by a RA), the choice of simple interaction processes, often mediated by a centralized coordination mechanism. However, analytical tractability alone is a poor justification of any modeling choice. As the Nobel laureate Harry Markowitz wrote, “if we restrict ourselves to models which can be solved analytically, we will be modeling for our mutual entertainment, not to maximize explanatory or predictive power” (as reported in [112]). Restricting to analytically solvable *models* – as they are called in the not sufficiently well-known paper by Axel Leijonhufvud [108] – looks dangerously close to the tale of the man who was searching for his keys under the light of a street lamp at night and, once asked if he had lost them there, he answered “No, but this is where the light is”.

Analysis of Model Behavior

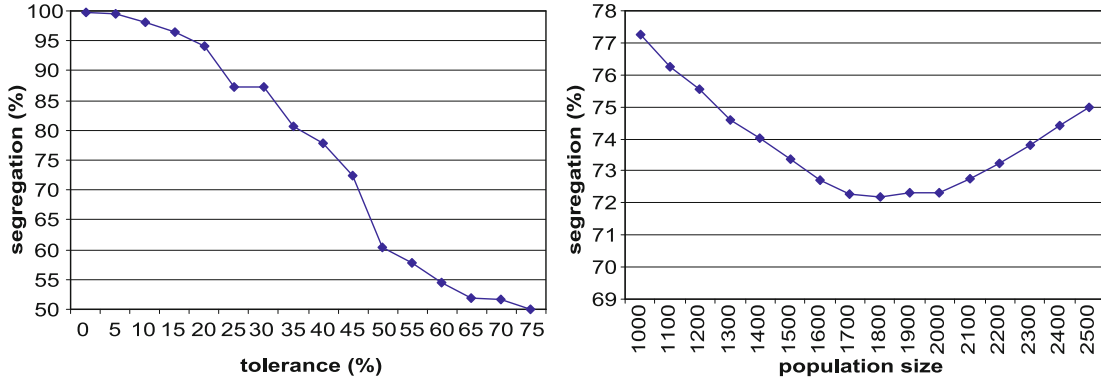
Being able to reach a close solution means that it is possible to connect inputs and outputs of the model, at any point in time, in a clear way: the *input-output transformation function*, or *reduced form*, implied by the *structural form* in which the model is expressed, is analytically obtained (e.g. the equilibrium expression of some aggregate variable of interest, as a function of the model parameters). Hence, theorems can be proved and laws expressed.

On the contrary, in a simulation model the reduced form remains unknown, and only *inductive* evidence about the input/output transformation implied by the model can be collected. Performing multiple runs of the simulation with different parameters does this. In other words, simulations suffer from the problem of stating general propositions about the dynamics of the model starting only from point observations. Since scientific explanations are generally defined as the derivation of general laws, which are able to replicate the phenomena of interests [93,94], simulations appear to be less scientific than analytical models. As Axelrod [19] points out, “like deduction, [a simulation] starts with a set of explicit assumptions. But unlike deduction, it does not prove theorems. Instead, a simulation generates data that can be analyzed

inductively”. Induction comes at the moment of explaining the behavior of the model. It should be noted that although induction is used to obtain knowledge about the behavior of a given simulation model, the use of a simulation model to obtain knowledge about the behavior of the real world refers to the logical process of abduction [109,117]. Abduction [66,132], also called *inference to the best explanation*, is a method of reasoning in which one looks for the hypothesis that would best explain the relevant evidence, as in the case when the observation that the grass is wet allows one to suppose that it rained.

Being constrained to unveil the underlying input-output transformation function by repetitively sampling the parameter space, simulations cannot prove necessity, i.e. they cannot provide in the traditional sense necessary conditions for any behavior to hold. This is because nothing excludes a priori that the system will behave in a radically different way as soon as the value of some parameter is changed, while it is generally not possible to sample all values of the parameter space. In other words, the artificial data may not be representative of all outcomes the model can produce. While analytical results are conditional on the specific hypothesis made about the *model* only, simulation results are conditional both on the specific hypothesis of the model and the specific values of the *parameters* used in the simulation runs: each run of such a model yields is a sufficiency theorem, [yet] a single run does not provide any information on the robustness of such theorems [20].

The sampling problem becomes increasingly harder as the number of the parameters increase. This has been referred to as *the curse of dimensionality* [143]. To evaluate its implications, two arguments should be considered. The first one is theoretical: if the impossibility to gain a full knowledge of the system applies to the *artificial* world defined by the simulation model, it also applies to the *real* world. The real data generating process being itself unknown, stylized facts (against which all models are in general evaluated) could in principle turn wrong, at some point in time. From an epistemological point of view, our belief that the sun will rise tomorrow remains a probabilistic assessment. The second, and more decisive, consideration is empirical: we should not worry too much about the behavior of a model for particular evil combinations of the parameters, as long as these combinations remain extremely rare (one relevant exception is when rare events are the focus of the investigation, e.g. in risk management, see [150]). If the design of the experiments is sufficiently accurate, the problem of how imprecise is the estimated input-output transformation function becomes marginal:



Agent Based Models in Economics and Complexity, Figure 3

Sensitivity analysis for the Schelling's segregation model. Segregation is measured as the share of racially similar neighbors. The reference parameter configuration is population size = 2000, tolerance level = 40%

While the curse of dimensionality places a practical upper bound on the size of the parameter space that can be checked for robustness, it is also the case that vast performance increases in computer hardware are rapidly converting what was once perhaps a fatal difficulty into a manageable one [20].

In conclusion, extensive experimentation is the only way to get a full understanding of the simulation behavior. Sampling of the parameter space can be done either systematically, i. e. by grid exploration, or randomly. Following Leombruni et al. [111], we can further distinguish between two levels at which sampling can be done: a *global* level and a *local* level. Local sampling is conducted around some specific parameter configurations of interest, by letting each parameter vary and keeping all the others unchanged. This is known as *sensitivity analysis*, and is the equivalent to the study of the partial derivatives of the input-output transformation function in an analytical model.

As an example, Fig. 3 reports a plot of the equilibrium level of segregation in the Schelling model, for decreasing values of tolerance (left panel) and increasing population density (right panel). Tolerance level is sampled in the range [0, .7] by increasing steps of .05, while population size is sampled in the range [1000, 2000] by increasing steps of 100. To get rid of random effects (in the initial residential pattern and in the choice of a different location of unsatisfied individuals), 100 runs are performed for every value of the parameter being changed, and average outcomes are reported. This gives an idea of the local effects of the two parameters around the central parameter configuration where the population size is equal to 2000 and the tolerance level is equal to 70%.

For what concerns the effect of tolerance on segregation (left panel), it should be noted that the somewhat irregular shape of the relationship is a consequence not of the sample size but of the small neighborhood individuals take into consideration (a maximum of eight adjacent cells, as we have seen), and the discretization it brings. As the effect of population size on segregation (right panel) is concerned, it may seem at a first glance counter-intuitive that segregation initially diminishes, as the population density increases. This is due to the fact that clusters can separate more if there are more free locations. Of course, nothing excludes the possibility that these marginal effects are completely different around a different parameter configuration. To check whether this is the case, it is necessary either to repeat the sensitivity analysis around other configurations, or to adopt a multivariate perspective.

Allowing all parameters to change performs *global sampling*, thus removing the reference to any particular configuration. To interpret the results of such a global analysis, a relationship between inputs and outputs in the *artificial* data can be estimated, e. g.:

$$Y = m(X_0) . \quad (4)$$

Where Y is the statistics of interest (say, the Gini coefficient of wealth), computed in equilibrium, i. e. when it has reached stationarity, and X_0 contains the initial conditions and the structural parameters of the model: $X_0 = \{s_0, a\}$. If the (not necessary unique) steady state is independent of the initial conditions, Eq. 4 simplifies to:

$$Y = m(A) . \quad (5)$$

Where A contains only the parameters of the simulation. The choice of the functional form m to be estimated, which

Agent Based Models in Economics and Complexity, Table 1

Regression results for Schelling's segregation model. Instead of repeating the experiment n times for each parameter configuration, in order to average out the random effects of the model, we preferred to test a number of different parameter configurations n times higher. Thus, population size is explored in the range [1000, 2000] by increasing steps of 10, and tolerance level is explored in the range [0, 7] by increasing steps of .05

Source	SS	df	MS
Model	666719.502	6	111119.917
Residual	12033.9282	2108	5.70869461
Total	678753.43	2114	321.075416

Number of obs = 2115
F(6, 2108) = 19465.03
Prob > F = 0.0000
R-squared = 0.9823
Adj R-squared = 0.9822
Root MSE = 2.3893

Segregation	Coef.	Std. Err.	t	P > t	[95% Conf. Interval]	
tolerance	3.379668	.0819347	41.25	0.000	3.218987	3.54035
tolerance_2	-.0655574	.0013175	-49.76	0.000	-.0681411	-.0629737
tolerance_3	.0003292	6.73e-06	48.94	0.000	.000316	.0003424
density	-23.83033	3.274691	-7.28	0.000	-30.25229	-17.40837
density_2	20.05102	2.372174	8.45	0.000	15.39897	24.70306
interaction	-.1745321	.0153685	-11.36	0.000	-.2046712	-.144393
_cons	57.31189	1.957341	29.28	0.000	53.47336	61.15041

is sometimes referred to as *metamodel* [103] is to a certain extent arbitrary, and should be guided by the usual criteria for model specification for the analysis of real data.

As an example, we performed a multivariate analysis on the artificial data coming out of the Schelling's segregation model, by letting both the population size and the tolerance threshold to vary. Overall, 2115 parameter configurations are tested. After some data mining, our preferred specification is an OLS regression of the segregation level on a third order polynomial of the *tolerance* threshold, a second order polynomial of population *density*, plus an *interaction* term given by the product of tolerance and density. The interaction term, that turns out to be highly significant, implies that the local analysis of Fig. 3 has no general validity.

The regression outcome is reported in Table 1.

Such a model allows predicting the resulting segregation level for any value of the parameters. Of course, as the complexity of the model increases (e. g. leading to multiple equilibria) finding an appropriate meta-model becomes increasingly arduous.

Finally, let's remark that the curse of dimensionality strongly suggests that the flexibility in model specification characterizing agent-based models is to be used with care, never neglecting the KISS (*Keep it simple, Stupid*) principle. Schelling's segregation model is in this respect an example of simplicity, since it has but a few parameters: this is not incoherent with the complexity approach, since it stresses how simple behavioral rules can generate very complex dynamics.

Validation and Estimation

The previous section has dealt with the problem of interpreting the behavior of an agent-based model, and we have seen that this can be done by appropriately generating and analyzing *artificial* data. We now turn to the relationship between artificial and *real* data, that is (i) the problem of choosing the parameter values in order to have the behavior of the model being as close as possible to the real data, and (ii) the decision whether a model is good enough, which often entails a judgment on "how close" as close as possible is. The first issue is referred to as the problem of *calibration* or *estimation* of the model, while the second one is known as *validation*.

Note that all models have to be understood. Thus, for agent-based models analysis of the artificial data is always an issue. However, not all models have to be estimated or validated. Some models are built with a theoretical focus (e. g. Akerlof's market for lemons), and thus comparison with the real data is not an issue – although it could be argued that some sort of evaluation is still needed, although of a different kind.

Estimation

Although the terms calibration and estimation are sometimes given slightly different meanings (e. g. [105]), we agree with Hansen and Heckman (p. 91 in [92]) that "the distinction drawn between calibrating and estimating the parameters of a model is artificial at best. Moreover, the justification for what is called *calibration* is vague and con-

fusing. In a profession that is already too segmented, the construction of such artificial distinctions is counterproductive.”

Our understanding is that, too often, calibration simply refers to a sort of rough estimation, e. g. by means of visual comparison of the artificial and real data. However, not all parameters ought to be estimated by means of formal statistical methods. Some of them have very natural real counterparts and their value is known (e. g. the interest rate): the simulation is run with empirical data. Unknown parameters have on the other had to be properly estimated.

In analytical models the reduced form coefficients, e. g. the coefficients linking output variables to inputs, can be estimated in the real data. If the model is identified, there is a one-to-one relationship between the structural and the reduced form coefficients. Thus, estimates for the structural coefficients can be recovered. In a simulation model this can't be done. However, we could compare the outcome of the simulation with the real data, and change the structural coefficient values until the distance between the simulation output and the real data is minimized. This is called *indirect inference* [85], and is also applied to analytical models e. g. when it is not possible to write down the likelihood. There are many ways to compare real and artificial data. For instance, simple statistics can be computed both in real and in artificial data, and then aggregated in a unique measure of distance. Clearly, these statistics have to be computed just once in the real data (which does not change), and once every iteration until convergence in the artificial data, which depends on the value of the structural parameters. The change in the value of the parameters of each iteration is determined according to some optimization algorithm, with the aim to minimize the distance.

In the *method of simulated moments* different order of moments are used, and then weighted to take into account their uncertainty (while the uncertainty regarding the simulated moments can be reduced by increasing the number of simulation runs, the uncertainty in the estimation of the real, population moment on the basis of real sample data cannot be avoided). The intuition behind this is to allow parameters estimated with a higher degree of uncertainty to count less, in the final measure of distance between the real and artificial data [174]. Having different weights (or no weights at all) impinges on the efficiency of the estimates, not on their consistency. If the number of moments is equal to the number of structural parameters to be estimated, the model is just-identified and the minimized distance, for the estimated values of the parameters, is 0. If the number of moments is higher than the number of parameters the model is over-identified and the minimized

distance is greater than 0. If it is lower it is under-identified. Another strategy is to estimate an *auxiliary model* both in the real and in the artificial data, and then compare the two sets of estimates obtained. The regression coefficients have the same role as the moments in the method of simulated moments: they are just a way of summarizing the data. Hence, if the number of coefficients in the auxiliary model is the same as the number of structural parameters to be estimated the model is just-identified and the minimized distance is 0. The specification of the auxiliary model is not too important. It can be proved that misspecification (e. g. omission of a relevant variable in the relationship to be estimated) only affects efficiency, while the estimates of the structural parameters remain consistent. A natural choice is of course the meta-model of Eq. 4.

Validation

A different issue is determining “how good” a model is. Of course, an answer to this question cannot be unique, but must be made in respect to some evaluation criterion. This in turn depends on the objectives of the analysis [62,102,111,164]. The need for evaluation of the model is no different in agent-based models and in traditional analytical models. However, like all simulations agent-based models require an additional layer of evaluation: the validity of the simulator (the program that simulates) relative to the model (*program validity*).

Assuming this is satisfied and the program has no bugs, Marks [121] formalizes the assessment of the *model validity* as follows: the model is said to be *useful* if it can exhibit at least some of the observed historical behaviors, *accurate* if it exhibits only behaviors that are compatible with those observed historically, and *complete* if it exhibits all the historically observed behaviors. In particular, letting \mathbf{R} be the real world output, and \mathbf{M} be the model output, four cases are possible:

- No intersection between \mathbf{R} and \mathbf{M} ($\mathbf{R} \cap \mathbf{M} = \emptyset$): the model is *useless*;
- \mathbf{M} is a subset of \mathbf{R} ($\mathbf{M} \subset \mathbf{R}$): the model is accurate, but *incomplete*;
- \mathbf{R} is a subset of \mathbf{M} ($\mathbf{M} \supset \mathbf{R}$): the model is complete, but *inaccurate* (or *redundant*, since the model might tell something about what could yet happen in the world);
- \mathbf{M} is equivalent to \mathbf{R} ($\mathbf{M} \Leftrightarrow \mathbf{R}$): the model is *complete* and *accurate*.

Of course, the selection of the relevant historical behaviors is crucial, and amounts to defining the criteria against which the model is to be evaluated. Moreover, the recogni-

tion itself of historical behavior passes through a process of analysis and simplification that leads to the identification of *stylized facts*, which are generally defined in stochastic terms. Thus, a model is eventually evaluated according to the extent to which it is able to statistically replicate the selected stylized facts.

Finally, let's note that the behavior of the model might change significantly for different values of the parameters. Hence, the process of validation always regards both the *structure* of the model and the *values* of the parameters. This explains why and how validation and estimation are connected: as we have already noted, estimation is an attempt to make the behavior of the model as close as possible to real behavior; validation is a judgment on how far the two behaviors (still) are. A model where the parameters have not been properly estimated and are e.g. simple guesses can of course be validated. However, by definition its performance can only increase should the values of the parameters be replaced with their estimates.

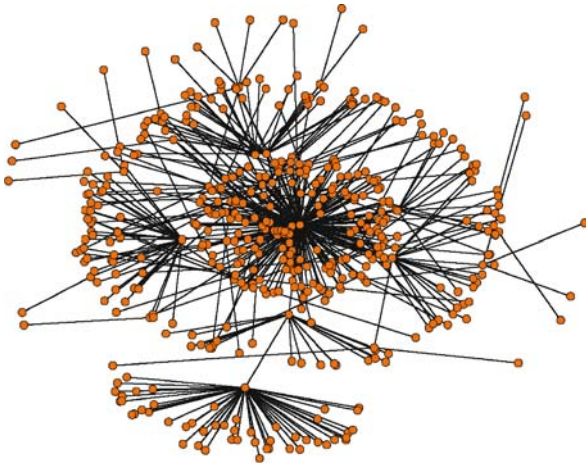
The Role of Economic Policy

Before *economics* was *political economy*. According to the classical economists, the economic science has to be used to control the real economies and steer them towards desirable outcomes. If one considers the economic system as an analogue of the physical one, it is quite obvious to look for *natural* economic policy prescriptions (*one policy fits all*). This is the approach of mainstream (neoclassical) economists. There is a widespread opinion, well summarized by Brock and Colander [33], that, with respect to the economic policy analysis of the mainstream, (i) complexity does not add anything new to the box of tools. This point needs substantial corrections (see also the reflections by Durlauf [3]). The complexity approach showed us that the age of certainty ended with the non-equilibrium revolution, exemplified by the works of Prigogine. Considering the economy as an evolving (adaptive) system we have to admit that our understanding of it is limited (there is no room for *Laplace' demon* in complexity). Individual behavioral rules evolve according to their past performance: this provides a mechanism for an endogenous change of the environment. As a consequence the rational expectation hypothesis loses significance. However, agents are still rational in that they do what they can in order not to commit systematic errors [113]. In this setting there is still room for policy intervention outside the mainstream myth of a neutral and optimal policy. Because emergent facts are transient phenomena, policy recommendations are less certain, and they should be institution and historically oriented [65,170]. In particular, it has been empha-

sized that complex systems can either be extremely fragile and turbulent (a slight modification in some minor detail brings macroscopic changes), or relatively robust and stable: in such a context, policy prescriptions ought to be case sensitive.

In a heterogeneous interacting agents environment, there is also room for an extension of the Lucas critique. It is well known that, according to it, because the underlying parameters are not policy-invariant any policy advice derived from large-scale econometric models that lack microfoundations would be misleading. The Lucas Critique implies that in order to predict the effect of a policy experiment, the so-called *deep parameters* (*preferences, technology and resource constraints*) that govern individual behavior have to be modeled. Only in this case it is possible to predict the behaviors of individuals, conditional on the change in policy, and aggregate them to calculate the macroeconomic outcome. But here is the trick: aggregation is a sum only if interaction is ignored. If non-price interactions (or other non-linearities) are important, then the interaction between agents may produce very different outcomes. Mainstream models focus on analytical solvable solutions: to get them, they have to simplify the assumptions e.g. using the RA approach or a Gaussian representation of heterogeneity. At the end, the main objective of these models is to fit the theory, not the empirics: how to explain, e.g., the scale-free network of the real economy represented in Fig. 1c by using the non interacting network of the mainstream model of Fig. 1a? At a minimum, one should recognize that the mainstream approach is a very primitive framework and, as a consequence, the economic policy recommendations derivable from it are very far from being adequate prescriptions for the real world.

Real economies are composed by millions of interacting agents, whose distribution is far from being stochastic or normal. As an example, Fig. 4 reports the distribution of the firms' trade-credit relations in the electronic-equipment sector in Japan in 2003 (see [47]). It is quite evident that there exist several hubs, i.e. firms with many connections: the distribution of the degree of connectivity is *scale free*, i.e. there are a lot of firms with one or two links, and very a few firms with a lot of connections. Let us assume the Central Authority has to prevent a *financial collapse* of the system, or the spreading of a financial crisis (the so-called *domino effect*, see e.g. [104] and [157]). Rather than looking at the average risk of bankruptcy (in power law distributions the mean may even not exist, i.e. there is an empirical mean, but it is not stable), and to infer it is a measure of the stability of the system, by means of a network analysis the economy can be analyzed in terms of differ-



Agent Based Models in Economics and Complexity, Figure 4
 Network of firms (electrical machinery and other machines sector, Japan). Source: De Masi et al. [47]

ent interacting sub-systems, and local intervention can be recommended to prevent failures and their spread.

Instead of a helicopter drop of liquidity, one can make targeted interventions to a given agent or sector of activity: Fujiwara, [72], show how to calculate the probability of going bankrupt by *solo*, i. e. because of idiosyncratic elements, or *domino* effect, i. e. because of the failure or other agents with which there exist credit or commercial links.

One of the traditional fields of applications of economic policy is *redistribution*. It should be clear that a sound policy analysis requires a framework built without the RA straight jacket. A redistributive economic policy has to take into account that individuals are different: not only they behave differently, e. g. with respect to saving propensities, but they also have different fortunes: the so-called *St. Thomas* (13:12) effect (*to anyone who has, more will be given and he will grow rich; from anyone who has not, even what he has will be taken away*), which is the road to Paradise for Catholics, and to the power-law distribution of income and wealth for the econophysicists.

Gaffeo et al. [75], show that there is a robust link between firms' size distribution, their growth rate and GDP growth. This link determines the distributions of the amplitude frequency, size of recessions and expansion *etc.* Aggregate firms' size distribution can be well approximated by a *power law* [21,74], while sector distribution is still right skewed, but without scale-free characteristics [22]. Firms' growth rates are far from being *normal*: in the central part of the distribution they are tent shaped with very fat tails. Moreover, empirical evidence shows that exit is an inverse function of firms' age and size and proportional to financial fragility. In order to reduce the

volatility of fluctuations, policy makers should act on the firms' size distribution, allowing for a growth of their capitalization, their financial solidity and wealth redistribution [48,49]. Since these emerging facts are policy sensitive, if the aggregate parameters change the shape of the curve will shift as well.

Differently from Keynesian economic policy, which theorizes aggregate economic policy tools, and mainstream neoclassical economics, which prescribes individual incentives because of the Lucas critique but ignores interaction which is a major but still neglected part of that critique, the ABM approach proposes a bottom up analysis. What generally comes out is not a one-size-fits-all policy since it depends on the general as well as the idiosyncratic economic conditions; moreover, it generally has to be conducted at different levels (from micro to meso to macro). In short, ABM can offer new answers to old unresolved questions, although it is still in a far too premature stage to offer definitive tools.

Future Directions

We have shown that mainstream approach to economics uses a methodology [71], which is so weak in its assumptions as to have been repeatedly ridiculed by the epistemologists [37], and dates back to the classical mechanical approach, according to which reductionism is possible. We have also seen that adopting the reductionist approach in economics is to say that agents *do not interact directly*: this is a very implausible assumption (billions of Robinson Crusoes who never meet Friday) and cannot explain the *emerging* characteristics of our societies, as witnessed by the empirical evidence. The reductionist approach of the mainstream is also theoretically incoherent, since it can be given no sound microfoundations [8,100].

In the fourth edition of his *Principles*, Marshall wrote, "The Mecca of the economist is biology". What he meant to say was that, because economics deals with learning agents, evolution and change are the *granum salis* of our economic world. A theory built upon the issue of allocations of given quantities is not well equipped for the analysis of change. This allocation can be optimal only if there are no externalities (increasing returns, non-price interactions etc.) and information is complete, as in the case of the *invisible hand* parabola. In the history of science, there is a passage from a view emphasizing centralized intelligent design to a view emphasizing *self organized criticality* [27], according to which a system with many heterogeneous interacting agents reaches a statistical aggregate equilibrium, characterized by the appearance of some (often scale free) stable distributions. These distributions are no longer op-

timal or efficient according to some welfare criterion: they are simply the natural outcome of individual interaction.

Because of the above-mentioned internal and external inconsistencies of the mainstream approach, a growing strand of economists is now following a different methodology based upon the analysis of systems with many heterogeneous interacting agents. Their interaction leads to empirical regularities, which emerge from the system as a whole and cannot be identified by looking at any single agent in isolation: these emerging properties are, according to us, the main distinguishing feature of a complex system. The focus on interaction allows the scientist to abandon the heroic and unrealistic RA framework, in favor of the ABM approach, the *science of complexity* popularized by the SFI. Where did the Santa Fe approach go? Did it really bring a revolution in social science, as some of its initial proponents ambitiously believed? Almost twenty years and two “The economy as an evolving complex system” volumes later, Blume and Durlauf summarized this intellectual Odyssey as follows:

“On some levels, there has been great success. Much of the original motivation for the Economics Program revolved around the belief that economic research could benefit from an injection of new mathematical models and new substantive perspectives on human behavior. [...] At the same time, [...] some of the early aspirations were not met” (Chaps. 1–2 in [29]).

It is probably premature to try to give definitive answers. For sure, ABM and the complexity approach are a very tough line of research whose empirical results are very promising (see e.g., Chaps. 2–3 in [77]). Modeling an agent-based economy however remains in itself a complex and complicated adventure.

Bibliography

- Allen PM, Engelen G, Sanglier M (1986) Towards a general dynamic model of the spatial evolution of urban systems. In: Hutchinson B, Batty M (eds) *Advances in urban systems modelling*. North-Holland, Amsterdam, pp 199–220
- Anderson PW (1972) More is different. *Science* 177:4047
- Anderson PW (1997) Some thoughts about distribution in economics. In: Arthur WB, Durlauf SN, Lane D (eds) *The economy as an evolving complex system II*. Addison-Wesley, Reading
- Anderson PW, Arrow K, Pines D (eds) (1988) *The economy as an evolving complex system*. Addison-Wesley, Redwood
- Aoki M (1996) *New approaches to macroeconomic modelling: evolutionary stochastic dynamics, multiple equilibria, and externalities as field effects*. Cambridge University Press, Cambridge
- Aoki M (2002) *Modeling aggregate behaviour and fluctuations in economics*. Cambridge University Press, Cambridge
- Aoki M, Yoshikawa H (2006) *Reconstructing macroeconomics*. Cambridge University Press, Cambridge
- Aoki M, Yoshikawa H (2007) *Non-self-averaging in economic models*. Economics Discussion Papers No. 2007-49, Kiel Institute for the World Economy
- Arrow KJ (1959) Towards a theory of price adjustment. In: Abramovits M (ed) *Allocation of economic resources*. Stanford University Press, Stanford
- Arrow KJ (1963) *Social choice and individual values*, 2nd edn. Yale University Press, New Haven
- Arrow KJ (1964) The role of securities in the optimal allocation of risk-bearing. *Rev Econ Stud* 31:91–96
- Arrow KJ (1971) A utilitarian approach to the concept of equality in public expenditures. *Q J Econ* 85(3):409–415
- Arrow KJ (1994) Methodological individualism and social knowledge. *Am Econ Rev* 84:1–9
- Arrow KJ, Debreu G (1954) Existence of an equilibrium for a competitive economy. *Econometrica* 22:265–290
- Arthur WB (2000) Complexity and the economy. In: Colander D (ed) *The complexity vision and the teaching of economics*. Edward Elgar, Northampton
- Arthur WB (2006) Out-of-equilibrium economics and agent-based modeling. In: Tesfatsion L, Judd KL (eds) *Handbook of computational economics*, vol 2: *Agent-Based Computational Economics*, ch 32. North-Holland, Amsterdam, pp 1551–1564
- Arthur WB, Durlauf S, Lane D (eds) (1997) *The economy as an evolving complex system II*. Addison-Wesley, Reading
- Askenazi M, Burkhart R, Langton C, Minar N (1996) *The swarm simulation system: A toolkit for building multi-agent simulations*. Santa Fe Institute, Working Paper, no. 96-06-042
- Axelrod R (1997) Advancing the art of simulation in the social sciences. In: Conte R, Hegselmann R, Terna P (eds) *Simulating social phenomena*. Springer, Berlin, pp 21–40
- Axtell RL (2000) Why agents? On the varied motivations for agent computing in the social sciences. *Proceedings of the Workshop on Agent Simulation: Applications, Models and Tools*. Argonne National Laboratory, Chicago
- Axtell RL (2001) Zipf distribution of US firm sizes. *Science* 293:1818–1820
- Axtell RL, Gallegati M, Palestrini A (2006) Common components in firms’ growth and the scaling puzzle. Available at SSRN: <http://ssrn.com/abstract=1016420>
- Bak P (1997) *How nature works. The science of self-organized criticality*. Oxford University Press, Oxford
- Batten DF (2000) *Discovering artificial economics*. Westview Press, Boulder
- Barone E (1908) Il ministro della produzione nello stato collettivista. *G Econ* 267–293, 391–414
- Beinhocker ED (2006) *The origin of wealth: Evolution, complexity, and the radical remaking of economics*. Harvard Business School Press, Cambridge
- Bénabou R (1996) Heterogeneity, stratification and growth: Macroeconomic implications of community structure and school finance. *Am Econ Rev* 86:584–609
- Blanchard OJ, Kiyotaki N (1987) Monopolistic competition and the effects of aggregate demand. *Am Econ Rew* 77:647–666
- Blume L, Durlauf S (eds) (2006) *The economy as an evolving*

- complex system, III. Current perspectives and future directions. Oxford University Press, Oxford
30. Blundell R, Stoker TM (2005) Heterogeneity and aggregation. *J Econ Lit* 43:347–391
 31. Bowles S (1998) Endogenous preferences: The cultural consequences of markets and other economic institutions. *J Econ Lit* 36:75–111
 32. Brock WA (1999) Scaling in economics: a reader's guide. *Ind Corp Change* 8(3):409–446
 33. Brock WA, Colander D (2000) Complexity and policy. In: Colander D (ed) *The complexity vision and the teaching of economics*. Edward Elgar, Northampton
 34. Brock WA, Durlauf SN (2001) Discrete choice with social interactions. *Rev Econ Stud* 68:235–260
 35. Brock WA, Durlauf SN (2005) Social interactions and macroeconomics. UW-Madison, SSRI Working Papers n.5
 36. Caballero RJ (1992) A Fallacy of composition. *Am Econ Rev* 82:1279–1292
 37. Calafati AG (2007) Milton Friedman's epistemology UPM working paper n.270
 38. Caldarelli G (2006) Scale-free networks. *Complex webs in nature and technology*. Oxford University Press, Oxford
 39. Clower RW (1965) The keynesian counterrevolution: A theoretical appraisal. In: Hahn F, Brechling F (eds) *The theory of interest rates*. Macmillan, London
 40. Cohen A, Harcourt G (2003) What ever happened to the Cambridge capital theory controversies. *J Econ Perspect* 17:199–214
 41. Cole HL, Mailath GJ, Postlewaite A (1992) Social norms, savings behaviour, and growth. *J Political Econ* 100(6):1092–1125
 42. Cooper RW (1999) Coordination games: Complementarities and macroeconomics. Cambridge University Press, Cambridge
 43. Crutchfield J (1994) Is anything ever new? Considering emergence. In: Cowan G, Pines D, Meltzer D (eds) *Complexity: Metaphors, models, and reality*. Addison-Wesley, Reading, pp 515–537
 44. Davis JB (2006) The turn in economics: Neoclassical dominance to mainstream pluralism? *J Inst Econ* 2(1):1–20
 45. Debreu G (1959) *The theory of value*. Wiley, New York
 46. Debreu G (1974) Excess demand functions. *J Math Econ* 1:15–23
 47. De Masi G, Fujiwara Y, Gallegati M, Greenwald B, Stiglitz JE (2008) Empirical evidences of credit networks in Japan. mimeo
 48. Delli Gatti D, Di Guilmi C, Gaffeo E, Gallegati M, Giulioni G, Palestrini A (2004) Business cycle fluctuations and firms' size distribution dynamics. *Adv Complex Syst* 7(2):1–18
 49. Delli Gatti D, Di Guilmi C, Gaffeo E, Gallegati M, Giulioni G, Palestrini A (2005) A new approach to business fluctuations: Heterogeneous interacting agents, scaling laws and financial fragility. *J Econ Behav Organ* 56(4):489–512
 50. Denzau AT, North DC (1994) Shared mental models: Ideologies and institutions. *Kyklos* 47(1):3–31
 51. Descartes R (1637) *Discours de la méthode pour bien conduire sa raison, et chercher la vérité dans les sciences*, tr. *Discourse on Method and Meditations*. The Liberal Arts Press, 1960, New York
 52. Dorogovtsev SN, Mendes JFF (2003) Evolution of networks from biological nets to the internet and the WWW. Oxford University Press, Oxford
 53. Di Guilmi C, Gallegati M, Landini S (2007) Economic dynamics with financial fragility and mean-field interaction: a model. arXiv:0709.2083
 54. Durlauf SN (1993) Nonergodic economic growth. *Rev Econ Stud* 60:349–366
 55. Durlauf SN (1997) What should policymakers know about economic complexity? *Wash Q* 21(1):157–165
 56. Durlauf SN, Young HP (2001) *Social dynamics*. The MIT Press, Cambridge
 57. Edgeworth FY (1925) The pure theory of monopoly In: *Papers relating to political economy*. McMillan, London
 58. Epstein JM (1999) Agent-based computational models and generative social science. *Complexity* 4:41–60
 59. Epstein JM (2006) Remarks on the foundations of agent-based generative social science. In: Tesfatsion L, Judd KL (eds) *Handbook of computational economics. Agent-based computational economics*, vol 2, ch 34. North-Holland, Amsterdam, pp 1585–1604
 60. Epstein JM (2006) *Generative social science: Studies in agent-based computational modeling*. Princeton University Press, New York
 61. Epstein JM, Axtell RL (1996) *Growing artificial societies: Social science from the bottom up*. The MIT Press, Cambridge
 62. Fagiolo G, Moneta A, Windrum P (2007) A critical guide to empirical validation of agent-based models in economics: Methodologies, procedures, and open problems. *Comput Econ* 30:195–226
 63. Farley R (1986) The residential segregation of blacks from whites: Trends, causes, and consequences. In: US Commission on Civil Rights, *Issues in housing discrimination*. US Commission on Civil Rights
 64. Feller W (1957) *An introduction to probability. Theory and its applications*. Wiley, New York
 65. Finch J, Orillard M (eds) (2005) *Complexity and the economy: Implications for economy policy*. Edward Elgar, Cheltenham
 66. Flach PA, Kakas AC (eds) (2000) *Abduction and induction. Essays on their relation and integration*. Kluwer, Dordrecht
 67. Flake GW (1998) *The computational beauty of nature*. The MIT Press, Cambridge
 68. Foellmer H (1974) Random economies with many interacting agents. *J Math Econ* 1:51–62
 69. Forni M, Lippi M (1997) *Aggregation and the micro-foundations of microeconomics*. Oxford University Press, Oxford
 70. Frazer J (1995) *An evolutionary architecture*. Architectural Association Publications, London
 71. Friedman M (1953) *Essays in positive economics*. University of Chicago Press, Chicago
 72. Fujiwara Y (2006) Proceedings of the 9th Joint Conference on Information Sciences (JCIS), *Advances in Intelligent Systems Research Series*. Available at http://www.atlantis-press.com/publications/aisr/jcis-06/index_jcis
 73. Gabaix X (2008) Power laws in Economics and Finance, 11 Sep 2008. Available at SSRN: <http://ssrn.com/abstract=1257822>
 74. Gaffeo E, Gallegati M, Palestrini A (2003) On the size distribution of firms, additional evidence from the G7 countries. *Phys A* 324:117–123
 75. Gaffeo E, Russo A, Catalano M, Gallegati M, Napoletano M (2007) Industrial dynamics, fiscal policy and R&D: Evi-

- dence from a computational experiment. *J Econ Behav Organ* 64:426–447
76. Gallegati M (1993) Composition effects and economic fluctuations. *Econ Lett* 44(1–2):123–126
 77. Gallegati M, Delli Gatti D, Gaffeo E, Giulioni G, Palestini A (2008) *Emergent macroeconomics*. Springer, Berlin
 78. Gallegati M, Palestini A, Delli Gatti D, Scalas E (2006) Aggregation of heterogeneous interacting agents: The variant representative agent framework. *J Econ Interact Coord* 1(1):5–19
 79. Gilbert N (ed) (1999) *Computer simulation in the social sciences*, vol 42. Sage, Thousand Oaks
 80. Gilbert N, Terna P (2000) How to build and use agent-based models in social science. *Mind Soc* 1:57–72
 81. Gilbert N, Troitzsch K (2005) *Simulation for the social scientist*. Open University Press, Buckingham
 82. Gintis H (2007) The dynamics of general equilibrium. *Econ J* 117:1280–1309
 83. Glaeser E, Sacerdote B, Scheinkman J (1996) Crime and social interactions. *Q J Econ* 111:507–548
 84. Glaeser J, Dixit J, Green DP (2002) Studying hate crime with the internet: What makes racists advocate racial violence? *J Soc Issues* 58(122):177–194
 85. Gourieroux C, Monfort A (1997) *Simulation-based econometric methods*. Oxford University Press, Oxford
 86. Greenwald B, Stiglitz JE (1986) Externalities in economies with imperfect information and incomplete markets. *Q J Econ* 101(2):229–264
 87. Grossman SJ, Stiglitz JE (1976) Information and competitive price systems. *Am Econ Rev* 66:246–253
 88. Grossman SJ, Stiglitz JE (1980) On the impossibility of informationally efficient markets. *Am Econ Rev* 70(3):393–408
 89. Guesnerie R (1993) Successes and failures in coordinating expectations. *Eur Econ Rev* 37:243–268
 90. Hahn F (1982) *Money and inflation*. Blackwell, Oxford
 91. Haken H (1983) *Synergetics. Nonequilibrium phase transitions and social measurement*, 3rd edn. Springer, Berlin
 92. Hansen L, Heckman J (1996) The empirical foundations of calibration. *J Econ Perspect* 10:87–104
 93. Hempel CV (1965) *Aspects of scientific explanation*. Free Press, London
 94. Hempel CV, Oppenheim P (1948) Studies in the logic of explanation. *Philos Sci* 15:135–175
 95. Hildenbrand W, Kirman AP (1988) *Equilibrium analysis: Variations on the themes by edgeworth and walras*. North-Holland, Amsterdam
 96. Horgan J (1995) From complexity to perplexity. *Sci Am* 272:104
 97. Horgan J (1997) *The end of science: Facing the limits of knowledge in the twilight of the scientific age*. Broadway Books, New York
 98. Jerison M (1984) Aggregation and pairwise aggregation of demand when the distribution of income is fixed. *J Econ Theory* 33(1):1–31
 99. Kirman AP (1992) Whom or what does the representative individual represent. *J Econ Perspect* 6:117–136
 100. Kirman AP (1996) Microfoundations – built on sand? A review of Maarten Janssen’s microfoundations: A Critical Inquiry. *J Econ Methodol* 3(2):322–333
 101. Kirman AP (2000) Interaction and markets. In: Gallegati M, Kirman AP (eds) *Beyond the representative agent*. Edward Elgar, Cheltenham
 102. Kleijnen JPC (1998) Experimental design for sensitivity analysis, optimization, and validation of simulation models. In: Banks J (ed) *Handbook of simulation*. Wiley, New York, pp 173–223
 103. Kleijnen JPC, Sargent RG (2000) A methodology for the fitting and validation of metamodels in simulation. *Eur J Oper Res* 120(1):14–29
 104. Krugman P (1998) Bubble, boom, crash: theoretical notes on Asia’s crisis. mimeo
 105. Kydland FE, Prescott EC (1996) The computational experiment: An econometric tool. *J Econ Perspect* 10:69–85
 106. Lavoie D (1989) Economic chaos or spontaneous order? Implications for political economy of the new view of science. *Cato J* 8:613–635
 107. Leibenstein H (1950) Bandwagon, snob, and veblen effects in the theory of consumers’ demand. *Q J Econ* 64:183–207
 108. Leijonhufvud A (1973) Life among the econ. *Econ Inq* 11:327–337
 109. Leombruni R (2002) The methodological status of agent-based simulations, LABORatorio Revelli. Working Paper No. 19
 110. Leombruni R, Richiardi MG (2005) Why are economists sceptical about agent-based simulations? *Phys A* 355:103–109
 111. Leombruni R, Richiardi MG, Saam NJ, Sonnessa M (2005) A common protocol for agent-based social simulation. *J Artif Soc Simul* 9:1
 112. Levy M, Levy H, Solomon S (2000) Microscopic simulation of financial markets. In: *From Investor Behavior to Market Phenomena*. Academica Press, New York
 113. Lewontin C, Levins R (2008) *Biology under the influence: Dialectical essays on the coevolution of nature and society*. Monthly Review Press, US
 114. Lucas RE (1976) Econometric policy evaluation: A critique. *Carnegie-Rochester Conference Series*, vol 1, pp 19–46
 115. Lucas RE (1987) *Models of business cycles*. Blackwell, New York
 116. Lucas RE, Sargent T (1979) After keynesian macroeconomics. *Fed Reserv Bank Minneap Q Rev* 3(2):270–294
 117. Magnani L, Belli E (2006) Agent-based abduction: Being rational through fallacies. In: Magnani L (ed) *Model-based reasoning in science and engineering*, *Cognitive Science, Epistemology, Logic*. College Publications, London, pp 415–439
 118. Manski CF (2000) Economic analysis of social interactions. *J Econ Perspect* 14:115–136
 119. Mantel R (1974) On the characterization of aggregate excess demand. *J Econ Theory* 7:348–353
 120. Mantegna RN, Stanley HE (2000) *An introduction to econophysics*. Cambridge University Press, Cambridge
 121. Marks RE (2007) Validating Simulation Models: A general framework and four applied examples. *Comput Econ* 30:265–290
 122. May RM (1976) Simple mathematical models with very complicated dynamics. *Nature* 261:459–467
 123. Mas-Colell A, Whinston MD, Green J (1995) *Microeconomic theory*. Oxford University Press, Oxford
 124. Miller JH, Page SE (2006) *Complex adaptive systems: An introduction to computational models of social life*. Princeton University Press, New York
 125. Mirowski P (1989) *More heat than light*. Cambridge University Press, Cambridge
 126. Muth RF (1986) The causes of housing segregation. *US Com-*

- mission on Civil Rights, Issues in Housing Discrimination. US Commission on Civil Rights
127. Nagel E (1961) *The structure of science*. Routledge and Paul Kegan, London
 128. Nicolis G, Prigogine I (1989) *Exploring complexity: An introduction*. WH Freeman, New York
 129. North MJ, Howe TR, Collier NT, Vos JR (2005) Repast symphony runtime system. In: Macal CM, North MJ, Sallach D (eds) *Proceedings of the agent 2005 Conference on Generative Social Processes, Models, and Mechanisms*, 13–15 Oct 2005
 130. Ostrom T (1988) Computer simulation: the third symbol system. *J Exp Soc Psychol* 24:381–392
 131. Page S (1999) Computational models from A to Z. *Complexity* 5:35–41
 132. Peirce CS (1955) *Abduction and induction*. In: J Buchler (ed) *Philosophical writings of peirce*. Dover, New York
 133. Phelan S (2001) What is complexity science, really? *Emergence* 3:120–136
 134. Pollack R (1975) Interdependent preferences. *Am Econ Rev* 66:309–320
 135. Railsback SF, Lytinen SL, Jackson SK (2006) Agent-based simulation platforms: Review and development recommendations. *Simulation* 82:609–623
 136. Rappaport S (1996) Abstraction and unrealistic assumptions in economics. *J Econ Methodol* 3(2):215–36
 137. Resnick M (1994) *Turtles, termites and traffic jams: Explorations in massively parallel microworlds*. MIT, Cambridge
 138. Richter MK, Wong K (1999) Non-computability of competitive equilibrium. *Econ Theory* 14:1–28
 139. Riess Rull V (1995) Models with heterogeneous agents. In: Cooley TF (ed) *Frontiers of business cycle research*. Princeton University Press, New York
 140. Rosser JB (1999) On the complexities of complex economic dynamics. *J Econ Perspect* 13:169–192
 141. Rosser JB (2000) Integrating the complexity vision into the teaching of mathematical economics. In: Colander D (ed) *The complexity vision and the teaching of economics*. Edward Elgar, Cheltenham, pp 209–230
 142. Rosser JB (2003) *Complexity in economics*. Edward Elgar, Cheltenham
 143. Rust J (1997) Using randomization to break the curse of dimensionality. *Econometrica* 65:487–516
 144. Saari DG (1995) Mathematical complexity of simple economics. *Notices Am Math Soc* 42:222–230
 145. Schelling TC (1969) Models of segregation. *Am Econ Rev* 59:488–493
 146. Schelling TC (1971) Dynamic models of segregation. *J Math Sociol* 1:143–186
 147. Schelling TC (1978) *Micromotives and macrobehaviour*. W.W. Norton, New York
 148. Schelling TC (2006) Some fun, thirty-five years ago. In: Tesfatsion L, Judd KL (eds) *Handbook of computational economics. Agent-based computational economics, vol 2, ch 37*. North-Holland, Amsterdam, pp 1639–1644
 149. Schumpeter JA (1960) *History of economic analysis*. Oxford University Press, Oxford
 150. Segre-Tossani L, Smith LM (2003) *Advanced modeling, visualization, and data mining techniques for a new risk landscape*. Casualty Actuarial Society, Arlington, pp 83–97
 151. Semmler W (2005) Introduction (multiple equilibria). *J Econ Behav Organ* 57:381–389
 152. Shy O (2001) *The economics of network industries*. Cambridge University Press, Cambridge
 153. Smith A (1776/1937) *The wealth of nations*. Random House, New York
 154. Solomon S (2007) *Complexity roadmap*. Institute for Scientific Interchange, Torino
 155. Sonnenschein H (1972) Market excess demand functions. *Econometrica* 40:549–563
 156. Stiglitz JE (1992) Methodological issues and the new keynesian economics. In: Vercelli A, Dimitri N (eds) *Macroeconomics: A survey of research strategies*. Oxford University Press, Oxford, pp 38–86
 157. Stiglitz JE (2002) *Globalization and its discontents*. Northon, New York
 158. Stoker T (1995) Empirical approaches to the problem of aggregation over individuals. *J Econ Lit* 31:1827–1874
 159. Tesfatsion L (ed) (2001) *Special issue on agent-based computational economics*. *J Econ Dyn Control* 25
 160. Tesfatsion L (ed) (2001) *Special issue on agent-based computational economics*. *Comput Econ* 18
 161. Tesfatsion L (2001) Agent-based computational economics: A brief guide to the literature. In: Michie J (ed) *Reader's guide to the social sciences*. Fitzroy-Dearborn, London
 162. Tesfatsion L (2002) Agent-based computational economics: Growing economies from the bottom up. *Artif Life* 8:55–82
 163. Tesfatsion L (2006) Agent-based computational economics: A constructive approach to economic theory. In: Tesfatsion L, Judd KL (eds) *Handbook of computational economics. Agent-based computational economics, vol 2, ch 16*. North-Holland, Amsterdam, pp 831–880
 164. Troitzsch KG (2004) Validating simulation models. In: Horton G (ed) *Proceedings of the 18th european simulation multi-conference. Networked simulations and simulation networks*. SCS Publishing House, Erlangen, pp 265–270
 165. Vriend NJ (1994) A new perspective on decentralized trade. *Econ Appl* 46(4):5–22
 166. Vriend NJ (2002) Was Hayek an ace? *South Econ J* 68:811–840
 167. Velupillai KV (2000) *Computable economics*. Oxford University Press, Oxford
 168. Velupillai KV (2002) Effectivity and constructivity in economic theory. *J Econ Behav Organ* 49:307–325
 169. Velupillai KV (2005) The foundations of computable general equilibrium theory. In: *Department of Economics Working Papers No 13*. University of Trento
 170. Velupillai KV (2007) The impossibility of an effective theory of policy in a complex economy. In: Salzano M, Colander D (eds) *Complexity hints for economic policy*. Springer, Milan
 171. von Hayek FA (1948) *Individualism and economic order*. University of Chicago Press, Chicago
 172. von Mises L (1949) *Human action: A treatise on economics*. Yale University Press, Yale
 173. Wilensky U (1998) *NetLogo segregation model*. Center for connected learning and computer-based modeling. Northwestern University, Evanston. <http://ccl.northwestern.edu/netlogo/models/Segregation>
 174. Winker P, Gilli M, Jeleskovic V (2007) An objective function for simulation based inference on exchange rate data. *J Econ Interact Coord* 2:125–145
 175. Wooldridge M (2001) *An introduction to multiagent systems*. Wiley, New York

Aggregation Operators and Soft Computing

VICENÇ TORRA

Institut d'Investigació en Intelligència Artificial – CSIC,
Bellaterra, Spain

Article Outline

Glossary

Definition of the Subject

Introduction

Applications

Aggregation, Integration and Fusion

Aggregation Operators and Their Construction

Some Aggregation Operators

Future Directions

Bibliography

Glossary

Information integration The whole process of obtaining some information from different sources and then using this information to achieve a concrete task.

Information fusion A general term that defines the whole area that studies techniques and methods to combine information for achieving a particular goal.

Aggregation operators Particular operators that are actually used for combining the information.

Definition of the Subject

Aggregation operators are the particular functions used for combining information in systems where several information sources have to be taken into consideration for achieving a particular goal.

Formally, aggregation operators are the particular techniques of information fusion which can be mathematically expressed in a relatively simple way. They are part of the more general process of information integration. That is the process that goes from the acquisition of data to the accomplishment of the final task.

As current systems (from search engines to vision and robotics systems) need to consider information from different sources, aggregation operators are of major importance. Such operators permit one to reduce the quantity of information and improve its quality.

Introduction

Information fusion is a broad field that studies computational methods and techniques for combining data and in-

formation. At present, different methods have been developed. Differences are on the type of data available and on the assumptions data and information sources (data suppliers) satisfy. For example, methods exist for combining numerical data (e. g. readings from sensors) as well as for combining structured information (e. g. partial plans obtained from AI planners). Besides, the complexity of the methods ranges from simple functions (as the arithmetic mean or the voting technique) that combine a few data to more complex ones (as procedures for dealing with satellite images) that combine a huge burden of data.

Soft computing is an area in computer science that encompasses a few techniques with the goal of solving complex problems when there are no analytic or complete solutions. Two typical problems are the control of a variable (when standard methods, as PID controllers, are not applicable) and the optimization of a function (when standard methods, as quadratic programming, cannot be applied because the objective function to be optimized is highly non linear). Among the techniques in soft computing, we can underline genetic algorithms and evolutionary computation, neural networks, and fuzzy sets and fuzzy systems [21].

Some of the techniques included in soft computing have been used for combining and fusing data. This is the case of neural networks. It is well known that neural networks (when applied in a supervised manner) can be used to build a system that reproduces up to some extent the behavior of a set of examples. So, in the case of aggregation and data fusion, they can be used to build models when we have a data set consisting on some examples (pairs of input data, expected outcome) that are to be reproduced by the system. Aggregation operators, which can also be seen from the soft computing perspective, follow a different principle. They are defined mathematically and are characterized in terms of their properties.

Aggregation operators are often studied in the framework of fuzzy sets and fuzzy systems, one of the components of soft computing. Such operators, when operating on the $[0, 1]$ interval are closely related to with t -norms and t -conorms, the basic functions in fuzzy logic to model conjunction and disjunction. Formally, t -norms and t -conorms are used in fuzzy logic to combine the membership degrees either in a conjunctive or disjunctive way. In this setting aggregation operators are also used to combine information but additional flexibility is permitted in the combination. Weighted minimum, and the fuzzy integrals are some examples of aggregation operators that are typically studied in this framework.

In this work, we present an overview of the area of information fusion and we focus on the aggregation oper-

ators. We review a few of them with some of their properties. We focus on those that have a tight link with soft computing, and especially on the ones that belong to the field of fuzzy sets and fuzzy logic.

Applications

Aggregation operators are used in a large variety of contexts and applications [8]. We underline a few of them below. For example, there are applications in economy (to define all kind of indices as e. g. the retail price index), biology (to combine DNA sequences and classifications, either expressed as dendrograms – trees – or partitions [7,12]), in education (to rate students, universities and educational systems [8]), in bibliometrics (to evaluate the quality of the researchers and rate them by means of citation indices as e. g. the number of citations or the h index [16,25]), and in computer science.

In computer science, aggregation operators are used for e. g. evaluating hardware and software, and for fusing preferences. In the first case, different criteria or performance measures are combined to define an index. For example, performance measures for evaluating runtime systems are defined as the aggregation of the execution times of different benchmarks. Some means (e. g. geometric means) are used for such aggregation. In the second case, different sets of preferences are fused into a new aggregated one. For example, meta-search engines proceed in this way aggregating the results of different search engines. Note that the result of each search engine is a preference on the links and the final ranking is another preference that is built as an aggregation of the initial preferences.

In addition, aggregation operators and information fusion are also used in artificial intelligence, robotics and vision. Knowledge integration, knowledge revision or belief merging for knowledge based systems can also be seen as a case of information fusion at large. In this case, a knowledge base, which is defined using e. g. description logic in terms of propositions, is checked to detect incoherences and, eventually, to correct them. In vision, fusion is used e. g. for improving the quality of images. Fusion is also used in robotics for a similar purpose. In particular, it is used for improving the quality of the information obtained from the sensors [18].

The different uses of information fusion in artificial intelligence and robotics can be classified into two main categories. On the one hand, we have the uses related to making decisions and, on the other hand, the uses related with the obtainment of a better description of the application domain (so that the system has a better understanding).

Decision making typically corresponds to the process of selecting the best alternative. Difficulties in decision making are found when each alternative is evaluated using different (conflicting) criteria. This case corresponds to the multi criteria decision making framework [13,20]. In such framework, depending on the criterion used in the selection, different alternatives are selected. That is, different Pareto optimal solutions can be selected according to the criteria. One approach to solve this situation is to use aggregation operators to combine the degrees of satisfaction for each criteria into a new aggregated degree of satisfaction (for a new *aggregated criterion*). Then selection uses this aggregated criterion to select the best rated alternative. Another approach consists of considering different preferences (one for each criterion) and then such preferences are aggregated into a new aggregated preference. Then, selection is based on the aggregated preference.

A different case, also related to decision making, is when the alternative to be selected (e. g. the solution of a problem) is constructed from the other ones. For example, several partial solutions are retrieved or obtained by a set of subsystems (maybe because such solutions have been applied before with success into similar problems, as in case-based reasoning). Then, if none of them is completely appropriate in the new situation, the new solution is built from such partial solutions. Some applications of case-based reasoning can be seen from this perspective. This is also the case in plan merging, where partial plans are combined to build the final plan. Another case corresponds to the ensemble methods in machine learning. Such methods assign an outcome (e. g., a class in a classification problem) for a given data taking into account the outcomes of different subsystems (e. g., different classifiers based on different assumptions constructed from the data in the learning set). Then, a final outcome is decided on the basis of the outcomes of the subsystems (e. g., using a voting strategy).

The second use of information fusion and aggregation operators is for improving the understanding of the application domain. This tries to solve some of the problems related to data of low quality, e. g., data that lacks accuracy due to errors caused by the information source or due to the errors introduced in the transmission. Another difficulty is that the data from a single source might describe only part of the application's domain, and thus, several sources are needed to have a more comprehensive description of the domain, e. g. either several video cameras or a sequence of images from the same video camera are needed to have a full description of a room or a building. Knowledge integration and knowledge revision can be seen from this perspective as the knowledge in a system

needs to be updated and refined when the system is under execution.

The different uses of aggregation operators and the large number of applications have stimulated the definition, study and characterization of aggregation operators on different types of data. Due to this, there are nowadays operators for a large number of different domains. For example, there are operators to combine data in nominal scales (no ordering or structure exists between the elements to be aggregated), ordinal scales (there is ordering between pairs of categories but no other operators are allowed), numerical data (ratio and measure scales), partitions (input data correspond to a partition of the elements of a given set), fuzzy partitions (as in the previous case, but fuzzy partitions are considered), dendrograms (e. g., tree-like structures for classifying species), DNA sequences (i. e., sequences of A, T, C, and G), images (in remote sensing applications), and logical propositions (in knowledge integration).

Aggregation, Integration and Fusion

In the context of information fusion there are three terms that are in common use. The terms are information integration, information fusion, and aggregation operator. Although these terms are often used as synonymous their meaning is not exactly equivalent. We define them below.

Information integration This corresponds to the whole process of obtaining some information from different sources and then using this information to achieve a concrete task. It is also possible that instead of information from several sources, information is extracted from the same source but at different times.

Information fusion This is a general term that defines the whole area that studies techniques and methods to combine information for achieving a particular goal. It is also understood as the actual process of combining the information (a particular algorithm that uses the data to obtain a single datum).

Aggregation operators They are some of the particular operators that are actually used for combining the information. They can be understood as particular information fusion techniques. In fact, such operators are the ones that can be formalized straightforwardly in a mathematical way. The most well known aggregation operators are the means. Other operators as t -norms and t -conorms, already pointed out above, are also considered by some as aggregation operators. In general, we consider operators that combine N values in a given domain D obtaining a new single datum in the

same domain D . For example, in the case of the arithmetic mean, we combine N values on the real line and obtain a new datum that is also a value in the real line.

Taking all this into account, we would say that a robot that combines sensory information includes some information integration processes in its internal system. When sensory data is combined with a mean, or a similar function, we would say that it uses a fusion method that is an aggregation operator. In contrast, if the robot combines the data using a neural network or another black-box approach, we would say that it uses an information fusion method but that no aggregation operator can be distinguished.

As said above, some people distinguish between aggregation operators and means. In this case, a mean is a particular aggregation operator that satisfies unanimity and monotonicity. Other operators that combine N values but that do neither satisfy unanimity nor monotonicity are not means but aggregation operators. Besides, there are aggregation operators that are not required to satisfy monotonicity because they are not defined on ordered domains. This is the case of operators on nominal scales. The plurality rule or voting is one of such operators.

In this work we consider aggregation in a narrow sense. Therefore, an aggregation operator is a function \mathbb{C} , from consensus, such that combines N values in a domain D and returns another value in the same domain, i. e., $\mathbb{C} : D^N \rightarrow D$. Such function satisfies unanimity (i. e., $\mathbb{C}(a, \dots, a) = a$ for all a in D) and, if applicable, monotonicity (i. e., $\mathbb{C}(a_1, \dots, a_N) \leq \mathbb{C}(b_1, \dots, b_N)$ when $a_i < b_i$, i. e., this latter condition is valid, as stated before, only when there is an order on the elements in D , and, otherwise, the condition is removed).

Operators on ordered domains defined in this way satisfy internality. That is, the aggregation of values a_1, \dots, a_N is a value between the minimum and the maximum of these values:

$$\min_i a_i \leq \mathbb{C}(a_1, \dots, a_N) \leq \max_i a_i .$$

Aggregation Operators and Their Construction

The definition of an aggregation operator has to take into account several aspects. On the one hand, it should consider the type of data to be combined (this aspect has been mentioned before). That is, whether data is numerical, categorical, etc. On the second hand, it should consider the level of abstraction. That is, whether aggregation has to be done at a lower level or at a higher one. Due to the data flow, the data in a system typically moves from low levels to higher ones. Then, the fusion of two data can be done

at different abstraction levels. Note that data at different levels have different representations, e. g. information at a lower level might be numerical (e. g. the values of a measurement), while at a higher level it might be symbolical (e. g. propositions in logic, or belief measures). The level of abstraction and the type of data that are appropriate for fusion depends on a third element, the type of information. That is, information from different sources might be redundant or complementary. Redundant information is usually aggregated at a low level (e. g. the values of measuring the same object using two similar sensors), while complementary information is usually aggregated at a higher level (e. g., the fusion of images to have a better representation of 3D objects cannot be done at a pixel-signal-level).

Once the appropriate environment for a function has been settled, it is the time for the definition of the function. Two main cases can be distinguished, (i) definition from examples and (ii) definitions from properties.

- (i) The use of examples for defining the operators follows machine learning theory. Two main cases can be distinguished. The first one is when we assume that the model of aggregation is known. e. g., we know that we will use a weighted mean for the aggregation, and then the problem is to find the appropriate parameters for such weighted mean. The second one is when the model is not known and only a loose model – black box model – is assumed. Then the examples are used to settle such model. The use of neural networks is an example of this kind of application.
- (ii) In the case of definition from properties, the system designer establishes the properties that the aggregation operator needs to satisfy. Then the operator is derived from such properties.

Functional equations [1,2] can be used for this purpose. Functional equations are equations that have functions as their unknowns. For example, $\phi(x + y) = \phi(x) + \phi(y)$ is an example of a functional equation (one of Cauchy's equations) with an unknown function ϕ . In this equation, the goal is to find the functions ϕ that satisfy such equation. In this case, when ϕ are continuous, solutions are functions $\phi(x)$ of the form $\phi(x) = \alpha x$.

In the case of aggregation, we might be interested in, for instance, functions $\phi(x, y)$ that aggregate two numerical values x and y , and that satisfy unanimity (i. e., $\phi(x, x) = x$), invariance to transformations of the form $\gamma(x) = x + t$ (i.e., $\phi(x_1 + t, x_2 + t) = \phi(x_1, x_2) + t$), and invariance to transformations of the form $\gamma(x) = rx$ – positive homogeneity (i. e., $\phi(rx_1, rx_2) = r\phi(x_1, x_2)$). Such functions are completely characterized and all of them are of the form $\phi(x_1, x_2) = (1 - k)x_1 + kx_2$.

Naturally, the definition from properties might cause the definition of an overconstrained problem. That is, we may include too restrictive constraints and define a problem with no possible solution. This is the case of the problem of aggregation of preferences [5] that lead to Arrow's impossibility theorem [4]. That is, there is a set of natural properties that no function for the aggregation of preferences can satisfy.

The definition of the aggregated value as the one that is located at a minimum distance from the objects being aggregated is another approach for defining aggregation operators. In this case, we need to define a distance between pairs of objects on the domain of application. Let D be the domain of application, let d be a distance function for pairs of objects in D (that is, $d: D \times D \rightarrow \mathbb{R}$). Then, the aggregation function $\mathbb{C}: D^N \rightarrow D$ applied to a_1, \dots, a_N is formally defined as

$$\mathbb{C}(a_1, \dots, a_N) = \arg \min_c \left\{ \sum_{i=1}^N d(c, a_i) \right\}.$$

Such definition is often known as the median rule. In the numerical setting, i. e., if the domain D is a subset of the real set, when $d(a, b) = (a - b)^2$, the aggregation operator results to be the arithmetic mean. When $d(a, b) = |a - b|$, the aggregation operator results into the median. Formally, the median is the only solution when N is of the form $2k + 1$ or when N is of the form $2k$ but the two central values are equal. Otherwise, (when N is of the form $2k$ and the two central values are different), the median is one of the solutions of the equation above but other solutions might be also valid. For example, let us assume that $N = 4$ and that $a_1 \leq a_2 < a_3 \leq a_4$, then another solution is,

$$\mathbb{C}(a_1, \dots, a_N) = \frac{(a_4 a_3 - a_2 a_1)}{(a_4 + a_3) - (a_2 + a_1)}.$$

When $d(a, b) = |a - b|^\infty$ for $k \rightarrow \infty$, we have that the midrange of the elements a_1, \dots, a_N is a solution of the equation above. That is,

$$\mathbb{C}(a_1, \dots, a_N) = \frac{(\min_i a_i - \max_i a_i)}{2}.$$

Some Aggregation Operators

There exist a large number of aggregation operators. We will discuss in this section a few of them, classified into different families. We begin with the quasi-arithmetic mean, a family of operators that encompasses some of the most well-known aggregation operators, the arithmetic mean, the geometric mean, and the harmonic mean.

Quasi-Arithmetic Means

A quasi-arithmetic mean is an aggregation operator \mathbb{C} of the form

$$\text{QAM}(a_1, \dots, a_N) = \varphi^{-1} \left(\frac{1}{N} \sum_{i=1}^N \varphi(a_i) \right).$$

with φ an invertible function.

We list below a few examples of quasi-arithmetic means.

- Arithmetic mean:

$$\text{AM}(a_1, \dots, a_N) = \frac{1}{N} \sum_{i=1}^N a_i.$$

- Geometric mean:

$$\text{GM}(a_1, \dots, a_N) = \sqrt[N]{\frac{1}{N} \sum_{i=1}^N a_i}.$$

- Harmonic mean:

$$\text{HM}(a_1, \dots, a_N) = \frac{N}{\sum_{i=1}^N \frac{1}{a_i}}.$$

- Power mean (Hölder mean, or root-mean-power, RMP):

$$\text{RMP}(a_1, \dots, a_N) = \sqrt[\alpha]{\frac{1}{N} \sum_{i=1}^N a_i^\alpha}.$$

- Quadratic mean (root-mean-square):

$$\text{RMS}(a_1, \dots, a_N) = \sqrt{\frac{1}{N} \sum_{i=1}^N a_i^2}.$$

These means are obtained from the quasi-arithmetic mean using the following definition for the function φ : $\varphi(x) = x$ (in the arithmetic mean), $\varphi(x) = \log x$ (in the geometric mean), $\varphi(x) = 1/x$ (in the harmonic mean), $\varphi(x) = x^\alpha$ (in the power mean), and $\varphi(x) = x^2$ (in the quadratic mean).

Several properties have been proved for these functions. In particular, some inequalities have been proven [10,15]. For example, it is known that the harmonic mean (HM) always return a value that is smaller than the one of the geometric mean (GM) and that the value of the arithmetic mean is smaller than the one of the arithmetic mean (AM). That is, the following inequality holds for all x, y ,

$$\text{HM}(x, y) \leq \text{GM}(x, y) \leq \text{AM}(x, y).$$

In the case of the power means, the following inequality holds,

$$\text{RMP}_r(x, y) \leq \text{RMP}_s(x, y) \quad \text{for all } r < s.$$

That is, the power means is monotonic with respect to its parameter.

Also related with the power means, we have that when $\alpha \rightarrow 0$, $\text{RMP}_\alpha = \text{GM}$, that when $\alpha \rightarrow \infty$, $\text{RMP}_\alpha = \max$ and that when $\alpha \rightarrow -\infty$, $\text{RMP}_\alpha = \min$.

From the point of view of the applications, it is of relevance whether the outcome of an aggregation operator is, in general, large or small. That is, whether the aggregation operator A is larger or smaller than the aggregation operator B . For example, whether it holds or not that

$$A(x, y) \leq B(x, y) \quad \text{for all } x, y.$$

Due to the importance in real practice of such ordering, some indices have been developed for analyzing the outcomes of the operators. The average value, the orness and the andness are examples of such indices.

The orness computes the similarity of the operator with the maximum. Recall that in fuzzy sets, the maximum is used to model the disjunction, i. e., or. In contrast to that, the andness computes the similarity of the operator with the minimum. Recall that in fuzzy sets, the minimum is used to model the conjunction, i. e., and. For these measures, it can be proven that (i) the operator maximum has a maximum orness (an orness equal to one) and a minimum andness (an andness equal to zero), (ii) the orness of the arithmetic mean is larger than the orness of the geometric mean, and this latter orness is larger than the one of the harmonic mean. This property follows from the following inequality that holds for all x and y ,

$$\text{HM}(x, y) \leq \text{GM}(x, y) \leq \text{AM}(x, y).$$

The interpretations for the andness and the orness ease the selection of the appropriate aggregation function. This is so because the orness can be interpreted as a degree of compensation. That is, when different criteria are considered, a high value of orness means that a few good criteria can compensate a large number of bad ones. In contrast to that, a low value of orness means that all criteria have to be satisfied in order to have a good final score. Taking this into account, and using the example above, when only low compensation is permitted, it is advisable to use the geometric mean or the harmonic mean. In contrast, a larger compensation requires the user to consider the arithmetic mean.

A similar approach can be applied for the power mean. The larger the parameter r in the function (i. e., RMP_r) the larger the compensation.

Weighted Means and Quasi-weighted Means

The weighted mean is the most well-known aggregation operator with weights, and the quasi-weighted mean is one of its generalizations. In such operator, weights are represented by means of a weighting vector. That is, a vector $w = (w_1, \dots, w_N)$ that satisfies

$$w_i \geq 0 \quad \text{and} \quad \sum w_i = 1.$$

- Weighted mean: given a weighting vector $w = (w_1, \dots, w_N)$, the weighted mean is defined as follows

$$\text{WM}(a_1, \dots, a_N) = \sum_{i=1}^N w_i a_i.$$

Naturally, when $w_i = 1/N$ for all $i = 1, \dots, N$, the weighted mean corresponds to the arithmetic mean.

In this definition, the weights can be understood as their reliability. The larger the importance, the larger the weights.

- Quasi-weighted mean: given a weighting vector $w = (w_1, \dots, w_N)$, the quasi-weighted mean (or quasi-linear mean) is defined as follows

$$\text{QWM}(a_1, \dots, a_N) = \varphi^{-1} \left(w_i \sum_{i=1}^N \varphi(a_i) \right),$$

with φ an invertible function, and $w = (w_1, \dots, w_N)$ a weighting vector.

The quasi-weighted mean generalizes the quasi-arithmetic mean given above introducing weights w_i for each of the source. Using appropriate functions φ , we obtain the weighted mean (with $\varphi(x) = x$), the weighted geometric mean (with $\varphi(x) = \log x$), the weighted harmonic mean (with $\varphi(x) = 1/x$), the weighted power mean (with $\varphi(x) = x^\alpha$), and the weighted quadratic mean (with $\varphi(x) = x^2$).

Losoncz's Means

This family of means, defined in [6], is related to the quasi-weighted means. While in the former the weights only depend on the information source (there is one weight for each a_i but such weight does not depend on the value a_i), this is not the case in the Losoncz's means. In this operator, weights depend on the values being aggregated.

Formally, a Losoncz's mean is an aggregation operator \mathbb{C} of the form,

$$\mathbb{C}(a_1, \dots, a_N) = \varphi^{-1} \left(\frac{\sum_{i=1}^N \pi_i(a_i) \varphi(a_i)}{\sum_{i=1}^N \pi_i(a_i)} \right),$$

with φ an invertible function.

Naturally, when $\pi_i(a)$ is constant for all a , we have that the function reduces to a quasi-weighted means. When $\pi_i(a) = a^q$ and $\varphi(a) = a^{p-q}$ with $q = p - 1$, the Losoncz's means reduces to the counter-harmonic mean, also known by Lehmer mean. That is,

$$\mathbb{C}(a_1, \dots, a_N) = \frac{\sum_{i=1}^N a_i^p}{\sum_{i=1}^N a_i^{p-1}}.$$

When $\pi_i(a) = w_i a^q$ and $\varphi(a) = a^{p-q}$ with $q = p - 1$, the Losoncz's means reduces to the weighted Lehmer mean. That is,

$$\mathbb{C}(a_1, \dots, a_N) = \frac{\sum_{i=1}^N w_i a_i^p}{\sum_{i=1}^N w_i a_i^{p-1}}.$$

Linear Combination of Order Statistics and OWA Operators

Given a_1, \dots, a_N the i th order statistics [3] is the element in a_1, \dots, a_N that occupies the i th position when such elements are ordered in increasing order. Formally, let i be an index $i \in \{1, \dots, N\}$. Then, the i th order statistics OS_i of dimension N is defined as follows:

$$\text{OS}_i(a_1, \dots, a_N) = a_{s(i)}.$$

Where $\{s(1), \dots, s(N)\}$ is a permutation of $\{1, \dots, N\}$ such that $a_{s(j)} \leq a_{s(j+1)}$ for all $j \in \{1, \dots, N-1\}$.

When N is of the form $2k + 1$, $\text{OS}_{(N+1)/2} = \text{OS}_{k+1}$ is the median; when N is of the form $2k$, the median corresponds to the mean value of $\text{OS}_{N/2}$ and $\text{OS}_{N/2+1}$. That is,

$$\text{median}(a_1, \dots, a_N) = (\text{OS}_{N/2} + \text{OS}_{N/2+1})/2.$$

The OWA (ordered weighted averaging) operator is a linear combination of order statistics (an L -estimator). Its definition is as follows.

- OWA operator: given a weighting vector $w = (w_1, \dots, w_N)$ (i. e., a vector such that $w_i \geq 0$ and $\sum w_i = 1$), the OWA operator is defined as follows

$$\text{OWA}(a_1, \dots, a_N) = \sum_{i=1}^N w_i a_{\sigma(i)},$$

where $\{\sigma(1), \dots, \sigma(N)\}$ is a permutation of $\{1, \dots, N\}$ such that $a_{\sigma(i-1)} \geq a_{\sigma(i)}$ for all $i \in \{2, \dots, N\}$. Equivalently,

$$\text{OWA}(a_1, \dots, a_N) = \sum_{i=1}^N w_i \cdot \text{OS}_i(a_1, \dots, a_N).$$

Naturally, when $w_i = 1/N$ for all $i = 1, \dots, N$, the OWA operator corresponds to the arithmetic mean.

The ordering of the elements, permit the user to express the importance of some of the values being aggregated (importance of the values with respect to their position). For example, the OWA permits us to model situations where importance is given to low values (e. g. in a robot low values are more important than larger ones to avoid collisions). So, informally, we have that the w_i for $i = N$ and $i = N - 1$ should be large.

- Geometric OWA operator: given a weighting vector $w = (w_1, \dots, w_N)$, the geometric OWA (GOWA) operator is defined as follows

$$\text{GOWA}(a_1, \dots, a_N) = \prod_{i=1}^N a_{\sigma(i)}^{w_i},$$

where $\{\sigma(1), \dots, \sigma(N)\}$ is a permutation of $\{1, \dots, N\}$ such that $a_{\sigma(i-1)} \geq a_{\sigma(i)}$ for all $i \in \{2, \dots, N\}$. That is, the GOWA operator is a geometric mean of the order statistics.

- Generalized OWA operator: given a weighting vector $w = (w_1, \dots, w_N)$, the generalized OWA operator is defined as follows

$$\text{Generalized OWA}(a_1, \dots, a_N) = \left(\sum_{i=1}^N w_i a_{\sigma(i)}^\alpha \right)^{1/\alpha},$$

where $\{\sigma(1), \dots, \sigma(N)\}$ is a permutation of $\{1, \dots, N\}$ such that $a_{\sigma(i-1)} \geq a_{\sigma(i)}$ for all $i \in \{2, \dots, N\}$. That is, the generalized OWA operator is a generalized mean (root-mean-powers) of the order statistics.

- Quasi-OWA operator: given a weighting vector $w = (w_1, \dots, w_N)$, the quasi-OWA (QOWA) operator is defined as follows

$$\text{QOWA}(a_1, \dots, a_N) = \varphi^{-1} \left(\sum_{i=1}^N w_i \varphi(a_{\sigma(i)}) \right).$$

where $\{\sigma(1), \dots, \sigma(N)\}$ is a permutation of $\{1, \dots, N\}$ such that $a_{\sigma(i-1)} \geq a_{\sigma(i)}$ for all $i \in \{2, \dots, N\}$. That is, the quasi-OWA operator is a quasi-weighted mean of the order statistics.

$$\text{OWA}(a_1, \dots, a_N) = \prod_{i=1}^N a_{\sigma(i)}^{w_i},$$

where $\{\sigma(1), \dots, \sigma(N)\}$ is a permutation of $\{1, \dots, N\}$ such that $a_{\sigma(i-1)} \geq a_{\sigma(i)}$ for all $i \in \{2, \dots, N\}$.

- BADD-OWA operator: given a weighting vector $w = (w_1, \dots, w_N)$, the BADD-OWA (basic defuzzification

distribution OWA) corresponds to the counter-harmonic mean (see above). That is,

$$\mathbb{C}(a_1, \dots, a_N) = \frac{\sum_{i=1}^N a_i^p}{\sum_{i=1}^N a_i^{p-1}}.$$

- Induced OWA operator: given a weighting vector $w = (w_1, \dots, w_N)$ and a priority vector $b = (b_1, \dots, b_N)$, the induced OWA (IOWA) operator is defined as follows

$$\text{IOWA}_b(a_1, \dots, a_N) = \sum_{i=1}^N w_i a_{\sigma(i)},$$

where $\{\sigma(1), \dots, \sigma(N)\}$ is a permutation of $\{1, \dots, N\}$ such that $b_{\sigma(i-1)} \geq b_{\sigma(i)}$ for all $i \in \{2, \dots, N\}$. Naturally, the OWA corresponds to the IOWA when a is used as the priority vector.

The WOWA Operator

Let p and w be two weighting vectors of dimension N . Then, the WOWA [23] operator (weighted ordered weighted averaging operator) of dimension N is defined as

$$\text{WOWA}_{p,w}(a_1, \dots, a_N) = \sum_{i=1}^N \omega_i a_{\sigma(i)},$$

where σ is defined as above (i.e. $a_{\sigma(i)}$ is the i th largest element in the collection a_1, \dots, a_N), and the weight ω_i is defined as

$$\omega_i = w^* \left(\sum_{j \leq i} p_{\sigma(j)} \right) - w^* \left(\sum_{j < i} p_{\sigma(j)} \right),$$

with w^* being a nondecreasing function that interpolates the points

$$\left\{ \left(i/N, \sum_{j \leq i} w_j \right) \right\}_{i=1, \dots, N} \cup \{(0, 0)\}.$$

The function w^* is required to be a straight line when the points can be interpolated in this way.

The WOWA operator generalizes the OWA and the weighted mean. When $p = (1/N, \dots, 1/N)$, the WOWA reduces to an OWA operator and when $w = (1/N, \dots, 1/N)$, the WOWA reduces to a weighted mean. That is,

$$\begin{aligned} \text{WOWA}_{p,w}(a_1, \dots, a_N) &= \text{OWA}_w(a_1, \dots, a_N), \\ &\text{when } p = (1/N, \dots, 1/N), \end{aligned}$$

and

$$\text{WOWA}_{p,w}(a_1, \dots, a_N) = \text{WM}_p(a_1, \dots, a_N),$$

when $w = (1/N, \dots, 1/N)$.

As the OWA and the weighted mean generalize the arithmetic mean, when the weights are all equal to $1/N$, the same happens with the WOWA. i. e.,

$$\text{WOWA}_{p,w}(a_1, \dots, a_N) = \text{AM}(a_1, \dots, a_N),$$

when $p = (1/N, \dots, 1/N)$ and $w = (1/N, \dots, 1/N)$.

As stated here, the WOWA operator reduces to the OWA and the weighted mean for appropriate weighting vectors. This property permits the user to interpret the two weighting vectors p and w used in the WOWA in terms of the corresponding interpretation in the OWA and the weighted mean. That is, p corresponds to the reliability of the information sources and w permits to express the importance of the values (with respect to their ordering). Thus, the WOWA permits the user to express which are the most reliable information sources (e. g., sensors) and which are the values that have to be considered more relevant (e. g. larger importance to small values in sensors to avoid collisions, or larger importance to large values if a high degree of compensation is permitted in multicriteria decision making).

Fuzzy Integrals

Fuzzy integrals can be used in aggregation [14,25]. Formally, they integrate a function with respect to a fuzzy measure. From the aggregation point of view, the function corresponds to the data to be aggregated and the fuzzy measure corresponds to some background knowledge (following the jargon in artificial intelligence) about the reliability or importance of the information sources. In this sense, the fuzzy measure can be seen as a generalization of the weights as they are used for the same purpose. The difference is that while the weights can only be used for a single information source, the measures are applied to sets of them.

In order to use the fuzzy integrals in aggregation we need first some formalization. First, we need to define the set of information sources, let it be $X = \{x_1, \dots, x_N\}$. That is, X is the set of sensors in a robot or the set of criteria in a multicriteria decision making problem. Then, we need to express the datum (a number) supplied by source x_i . This is represented by a function f from X to \mathbb{R} . That is, $f(x_i)$ is the data supplied by sensor x_i or the evaluation of the criterion x_i . In terms of the notation used above in this

work, $f(x_i) = a_i$. Then, the aggregation of the data supplied by the information sources X can be computed as the integral of the function f with respect to a fuzzy measure on X . The measure is a function defined on the subsets of X . So, for any set of information sources, we have its *weight*.

Formally, a fuzzy measure is a set function on X into $[0, 1]$. That is, a function from $\wp(X)$ into $[0, 1]$ or, in other words, a function that assigns a value in $[0, 1]$ to each subset A of X . Furthermore, fuzzy measures are monotonic with respect to set inclusion. Their definition is as follows.

A fuzzy measure μ is a set function $\mu: \wp(X) \rightarrow [0, 1]$ that satisfies:

- $\mu(\emptyset) = 0$,
- $\mu(X) = 1$,
- $A \subseteq B$ implies $\mu(A) \leq \mu(B)$.

When the measure of A is understood as its importance, the last condition in this definition states that the more sources we have, the larger their importance. Boundary conditions (the first two conditions) imply that the minimum importance (the one of the empty set) is zero and that the maximum importance (the one of the full set) is one. The boundary condition $\mu(X) = 1$ is arbitrary; nevertheless, from the point of view of the aggregation, it can be justified as being similar to the one existing for the weights in the weighted mean, and having a similar rationale. That is, the weights w_i in a weighting vector add to one: $\sum_i w_i = 1$. Note that such restriction is appropriate in the case of the weighted mean because it implies that the weighted mean satisfies unanimity ($\text{WM}(a, \dots, a) = a$). Note that any other arbitrary bound K (i. e., $\sum_i w_i = K$) would just result into a weighted mean such that unanimity does not hold but that $\text{WM}(a, \dots, a) = Ka$.

Due to the usefulness of the fuzzy measures [14], several families have been defined and studied in the literature. Some of them are the following: Sugeno lambda measures, decomposable fuzzy measures, hierarchically decomposable fuzzy measures, k -order additive fuzzy measures.

As said, fuzzy measures are used in combination with fuzzy integrals. Several fuzzy integrals have been defined in the literature. We review a few of them below.

- Choquet integral [11]: The Choquet integral of a function $f: X \rightarrow \mathbb{R}^+$ with respect to a fuzzy measure μ is defined by:

$$\begin{aligned} \text{CI}_\mu(f) &= (C) \int f \, d\mu \\ &= \sum_{i=1}^N [f(x_{s(i)}) - f(x_{s(i-1)})] \mu(A_{s(i)}). \end{aligned}$$

where $f(x_{s(i)})$ indicates that the indices have been permuted so that $0 \leq f(x_{s(1)}) \leq \dots \leq f(x_{s(N)}) \leq 1$, where $f(x_{s(0)}) = 0$ and where $A_{s(i)} = \{x_{s(i)}, \dots, x_{s(N)}\}$.

The Choquet integral generalizes the arithmetic mean, the weighted mean, the OWA operator and the WOWA operator.

- (a) It generalizes the arithmetic mean when we have that the measure is proportional to the cardinality of the set, i.e., when $\mu(A) = |A|/|X|$, the Choquet integral of f corresponds to the arithmetic mean of $a_i = f(x_i)$.
- (b) Given a weighting vector $w = (w_1, \dots, w_N)$, if we define a fuzzy measure as follows

$$\mu(A) = \sum_{x_i \in A} w_i,$$

we have that the Choquet integral of f with respect to μ corresponds to the weighted mean of a_i with respect to w .

- (c) Given a weighting vector $w = (w_1, \dots, w_N)$, if we define a fuzzy measure μ as follows

$$\mu(A) = \sum_{i=1}^{|A|} w_i,$$

we have that the Choquet integral of f with respect to μ corresponds to the OWA of a_i with respect to w .

- (d) Given weighting vectors w and p , we can define a fuzzy measure in terms of the interpolating function w^* . Once such function is built, we define the fuzzy measure μ as follows:

$$\mu(A) = w^* \left(\sum_{i=1}^{|A|} w_i \right),$$

we have that the Choquet integral of f with respect to such μ corresponds to the WOWA of the a_i with respect to weighting vectors w and p .

- Sugeno integral [22]: The Sugeno integral of a function $f: X \rightarrow \mathbb{R}^+$ with respect to a fuzzy measure μ is defined by:

$$SI_\mu(f) = (S) \int f d\mu = \max_{i=1}^N \left(\min \left(f(x_{s(i)}), \mu(A_{s(i)}) \right) \right).$$

Here, s and $A_{s(i)}$ are defined as above.

Other fuzzy integrals exist. Among them, we have the twofold integral that generalizes both Choquet and Sugeno integrals. The generalization uses two fuzzy measures μ_C and μ_S that correspond, respectively, to the measures used

in the Choquet and Sugeno integrals. The measure of the Choquet integral has a probabilistic flavor as the Choquet integral generalizes the weighted mean, that can be seen as the expectation of the a_i with respect to the probability distribution $w = (w_1, \dots, w_N)$. Note that the weighting vector in the weighted mean is compatible with a probability distribution because weights are positive and add to one. Instead, the Sugeno integral has a fuzzy flavor as it generalizes the weighted min. and the weighted max., which are used in fuzzy systems. Such operators are defined in terms of weighting vectors that do not add to one but have a maximum of one. So, they are like possibility distributions.

In short, the twofold integral integrates a function with respect to two fuzzy measures μ_C and μ_S . We define such integral below.

- Twofold integral [17,23]: The twofold integral of a function $f: X \rightarrow \mathbb{R}^+$ with respect to a fuzzy measure μ is defined by:

$$TI_{\mu_S, \mu_C}(f) = \sum_{i=1}^N \left(\left(\max_{j=1}^i \left(\min \left(f(x_{s(i)}), \mu_S(A_{s(i)}) \right) \right) \right) \cdot \left(\mu_C(A_{s(i)}) - \mu_C(A_{s(i+1)}) \right) \right).$$

Here, s and $A_{s(i)}$ are defined as above. Naturally, $A_{s(n+1)} = \emptyset$.

When $\mu_C = \mu^*$, the twofold integral reduces to the Sugeno integral and when $\mu_S = \mu^*$, the twofold integral reduces to the Choquet integral. Here, μ^* represents ignorance and is defined by $\mu(\emptyset) = 0$ and $\mu^*(A) = 1$ when $A \neq \emptyset$. In addition, when $\mu_C = \mu_S = \mu^*$, we have that the twofold integral reduces to the maximum.

Future Directions

Aggregation operators have been studied for a long time. For example, the inequality

$$HM(x, y) \leq GM(x, y) \leq AM(x, y),$$

that involves the harmonic mean (HM), the geometric mean (GM), and the arithmetic mean (AM) was already known by Pappus of Alexandria (fl. c. 300–c. 350). It is given in his book *Synagoge* (book III). Since then, a large number of aggregation operators have been proposed fostered by the new applications and the theoretical studies.

From the practical point of view, current research is oriented towards the embedding of such operators into real applications. This has motivated the study and the de-

velopment of methods and techniques for parameter determination. e.g., the development of methods that permits a user to find the appropriate parameters in a given application. For example, methods have been defined to determine the weights in the weighted mean, and to determine the fuzzy measure in a fuzzy integral. This is an active line of research.

Another related topic is the selection of the appropriate function. Now, the methods for parameter determination mainly presume that the appropriate operator is known beforehand and that the only elements to be determined are the parameters. Tools and methods are needed to help the users to select which is the appropriate function.

Finally, there is the need for developing architectures for information integration. That is, architectures that encompass all the processes related with fusion: from the input data (acquisition of the data) to the final decision or output.

All these aspects, oriented towards the real application of aggregation operators are combined with new research on the theoretical side. Such research includes the definition of new functions (needed so that systems can work on new types of data), the characterization of the functions (needed so that we can know which are the properties of the functions and so that a user can then select the appropriate one based on his/her problem requirements) and the study of composite models using aggregation operators (i. e., hierarchical models that combine several aggregation operators).

Bibliography

Primary Literature

1. Aczél J (1966) Lectures on functional equations and their applications. Academic, New York, London
2. Aczél J (1987) A short course on functional equations. Reidel, Dordrecht
3. Arnold BC, Balakrishnan N, Nagaraja HN (1992) A first course in order statistics. Wiley, New York
4. Arrow KJ (1951) Social choice and individual values, 2nd edn. Wiley, New York
5. Arrow KJ, Sen AK, Suzumura K (eds) (2002) Handbook of social choice and welfare. Elsevier, Amsterdam
6. Bajraktarević M (1958) Sur une équation fonctionnelle aux valeurs moyennes. Glasnik Mat Fiz I Astr 13(4):243–248
7. Barthélemy JP, McMorris FR (1986) The median procedure for n-trees. J Classif 3:329–334
8. Bouyssou D, Marchant T, Pirlot M, Perny P, Tsoukiàs A, Vincke P (2000) Evaluation and decision models: A critical perspective. Kluwer's International Series. Kluwer, Dordrecht
9. Bullen PS (2003) Handbook of means and their inequalities. Kluwer, Dordrecht
10. Bullen PS, Mitrinović DS, Vasić PM (1988) Means and their inequalities. Reidel, Dordrecht
11. Choquet G (1953) Theory of capacities. Ann Inst Fourier 5:131–295
12. Fishburn PC, Rubinstein A (1986) Aggregation of equivalence relations. J Classif 3:61–65
13. Fodor J, Roubens M (1994) Fuzzy preference modelling and multicriteria decision support. Kluwer, Dordrecht
14. Grabisch M, Murofushi T, Sugeno M (2000) Fuzzy measures and integrals: Theory and applications. Physica, Heidelberg
15. Hardy GH, Littlewood JE, Pólya G (1934) Inequalities, 2nd edn. Cambridge University Press, Cambridge
16. Hirsch JE (2005) An index to quantify an individual's scientific research output. Proc Natl Acad Sci 102(45):16569–16572
17. Narukawa Y, Torra V (2004) Twofold integral and Multi-step Choquet integral. Kybernetika 40(1):39–50
18. Mitchell HB (2007) Multi-sensor data fusion. An introduction. Springer, Heidelberg
19. Pappus (1982) La collection mathématique. Librairie Scientifique et Technique Albert Blanchard, Paris
20. Roy B (1996) Multicriteria methodology for decision aiding. Kluwer, Dordrecht
21. Ruspini EH, Bonissone PP, Pedrycz W (1998) Handbook of fuzzy computation. IOP, London
22. Sugeno M (1974) Theory of fuzzy integrals and its applications. Ph D dissertation, Tokyo Institute of Technology, Japan
23. Torra V (1997) The weighted OWA operator. Int J Intell Syst 12:153–166
24. Torra V (2003) La integral doble o *twofold* integral: Una generalització de les integrals de Choquet i Sugeno. Butlletí de l'Associació Catalana d'Intel·ligència Artificial 29:13–19. Preliminary version in English: Twofold integral: A generalization of Choquet and Sugeno integral. IIIA Technical Report TR-2003-08
25. Torra V, Narukawa Y (2007) Modeling decisions: Information fusion and aggregation operators. Springer, Heidelberg
26. Torra V, Narukawa Y (2008) The h-index and the number of citations: Two fuzzy integrals. IEEE Trans Fuzzy Syst 16(3): 795–797

Books and Reviews

- [26] gives a general description of the field of aggregation operators, it defines the main operators and discusses a few practical topics about their applications (e. g. parameter determination). (Calvo et al, 2002) is an edited book that contains state-of-the-art chapter on different topics related with aggregation and fusion. A few properties on the aggregation operators (mainly related with inequalities) can be found in the books by Bullen [9] and Bullen, Mitrinović and Vasić [10], and the excellent book by Hardy, Littlewood and Pólya [34]. [14] is an edited volume on fuzzy measures and fuzzy integrals. [18] is an introduction to multisensor data fusion, a topic very much related with aggregation operators
- Alsina C, Frank MJ, Schweizer B (2006) Associative functions: Triangular norms and copulas. World Scientific, Singapore
- Calvo T, Mayor G, Mesiar R (2002) Aggregation operators. Physica, Heidelberg
- Pap E (2002) Handbook of measure theory, vols I, II. North-Holland, Amsterdam

Yager RR (1988) On ordered weighted averaging aggregation operators in multi-criteria decision making. *IEEE Trans Syst Man Cybern* 18:183–190

Air Traffic Control, Complex Dynamics of

BANAVAR SRIDHAR, KAPIL SHETH
NASA Ames Research Center, Moffet Field, USA

Article Outline

Glossary
Definition of the Subject
Introduction
Complex Network Analysis
Current US Air Traffic Network
Future Air Traffic Scenarios
Concluding Remarks
Bibliography

Glossary

Air traffic flow Air Traffic Flow represents the dynamic distribution of air traffic over a region of space. Air traffic is undergoing major changes both in developed and developing countries. The demand for air traffic depends on population growth and other economic factors. Air traffic in the United States is expected to grow to 2 or 3 times the current levels of traffic in the next few decades. An understanding of the characteristics of the current and future flows is essential to design a good traffic flow management strategy.

Traffic flow management Safety limits the number of aircraft arriving at an airport or in a region of the airspace. Airspace Capacity, the maximum number of aircraft in a region, depends on the technology to keep aircraft separated by a safe distance and weather conditions. Airspace capacity decreases in the presence of severe weather and aircraft may have to be rerouted or delayed on the ground to maintain safety. The imbalance between airspace capacity and traffic flow demand leads to delays. Traffic flow management tries to maintain efficiency of the flows while not exceeding capacity limits.

Complex networks A network connects components of a system. The connections and the number of components vary with the function of the network. It is extremely difficult to analyze and visualize the behavior of networks when the number of components in the

system becomes large. Recently, there has been a major advance in our understanding of the behavior of networks with large number of components. Several theories have been advanced about the evolution of large biological and engineering networks by authors in diverse disciplines like physics, mathematics, biology and computer science.

Scale-free networks Several large biological and engineering networks exhibit a scale-free property in the sense that the probabilistic distribution of their nodes as a function of connections decreases slower than an exponential. These networks are characterized by the fact that a small number of components have a disproportionate influence on the performance of the network. Scale-free networks are tolerant to random failure of components; but are vulnerable to selective attack on components.

Definition of the Subject

Civil Aviation is a vital sector of the US economy. In 2004, air transportation and related industries generated an output of \$1.37 trillion employing 12.3 million people in the US [5]. The National Airspace System (NAS) refers to all the hardware, software and people involved in managing air traffic in the United States. More than 50 000 commercial flights operated in the US airspace alone on a typical day at the present time. Demand for air transportation has seen a six-fold increase in the past 30 years and estimates call for a strong average growth rate of 4.7% during the next 20 years [1]. This increase in demand will put a further strain on the airports and airspace resulting in delays. Air Transport Association, a group representing airlines, estimated the cost of delays to airlines in 2005 at 5.9 billion dollars. To address the changes required in the air transportation system, the US has created a multi-agency Joint Planning and Development Office to lead the required transformations. Similar activities are being pursued in Europe and Asia [14].

Introduction

Air traffic in the United States continues to grow at a steady pace except for a dip immediately after the tragic events of September 11, 2001. There are different growth scenarios associated both with the magnitude and the composition of the future air traffic. The Terminal Area Forecast (TAF), prepared every year by the Federal Aviation Administration (FAA), projects the growth of traffic in the United States [4]. Both Boeing and Airbus publish market outlooks for air travel annually. Although predicting the future growth of traffic is difficult, there are two sig-

nificant trends: (1) heavily congested major airports continue to see an increase in traffic, and (2) the emergence of regional jets and other smaller aircraft with fewer passengers operating directly between non-major airports. The interaction between air traffic demand and the ability of the system to provide the necessary airport and airspace resources can be modeled as a network. The size of the resulting network varies depending on the choice of its nodes. It would be useful to understand the properties of this network to guide future design and development. Many questions, such as the growth of delay with increasing traffic demand and impact of the en route weather on future air traffic, require a systematic understanding of the properties of the air traffic network.

Recently, there has been a major advance in the understanding of the behavior of networks with large number of components. Several theories have been advanced about the evolution of large biological and engineering networks by authors in diverse disciplines like physics, mathematics, biology and computer science [21]. Several networks exhibit a scale-free property in the sense that the probabilistic distribution of their nodes as a function of connections decreases slower than an exponential. These networks are characterized by the fact that a small number of components have a disproportionate influence on the performance of the network. Scale-free networks are tolerant to random failure of components, but are vulnerable to selective attack on components [2,17].

This paper examines two network representations for the current air traffic system. A network defined with the 40 major airports as nodes and with standard flight routes as links has a characteristic scale: all nodes have 60 or more links and no node has more than 460 links. Another network defined with current aircraft routing structure exhibits an exponentially truncated scale-free behavior. Its degree ranges from 2 connections to 2900 connections, and 217 nodes have more than 250 connections. Furthermore, those high-degree nodes are homogeneously distributed in the airspace. A consequence of this scale-free behavior is that the random loss of a single node has little impact, but the loss of multiple high-degree nodes (such as occurs during major storms in busy airspace) can adversely impact the system. Two future scenarios of air traffic growth are used to predict the growth of air traffic in the United States. It is shown that a three-times growth in the overall traffic may result in a ten-times impact on the density of traffic in certain parts of the United States.

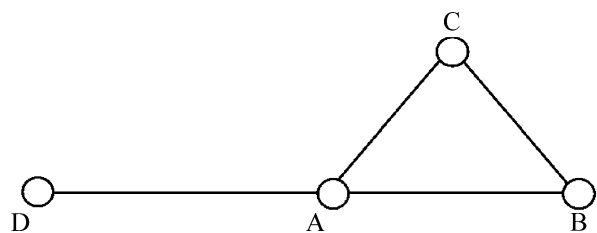
The paper is organized as follows. The Sect. “Complex Network Analysis” provides a brief overview and terminology of the complex networks useful in the analysis of air traffic. This is followed by an application of the com-

plex network methodology to the current air traffic system. The Sect. “Future Air Traffic Scenarios” considers different scenarios for the evolution of air traffic in the United States during the next 25 years and looks for changes in the behavior of the network. An analysis of the future air traffic scenarios shows that a three-times growth in overall traffic can result in a tenfold impact on the density of highly connected nodes in certain parts of the United States. Thus, in addition to bottlenecks at major airports, the risk of airspace congestion calls for route restructuring and the introduction of new procedures and automation to increase airspace capacity. Concluding remarks and possible research in this area are discussed in the last section.

Complex Network Analysis

Complex systems have many agents or components interacting with each other, and their collective behavior is not a simple combination of the individual behavior. The pattern of interaction between agents can be studied as a network of connections between agents. Networks of many types are pervasive in modern society, and scientists from many fields are trying to broaden their understanding of the behavior of the networks. A network is made up of basic components called either vertices or nodes. Each node is connected to other nodes in the system. The line connecting two nodes is referred to as a link or an edge. An edge is directed if it runs only in one direction and undirected if it runs in both directions. Degree is the number of edges connected to a node. Figure 1 shows a simple network with nodes A, B, C and D and undirected edges connecting them. The degrees of nodes A, B, C and D are 3, 2, 2 and 1, respectively.

The structure of the network strongly influences the functions performed by the network. It is possible to analyze networks of small sizes (less than 100) by drawing the picture of the network and analyzing its properties. The number of nodes in a complex engineering system, such as the worldwide web [10], can easily be as large as a few million nodes. The behavior of Random Graphs, graphs



Air Traffic Control, Complex Dynamics of, Figure 1
Nodes and edges in a network

where nodes are connected to other nodes randomly, as the number of nodes becomes large has been studied extensively by mathematicians [13]. The behavior of Regular Networks, idealized systems where each component is identical, has been studied by several authors [17]. Networks describing real systems are neither random nor regular. The ability to model real systems and capture the behavior of key variables is extremely difficult. Recent developments in complex networks examine the statistical properties of large networks and help answer questions about the dynamics and stability of such networks. However, studies on the effects of structure on system behavior are still in their early stages.

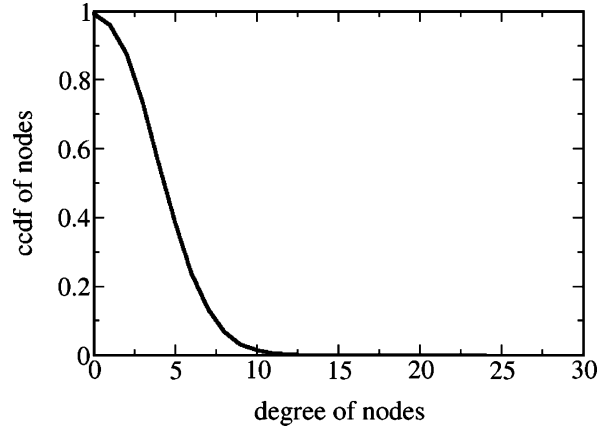
A major area of interest is the role played by the distribution of the degree of nodes in a network. Let p_k be the fraction of nodes in the network that has degree k . The degree distribution for the network can be computed by generating a histogram of p_k . The degree distribution for large random graphs, where each edge is present or absent with equal probability, can be modeled as a Poisson distribution. The distribution tends to peak for a small value of k and decays exponentially for large values of k . When using real data, to reduce the effect of noise in small datasets, the distribution of degree is expressed in terms of the complementary cumulative distribution function (ccdf), P_k , the probability of the number of nodes in the graph with degree greater than k . P_k is computed using the expression

$$P_k = \sum_{i=k}^{\infty} p_i.$$

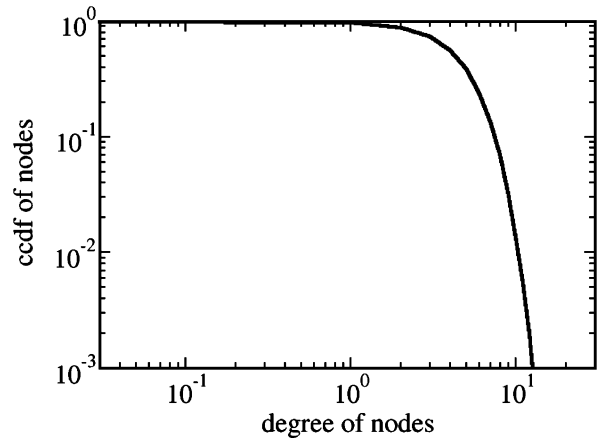
Figure 2 shows the ccdf of the Poisson distribution, $e^{-\lambda} \lambda^k / k!$, for $\lambda = 4$. Figure 3 shows the ccdf for the Poisson distribution on a logarithmic scale.

It has been observed that the degree distributions for some real-world networks, such as the Internet and biological networks, are highly skewed with tails several times longer than the mean. These real networks have a small number of nodes, or hubs, with a high level of connectivity. Hubs play an important role in influencing the properties of the network. Real networks exhibiting a small number of hubs are referred to as “scale-free” networks [6]. The distributions of degree for a number of networks, e.g., Internet, World Wide Web, collaboration network of mathematicians, etc., show a power law in their tails: $P_k \approx k^{-\alpha}$ for some constant value of α . Figure 4 shows the ccdf for $\alpha = 2$. Power law distributions (Fig. 4) appear linear in a logarithmic scale (Fig. 5).

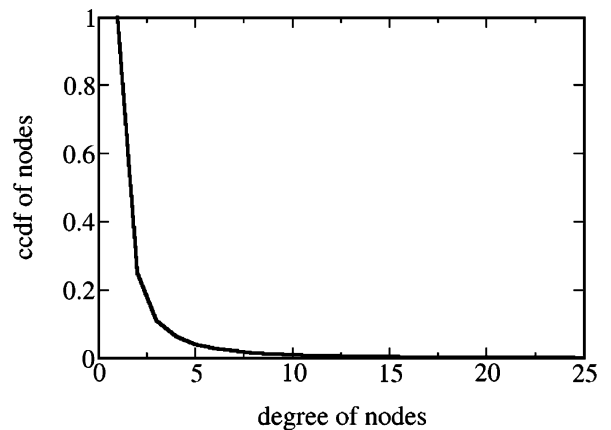
The appearance of hubs in scale-free networks is explained in terms of two behaviors both in people and networks, namely, real systems-growth and preferential at-



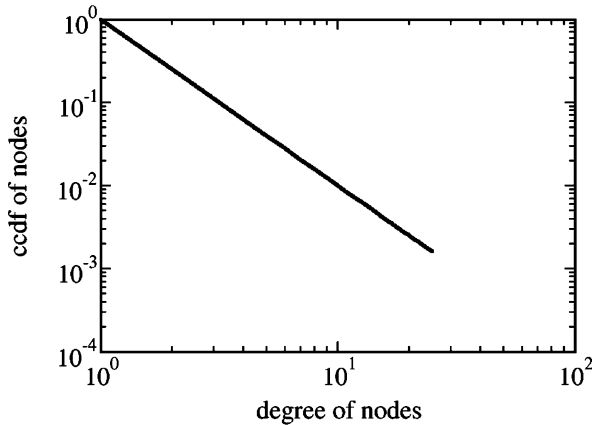
Air Traffic Control, Complex Dynamics of, Figure 2
Poisson distribution



Air Traffic Control, Complex Dynamics of, Figure 3
Poisson distribution in logarithmic scale



Air Traffic Control, Complex Dynamics of, Figure 4
Power law distribution



Air Traffic Control, Complex Dynamics of, Figure 5
Power law distribution in logarithmic scale

tachment. There is a tendency for new growth to gravitate towards desirable locations. Generally, all the desirable locations already have some existing nodes. The preferential attachment mechanism leads to the creation of more powerful nodes or hubs. The various hubs compete to attract new nodes and absorb them, resulting in an increase of the number of links in the initial hubs.

Scale-free networks with power law distributions exhibit two important properties: (a) remarkable resistance to random failure of nodes and (b) extreme vulnerability to targeted attacks. The functionality and performance of the network depends on the existence of the edges between pairs of nodes. The distance between the remaining vertices gets longer as nodes are removed from a network and the removal of a certain number of nodes may result in a collapse of the entire system. The tolerance of networks to failures or removal of nodes varies with their level of resilience. The nodes from a network could be removed randomly or in a coordinated approach. Several studies have shown that scale-free networks are generally robust to random removal of nodes. However, they are less tolerant to the selective removal of hubs.

The next section will examine the air traffic in the United States to see if it exhibits the properties of a scale-free network and draw analogies from the attack vulnerabilities of other complex networks. The methodology can also be used to understand the properties of the Next Generation Air Transportation System as it evolves over the next few decades [18].

Current US Air Traffic Network

The National Airspace System (NAS) refers to the collection of hardware, software and people, including runways,

radars, networks, FAA, airlines, etc., involved in Air Traffic Management (ATM) in the US. It has been pointed out that the NAS should be modeled as a complex adaptive system [12]. The NAS can be looked at as a network at several different levels [11]. Some examples of networks involving components of the NAS are the route network of an airline connecting different airports, a network of sectors (geographical region used to monitor safe separation between aircraft) interconnected by geographical proximity, network of airline crew located in different cities and a surface network of ramps, runways and departure gates at an airport. The network components and links vary with the planning interval and the modeling problem. The analysis of current traffic uses data from 2006 and constructs two different types of networks for a typical day of operations in the NAS. The analysis is based on actual traffic for a day in July 2006. The traffic data are processed using Future Air traffic management Concept Evaluation Tool (FACET) simulation software [19] for conducting the network flow analysis.

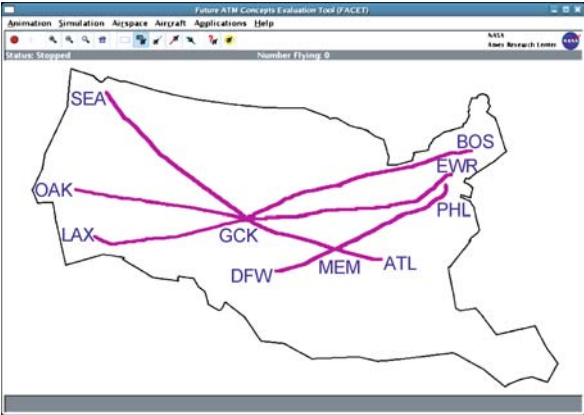
The major distinctions regarding the two air traffic networks are described in Table 1. The nodes in the Airport Network (AN) correspond to the 40 major airports in the US. Today, a set of predefined alternative routes are used for flying between particular city pairs. Each node in network AN is connected to another node through one or more routes. The Airport and Airspace Network (AAN) includes all airports in the US and routes connecting these airports. A route between two airports is defined by a series of geographical positions or fixes in the airspace. Each fix together with the airports represent a node in the AAN network.

Computation of the Degree of a Node and the Distribution Function

The computation of the degree of a node and the distribution function is illustrated by an example. The FACET simulation software generates the departure airport, arrival airport and the fixes through which the aircraft travels en route based on air traffic data. Figure 6 shows the intersection of different routes connecting airports through common fixes. In the figure, aircraft traveling from Seattle (SEA) to Atlanta (ATL), aircraft traveling from Oakland (OAK) to Newark (EWR), and aircraft traveling from Los Angeles (LAX) to Boston (BOS) pass through the fix Garden City, Kansas (GCK). The degree of GCK is equal to the total number of aircraft traveling along these three routes during the day. The degree of node GCK is three if a single aircraft travels between each of these three pairs

Air Traffic Control, Complex Dynamics of, Table 1
Different types of Air Traffic Management networks

Network	Nodes	Number of Nodes	Edges
Airport Network (AN)	40 major airports	40	Routes between 40 major airports
Airport and Airspace Network (AAN)	All airports and fixes along routes connecting the airports	8170	Routes between all airports



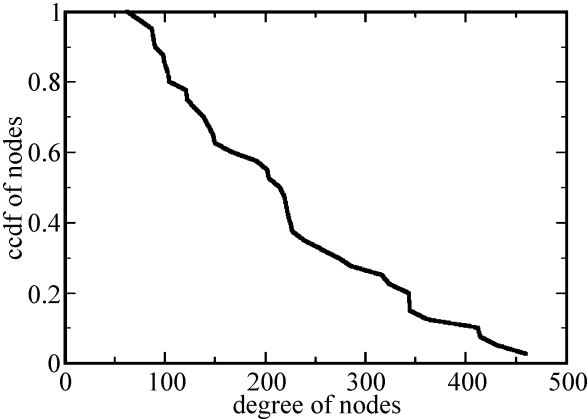
Air Traffic Control, Complex Dynamics of, Figure 6
Example of intersection of routes at common fixes

of airports. Similarly, aircraft traveling from Dallas (DFW) to Philadelphia (PHL) and Seattle to Atlanta pass through the fix Memphis, TN (MEM). The degree of fix MEM is two if a single aircraft travels between each of these city pairs. The distribution and the ccdf can be computed easily given the degree associated with all the airports and en route fixes and are described in the next section.

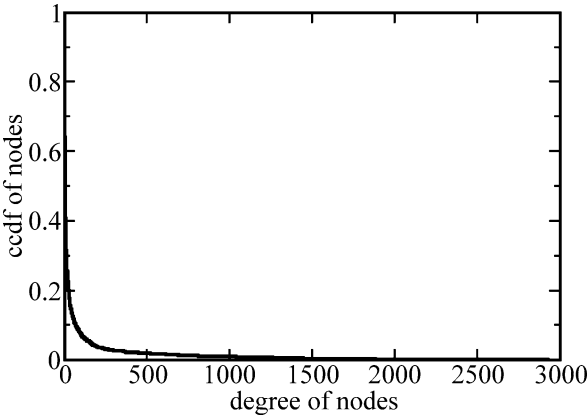
**Behavior of the Degree of Nodes
in Air Traffic Management Network**

The ccdf for the nodes in network AN is shown in Fig. 7. The figure shows the probability that an airport in the network has more than a certain number of routes originating or terminating at the airport. The number of connections at all airports exceeds 60 and Chicago O’Hare airport has the maximum number (460) of connections. The distribution of nodes in the airport network does not show scale-free behavior. The airport connection distribution in the U.S is similar to the distribution of the connections between the networks of world airports [3].

Figure 8 shows the ccdf of the nodes in network AAN. The ccdf of the nodes in network AAN shows a rapid decay of nodes with the degree of the node. However, it has 217 nodes with more than 250 links, indicative of high volumes



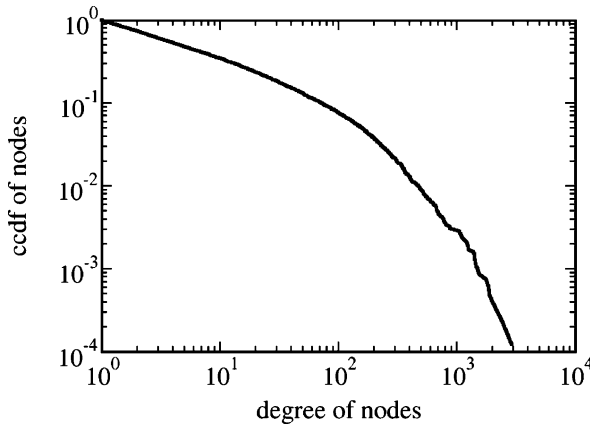
Air Traffic Control, Complex Dynamics of, Figure 7
Cumulative distribution function for nodes (40 major airports) in network AN



Air Traffic Control, Complex Dynamics of, Figure 8
Distribution of nodes in AAN with current traffic

of traffic through these nodes. Figure 9 shows the logarithmic behavior of the ccdf of the nodes in network AAN.

The nodes of network AAN initially follow the power law curve similar to several scale-free networks [17]. However, towards the end of the tail, the distribution shows a deviation from the power law behavior indicating a limit on the number of nodes that have a high number of connections. The limit on the growth of the large hub nodes, in the distribution of the ATM network, is due to the con-



Air Traffic Control, Complex Dynamics of, Figure 9
Logarithmic behavior of nodes in network AAN



Air Traffic Control, Complex Dynamics of, Figure 10
Geographical distribution of 250G nodes in network AAN with current traffic and severe weather polygons

straints imposed on traffic demand by the location of the cities, economic development and government policies.

The existence of hub and secondary airports characterizes current air traffic operations [8]. The network analysis provides an additional ability to study traffic behavior in the en route airspace. Figure 10 shows geographical distribution of the nodes with degree higher than 250 in network AAN. These nodes will be referred to as 250G nodes. 250G nodes represent 2.7% of the total nodes in the network. Table 2 shows the number of 250G nodes in different traffic control regions, referred to as Centers. The Centers, Chicago (ZAU), Boston (ZBW), Atlanta (ZTL), Kansas City (ZKC), Washington, DC (ZDC), Indianapolis (ZID), Jacksonville (ZJX), Los Angeles (ZLA), Cleveland (ZOB) and New York (ZNY), each have more than ten of the 250G nodes. The Centers in the western part of the United States have fewer 250G nodes indicative of the lower traffic density in these Centers. It is informative to express the distribution of the 250G nodes in units of number of nodes per 10,000 square nautical miles (10 Ksqnm). Using this measure of nodal traffic density, ZNY has the highest nodal density with six 250G nodes per 10 Ksqnm. Table 3 shows the nodal traffic density for the 20 Centers in the continental US.

The behavior of the degree of the nodes in an ATM network should not be surprising, since the ATM network has evolved to serve population densities in the US. Network analysis helps to visualize and quantify the characteristics of the network.

Resilience of Air Traffic Management Networks

The tolerance of complex networks to random and targeted failures depends on their network structure. As ob-

served earlier, a scale-free network is tolerant to random failure, since the hubs are few and the chance of a hub being selected randomly is low. However, the same network may be prone to targeted attacks on a small percentage of vital nodes. In air traffic management networks, weather can be regarded as an agent of attacks on the system.

Convective weather is a major source of uncertainty in ATM networks. One effect of severe weather is to make airspace unavailable for the flow of air traffic. The removal of airspace may result in the failure of hubs and increase the average path length in the network. The impact of weather on ATM system performance appears as delay [20]. The tolerance of the ATM network to weather depends on the geographical distribution of the weather and the coverage of nodes with high degree.

Future Air Traffic Scenarios

National and international projections of traffic growth indicate a tripling of passengers by 2025 [7]. There may be increased traffic due to the growing presence of on-demand air taxis and unmanned air vehicles. It is estimated that 5000 micro jets may be operational by 2010 and 13 500 by 2022 [22]. The TAF provides forecasts for airports in the NAS. The forecast for the major airports in the United States receives more detailed modeling that takes into account local economic conditions and airline costs. TAF is the basis for most aviation demand forecasts. The TAF data published in March 2006 provides traffic growth rates for the period 2004–2020. The AvDemand model [16] provides three times (3X) current traffic scenarios starting with TAF and assuming slightly different conditions for traffic growth beyond 2025. The Transportation System Analysis Model (TSAM) [23] predicts

future traffic demand based on demographics and population at the county level. It provides a complimentary alternative to the FAA forecast. The results in this study are based on data generated by AvDemand.

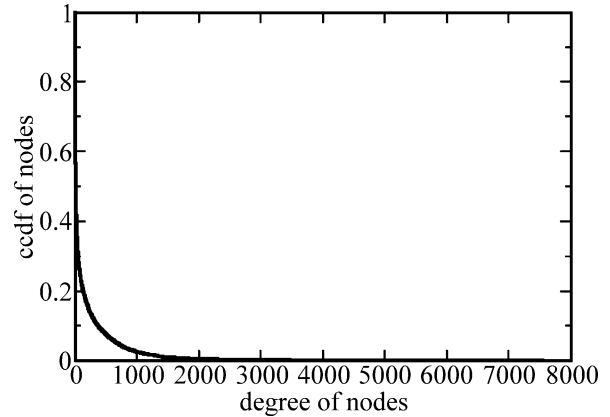
Two future scenarios are considered. These two scenarios and the current traffic provide a baseline to compare the impact of future traffic changes on the ATM system. The initial values of aircraft routes and schedules used in the extrapolation are based on traffic data for a day in May 2002. Assuming traffic growth from TAF and the nominal traffic schedule, Fratar algorithm [23] can be used to create future daily total number of flights between each origin and destination pair of the current schedule. The traffic growth scenarios using TAF airport growth out to 2025 predict a future NAS-wide traffic growth by a factor of 1.49. To achieve 3-times current day traffic (3X) scenarios assumed in future planning [9], consider two different scenarios for post-2025 traffic growth. The compound extrapolation approach grows traffic until it reaches 3X by assuming the TAF airport growth rates for traffic beyond 2025. As an alternative, the homogeneous extrapolation approach assumes the same growth rate at all airports until the traffic level reaches 3X. These two scenarios will be referred to as 3X Compound (3XC) and 3X Homogeneous (3XH) scenarios in the rest of the chapter. The scenarios are processed to compute the changes in the network properties of future ATM systems.

Behavior of Future ATM Networks

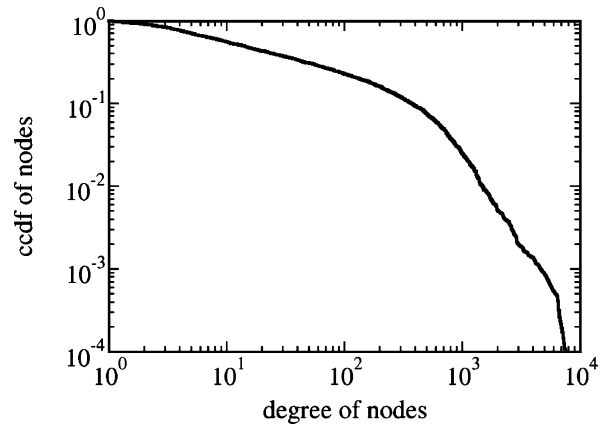
The two scenarios described earlier can be used to study the characteristics of the 3X traffic under the two assumptions. The properties of future ATM networks can be derived similar to the properties of the current ATM networks. The computations can be used to compare the geographical distribution of the hubs compared to the distribution today. Figures 11 and 12 show the distribution of the nodes in the AAN network for the 3XC traffic scenario. The results are similar for the 3XH traffic scenario.

The future ATM networks, under both scenarios, show exponentially truncated scale-free behavior similar to the current ATM network. The distribution of the hubs will have an impact on the growth of delay in future ATM networks subjected to severe weather. The vulnerability of the network to reduction in capacity caused by certain weather patterns will be significant and disruptive to the operation of the system. The impact of the weather on future ATM networks will be described subsequently in the paper.

The geographical distribution of the 250G nodes under 3XC is shown in Fig. 13. Table 2 shows the same distribu-



Air Traffic Control, Complex Dynamics of, Figure 11
Distribution of nodes in AAN using 3X Compound Scenario



Air Traffic Control, Complex Dynamics of, Figure 12
Logarithmic behavior of nodes in network AAN under 3X Compound traffic



Air Traffic Control, Complex Dynamics of, Figure 13
Geographical distribution of 250G nodes in network N5 with 3X Compound traffic and severe weather polygons

Air Traffic Control, Complex Dynamics of, Table 2

Distribution of 250G nodes by Centers under current and future traffic scenarios

Center	Current	3XH	3XC
ZAB	7	70	78
ZAU	15	67	70
ZBW	16	77	87
ZDC	22	134	143
ZDV	8	51	68
ZFW	3	59	68
ZHU	8	60	72
ZID	15	71	71
ZJX	14	71	81
ZKC	11	58	50
ZLA	16	102	119
ZLC	5	29	40
ZMA	4	44	54
ZME	10	38	42
ZMP	6	46	53
ZNY	14	80	88
ZOA	10	57	67
ZOB	13	90	89
ZSE	5	35	41
ZTL	15	59	62
Total	217	1298	1443

tion by Centers for current traffic and 3X scenarios. An examination of Table 2 is helpful in drawing distinctions between current day traffic and 3X demand traffic. The 3X traffic demand creates close to a six-fold increase in the 250G traffic nodes. Earlier, it was noted that ZNY has the highest 250G traffic nodal density. Under the 3X demand, as can be seen from Table 3, ZNY nodal density of 2006 is equaled or exceeded by a majority of the 20 Centers. Even more alarmingly, the nodal density of 250G nodes in ZNY is 6 times the current value, and Cleveland and Washington Centers have twice the nodal density of ZNY in 2006. The 3X traffic also gives rise to nodes with even more connections. The hubs, 500G, 750G and 1000G, are defined similarly to the 250G hub. The number of 250G, 500G, 750G and 1000G hubs under different traffic conditions are shown in Table 4.

Impact of Weather on ATM Networks

Another way to view the growth of traffic is to compare how similar weather patterns may affect current traffic and 3X traffic. The geographical distribution of the 250G nodes shown in Fig. 13 is based on traffic during July 2, 2006. This was a calm weather day with a total NAS aggregate

Air Traffic Control, Complex Dynamics of, Table 3

Nodal density of 250G nodes by Centers under current and future traffic scenarios

Center	Area	ρ_{1X}	ρ_{3XH}	ρ_{3XC}
ZAB	18.04	0.39	3.88	4.32
ZAU	7.60	1.97	8.82	9.21
ZBW	11.62	1.38	6.63	7.49
ZDC	12.69	1.73	10.56	11.27
ZDV	20.18	0.40	2.53	3.37
ZFW	12.32	0.24	4.79	5.52
ZHU	27.64	0.29	2.17	2.61
ZID	7.07	2.12	10.04	10.04
ZJX	14.55	0.96	4.88	5.57
ZKC	13.37	0.82	4.34	3.74
ZLA	13.55	1.18	7.53	8.78
ZLC	31.57	0.16	0.92	1.27
ZMA	28.66	0.14	1.53	1.88
ZME	10.75	0.93	3.53	3.91
ZMP	28.82	0.21	1.60	1.84
ZNY	2.43	5.76	32.92	36.21
ZOA	13.54	0.74	4.21	4.95
ZOB	6.74	1.93	13.36	13.21
ZSE	19.64	0.25	1.78	2.09
ZTL	9.18	1.63	6.43	6.75
Total	309.96	0.70	4.19	4.66

Air Traffic Control, Complex Dynamics of, Table 4

Total number of nodes for different traffic scenarios

Types of nodes	Current	3XH	3XC
250G	217	1312	1468
500G	72	620	806
750G	33	304	452
1000G	22	165	262

delay of 11 997 minutes [15]. The traffic on July 2, 2006 will be assumed as traffic unaffected by weather in this discussion [20].

Whenever there is severe weather in the NAS, airspace capacity is reduced and traffic is rerouted or held on the ground causing delays in the system. The Collaborative Convective Forecast Product (CCFP) is a model of severe weather activity and the areas marked blue in Fig. 10 shows the CCFP for July 13, 2006. On July 13, 2006 the NAS experienced a significant total delay of 219 350 minutes with ZNY, ZDC and ZTL Centers contributing a delay of 116 654, 49 400 and 23 822 minutes respectively. Next, as shown in Fig. 13, the same severe weather is overlaid in blue polygons on the geographical distribution of the 250G

Air Traffic Control, Complex Dynamics of, Table 5
Total number of nodes affected by severe weather

Types of Nodes	1X	3XH	3XC
250G	34	191	207
500G	9	96	118
750G	4	49	67
1000G	3	23	41

nodes under 3XC demand. Table 5 shows hub nodes affected by the weather today and in the future.

A greater number of hubs, between six to ten times, are affected by the same weather pattern under future traffic scenarios than today. If one considers the non-linear growth of delay, in regions such as ZNY, ZDC and ZTL where the demand on the airspace is close to capacity, the increased density of high traffic nodes in these regions will result in much larger delays compared to 2006.

Concluding Remarks

Air traffic in the United States can be modeled as a network to understand the impact of predicted growth in demand on the performance of the system. It is demonstrated that the air traffic network with current en route flight plan intersections as nodes and with the flight plans as links shows scale-free properties typical of several large engineering and biological networks. A consequence of this property is the non-linear growth of traffic in certain regions of the United States. A preliminary analysis indicates that a three-times growth in the overall traffic may result in a ten-times impact on the density of traffic in certain parts of the United States. The air traffic system currently experiences significant delay during periods of severe weather activity. The impact of weather of same severity will be magnified several times, especially in the northeastern parts of the United States, leading to lower system performance. Research must be conducted to determine whether this risk can be mitigated through restructuring routes or by introducing new operational concepts, such as automation assistance to controllers to increase airspace capacity. The network analysis described in the chapter can be used to guide the development of various traffic flow management concepts to increase the efficiency of air traffic systems.

Bibliography

1. Airbus SAS (2002) Airbus Global Market Forecast 2001–2002. Reference CB 390.0008/02
2. Albert R, Jeong H, Barabasi AL (2000) Error and attack tolerance of complex networks. *Nature* 406:378–382
3. Amaral LAN, Scala A, Barthelemy M, Stanley HE (2000) Classes of Small-world Networks. *Proc Natl Acad Sci USA* 97:1149–1152
4. Anonymous (2005) Terminal Area Forecast Summary, Fiscal Years 2004–2020. Report No. FAA-APO-05-1, US Department of Transportation, Federal Aviation Administration
5. Anonymous (2006) Commercial Aviation and the American Economy. The Campbell-Hill Aviation Group Inc, Washington DC
6. Barabasi AL, Bonabeau E (2003) Scale-Free Networks. *Sci Am* 288:60–69
7. Boeing (2007) Current Market Outlook. <http://www.boeing.com/commercial/cmo> Accessed June 26, 2007
8. Bonnefoy PA, Hansman RJ (2004) Emergence and Impact of Secondary Airports in the United States. 4th AIAA Aviation Technology, Integration and Operations Conference, Chicago, IL, September 2004
9. Borener S, Carr G, Ballard D, Hasan S (2006) Can NGATS Meet the Demands of the Future? *J Air Traffic Control* 48(1):34–38
10. Broder A, et al (2000) Graph structure in the web. *Comput Netw* 33:309–320
11. DeLaurentis D, Han E, Kotegawa T (2006) Establishment of a Network-based Simulation of Future Air Transportation Concepts. 6th AIAA Aviation Technology, Integration and Operations Conference, Wichita, KS, September 2006
12. Donohue G (2003) Air Transportation is a complex Adaptive System: Not an Aircraft Design. AIAA/ICAS International Air and Space Symposium and Exposition, Dayton, Ohio, July 2003
13. Erdos P, Renyi A (1960) On the Evolution of Random Graphs. *Publ Math Inst, Hung Acad Sci* 5:17–61
14. Eurocontrol (2007) An Assessment of Air Traffic Management in Europe during the Calendar Year 2006: Performance Review Report. Eurocontrol, Brussels, Belgium
15. Federal Aviation Administration (2004) Order 7210.55C: Operational Data Reporting Requirements. US Department of Transportation
16. Huang AS, Schleicher D, Hunter G (2006) Future Flight demand Generation Tool. 6th AIAA Aviation Technology, Integration and Operations Conference, Wichita, KS, September 2006
17. Newman MEJ (2003) The structure and function of complex networks. *SIAM Rev* 45:167–256
18. Pearce RA (2006) The next generation air transportation system: Transformation starts now. *J Air Traffic Control* 48(1):7–10
19. Sridhar B, Sheth K, Smith P, Leber W (2005) Migration of FACET from Simulation Environment to Dispatcher Decision Support System. Proceedings of Digital Avionics Systems Conference, Washington, DC, November 2005
20. Sridhar B, Swei SM (2006) Relationship between Weather, Traffic and Delay Based on Empirical Methods. 6th AIAA Aviation Technology, Integration and Operations Conference, Wichita, KS, September 2006
21. Strogatz SH (2001) Exploring complex networks. *Nature* 410:268–276
22. Teal Group (2003) World Unmanned Aerial Vehicle Systems. Market Profile and Forecast, Fairfax
23. Viken J, et al (2006) NAS Demand Predictions, Transportation Systems Analysis Model (TSAM) Compared with Other Forecasts. 6th AIAA Aviation Technology, Integration and Operations Conference, Wichita, KS, September 2006

Algorithmic Complexity and Cellular Automata

JULIEN CERVELLE¹, ENRICO FORMENTI²

¹ Laboratoire d'Informatique de l'Institut
Gaspard-Monge, Université Paris-Est,
Marne la Vallée, France

² Laboratoire I3S,
Université de Nice-Sophia Antipolis,
Sophia Antipolis, France

Article Outline

Glossary
Definition of the Subject
Introduction
Kolmogorov Complexity
Dynamical Systems and Symbolic Factors
Cellular Automata
Algorithmic Complexity as a Demonstration Tool
Measuring CA Structural Complexity
Measuring the Complexity of CA Dynamics
Future Directions
Acknowledgments
Bibliography

Glossary

Algorithmic complexity of object x shortest program for outputting a description for x (w.r.t. a universal representation system)
Equicontinuity all points are equicontinuity points (in compact settings)
Equicontinuity point a point for which the orbits of nearby points remain close
Expansivity from two distinct points orbits eventually separate
Incompressible word a word for which the shortest program outputting it has “almost” the same length as the word itself
Injectivity the next state function is injective
Kolmogorov complexity see “algorithmic complexity”
Rich configuration a configuration that contains all possible finite patterns over a given alphabet
Sensitivity to initial conditions for any point x there exist arbitrary close points whose orbits eventually separate from the orbit of x
Surjectivity the next state function is surjective
Transitivity there always exist points that eventually move from any arbitrary neighborhood to any other

Definition of the Subject

In the last 10 years the field of complex systems has enjoyed astonishing development with meaningful applications in most scientific domains.

Cellular automata (CA) are a very used model for complex systems characterized by a multitude of small identical agents capable of building a complex behavior on the basis of local interactions.

The huge variety of CA dynamical behaviors has been popularized since the early 1980s by the work of S. Wolfram (see [18] for an exhaustive review).

Later, researchers started a systematic study of CA. Most of the results are summarized in the chapters of this book (see ► [Chaotic Behavior of Cellular Automata](#), ► [Dynamics of Cellular Automata in Non-compact Spaces](#), ► [Topological Dynamics of Cellular Automata](#) and ► [Ergodic Theory of Cellular Automata](#) for instance).

However, many questions seem to be intractable and vanquish researchers' efforts (some such efforts are reported by ► [Chaotic Behavior of Cellular Automata](#), ► [Dynamics of Cellular Automata in Non-compact Spaces](#), ► [Topological Dynamics of Cellular Automata](#) and [14]). We strongly believe that Kolmogorov complexity can be a valuable tool in the quest for answers in this domain.

Introduction

Kolmogorov (or algorithmic) complexity was introduced to precisely formalize the notion of “randomness.” Former definitions of randomness were based on probabilistic concepts: for instance, a sequence of 0 and 1 is *random* if it is obtained by repeated coin tosses.

The drawback of such a definition is that it does not formally specify what is and what is not a random sequence since all sequences (of a fixed length) have the same probability to be obtained through unbiased coin tosses. However, if one considers the following two sequences:

00000000000000000000000000000000

10010010010111010110101110101

one would say that the first one is not random while the second one is.

The idea of Kolmogorov complexity comes from this simple reasoning. What makes the first sequence simple is that it can briefly be described by the sentence “twenty-nine zeros,” while the second cannot be described in a shorter way than by describing it literally: “the sequence one zero zero one zero zero one zero zero one

zero one one one zero one zero one one zero one zero one one one zero one zero one.” If we use regular expressions instead of English as a description language, the first sequence is 0^{29} and the second is itself.

A first idea to formalize this process is to say that the complexity of a sequence is the length of its compressed form for a given lossless compressor. The issue of such a definition is that there is no universal compression tool as compression usually depends on the type of data (sound, picture, text). However, Kolmogorov and Solomonoff proved that there exists a universally optimal compressor that compresses sequences better up to some additive constant w.r.t. any other compressor (Theorem 1).

Kolmogorov complexity is not computable, but its combinatorial properties make it a useful demonstration tool. The major technique in Kolmogorov complexity is the *incompressibility method*, and it is briefly reviewed in Sect. “[Algorithmic Complexity as a Demonstration Tool](#)”. For more on the subject see [13].

On the one hand, Kolmogorov complexity allows one to give a formal context to study randomness in discrete dynamical systems; on the other hand, it is a helpful tool for decreasing the combinatorial complexity of problems. Both of these aspects of Kolmogorov complexity will figure prominently in this chapter.

Kolmogorov Complexity

In this section we give the formal definition of Kolmogorov complexity and its main properties. For simplicity sake we restrict ourselves to the alphabet $\{0,1\}$ although Kolmogorov complexity can be defined on more general alphabets and even on integers [13].

Let $\{0,1\}^*$ be the set of words on $\{0,1\}$, $|w|$ the length of word w and for $0 \leq i < |w|$, w_i the i th letter of w (indexed from 0), and ε the empty word.

A *partial computable function* is a computable function that is defined on a subset of $\{0,1\}^*$ (being undefined on an input means that the program that computes the function does not halt). Denote by D_φ the set of inputs on which φ is defined. The *composition* of partial computable functions f and g is the partial computable function $f \circ g$ defined on the set $\{x \in D_g, g(x) \in D_f\}$ by $f \circ g(x) = f(g(x))$. If f is injective, the *inverse* of f is the partial computable function f^{-1} defined by

$$f^{-1}(w) = \begin{cases} x & \text{the unique word such that } f(x) = w \text{ if it exists,} \\ \text{undefined} & \text{otherwise.} \end{cases}$$

If f is computable, then f^{-1} is computed by the algorithm, which tries all possible x and halts when it finds one whose image is w . If no such x exists, then the program does not halt.

A *representation system* is a partial computable function φ from $(\{0,1\}^*)^2$ to $\{0,1\}^*$. It plays the role of a decompressor program used to build words from their compressed form. There are no requirements, and, in particular, it is not mandatory either that each word have a compressed form (i. e., the image set of φ need not be $\{0,1\}^*$) or that a word have only one compressed form (i. e., φ need not be injective).

Definition 1 (Kolmogorov complexity given a representation system) Let φ be a representation system. The *Kolmogorov complexity given φ of a word w knowing v* , denoted by $K_\varphi(w|v)$, is defined by the length of the shortest word y such that $\varphi(y, v) = w$ or $+\infty$ if such a word does not exist:

$$K_\varphi(w|v) = \min\{|y|, y \in \{0,1\}^* \text{ s.t. } \varphi(y, v) = w\} \\ \text{with the convention } \min \emptyset = \infty.$$

If y is such that $\varphi(y, v) = w$, we say that y is a *program* for w knowing v . If y is the shortest word such that $\varphi(y, v) = w$, we say that y is the *shortest program* for w knowing v . The *Kolmogorov complexity given φ of a word w* , denoted $K_\varphi(w)$, is defined by

$$K_\varphi(w) = K_\varphi(w|\varepsilon).$$

If y is such that $\varphi(y, \varepsilon) = w$, we say that y is a *program* for w . If y is the shortest word such that $\varphi(y, \varepsilon) = w$, we say that y is the *shortest program* for w .

In the previous definition, the name *program* can be a little confusing since φ seems to be the program and y seems to be its input. This name comes from the additively optimal universal representation system introduced in Theorem 1, where the input is a program for the universal Turing machine.

As stated in the introduction, this definition lacks robustness since it depends on a particular (de)compressor. However, the following results prove that we can find a best compressor in order to define a robust notion of complexity.

Definition 2 (Additively optimal universal representation system) A representation system φ_0 is *additively optimal universal* if for any representation system φ there exists a constant c such that for all w and v in $\{0,1\}^*$

$$K_\varphi(w|v) + c \geq K_{\varphi_0}(w|v). \quad (1)$$

This definition means that a universal representation system compresses better than any other compression system up to some additive constant.

Remark 1 The constant in Inequality (1) is mandatory since a representation system cannot be better than any other one because of very specialized compressors: for any word w , the representation system δ_w by $\delta_w(y, v) = w$ for all words y and v is one of the best for word w (i.e., $K_{\delta_w}(w|v) = 0$).

Proposition 1 Let φ_0 and φ_1 be two additively optimal universal representation systems. Then, there is a constant c such that for all words w and v

$$|K_{\varphi_0}(w|v) - K_{\varphi_1}(w|v)| < c.$$

Proof From the definition of additively optimal universal representation system, there are constants c_0 and c_1 such that for all words w and v and for all representation systems φ

$$K_{\varphi}(w|v) + c_0 \geq K_{\varphi_0}(w|v) \text{ and } K_{\varphi}(w|v) + c_1 \geq K_{\varphi_1}(w|v).$$

The thesis is obtained by choosing $c = \max\{c_0, c_1\}$. \square

In order to define completely Kolmogorov complexity, one needs an additively optimal universal representation system. The key idea for conceiving such a system is to embed the program for decompression in the compressed form.

In order to join the decompressing program and the compressed word in one string, we need a special combining function. We denote by $w^{\times 2}$ a word u of length $2|w|$ such that for all $0 \leq i < |w|$, $u_{2i} = u_{2i+1} = w_i$, which is word w , where all letters are repeated twice. Define

$$p \odot w = p^{\times 2} 01w.$$

For instance, if $p = 001$ and $w = 1001011$, then $p \odot w = 000011 01 1001011$.

This way of combining words has two interesting properties. First, from $p \odot w$ a program can compute p and w since, while all bits are repeated twice, we read p , and after skipping the 01, we read w . Second, it holds that

$$|p \odot w| = 2|p| + 2 + |w|.$$

This last equation is fundamental to proving the additively optimality property.

Remark 2 The word $p^{\times 2} 01$ is called a self-delimiting encoding for p since it encodes p in a unique way and the

end of the code is computable. Hence, if one concatenates this code for p and another word, then one can compute back p and the second word. Using the same reasoning, from $p^{\times 2} 01 q^{\times 2} 01 w$ one can compute each of the three words p , q , and w .

Theorem 1 There exists an additively optimal universal representation system.

Proof Let φ_0 be the function defined from $(\{0, 1\}^*)^2$ to $\{0, 1\}^*$ by

$$\varphi_0: \begin{cases} (\{0, 1\}^*)^2 \rightarrow \{0, 1\}^*, \\ (p \odot w, v) \mapsto T_p(w, v), \end{cases}$$

where T_p is the function from $(\{0, 1\}^*)^2$ to $\{0, 1\}^*$ computed by the p th Turing machine (two tapes for the two inputs). From computability theory we know that this function is computable since there exists a universal Turing machine.

Let φ be a representation system. Since φ is computable, it is computed by a Turing machine whose number is p . For all words w and v , let y be a shortest program for w knowing v with representation system φ . Then

$$\varphi_0(p \odot y, v) = T_p(y, v) = \varphi(y, v) = w,$$

and so $p \odot y$ is a program (perhaps not the shortest) for w knowing v with representation system φ_0 . Since

$$|p \odot y| = 2|p| + 2 + |y| = 2|p| + 2 + K_{\varphi}(w|v),$$

it holds that, for all words w ,

$$K_{\varphi_0}(w|v) \leq K_{\varphi}(w|v) + 2|p| + 2.$$

As p depends only on φ , we have that for all representation systems φ , there is a constant $c = 2|p| + 2$ such that

$$K_{\varphi_0}(w|v) < K_{\varphi}(w|v) + c.$$

We conclude that φ_0 is an additively optimal representation system. \square

Now we are able to define Kolmogorov complexity.

Definition 3 Let φ_0 be an additively optimal representation system. The *Kolmogorov complexity of a word w knowing v* is defined by

$$K(w|v) = K_{\varphi_0}(w|v),$$

and the *Kolmogorov complexity of a word w* is defined by

$$K(w) = K_{\varphi_0}(w) = K(w|\varepsilon).$$

Remark 3 Kolmogorov complexity is defined up to an additive constant from Proposition 1. This means that any meaningful result must include in its statement something like “there is a constant c such that.”

Properties

Basic Relations Kolmogorov complexity allows one to express relations that seem natural between complexities of words. For instance, the complexity of ww is close to the complexity of w . The complexity of uv is less than the sum of the complexities of u and v (in fact up to the logarithm of the size of u or v). They will be used in the sequel.

We give here a few examples of how to prove such results.

Theorem 2 *There exists a constant c such that for all words w and v ,*

$$K(w|v) < |w| + c.$$

Proof Let π be the representation system defined by $\pi(w, x) = w$. As φ_0 is additively optimal, there is a constant c such that

$$K(w|v) < K_\pi(w|v) + c.$$

Since w is the only word such that $\pi(w) = w$, one has $K_\pi(w|v) = |w|$. We conclude that

$$K(w|v) < K_\pi(w|v) + c = |w| + c.$$

□

This result is interesting since it gives a computable upper bound for Kolmogorov complexity. Moreover, in Subsect. “Incompressible Words” we will see that this bound is tight.

Theorem 3 *Let f and g be partial computable functions. There exists a constant c such that for all words $w \in D_f$ and all $v \in D_g$*

$$K(f(w)|v) < K(w|g(v)) + c.$$

Proof As f and g are computable, function h , defined by $h(x, y) = f(\varphi_0(x, g(y)))$, is a representation system. Hence, by additive optimality of φ_0 , there is a constant c such that for all w and v

$$K(f(w)|v) < K_h(f(w)|v) + c. \quad (2)$$

Let x be a shortest program for w knowing $g(v)$ with representation system φ_0 . Since $\varphi_0(x, g(v)) = w$, $h(x, v) = f(\varphi_0(x, g(v))) = f(w)$, and so x is a program for $f(w)$ given v with representation system h . We conclude that

$$K_h(f(w)|v) \leq |x| = K(w|g(v)),$$

and hence, from Eq. (2),

$$K(f(w)|v) < K_h(f(w)|v) + c \leq K(w|g(v)) + c.$$

□

This proposition formalizes the intuition that if a word w can be computed from x , then w is simpler than x , of course up to some additive constant. This means that w contains less information than x . Of course, if f is injective, as we can go back from $f(w)$ to w , then w and $f(w)$ have the same amount of information. This is stated by the next corollary.

Corollary 1 *Let f and g be injective partial computable functions. There exists a constant c such that for all words w and v ,*

$$|K(f(w)|v) - K(w|g(v))| < c.$$

Proof From the previous theorem, as f , g , f^{-1} , and g^{-1} are computable, there are constants c_1 and c_2 such that for all w and v

$$K(f(w)|v) < K(w|g(v)) + c_1 \text{ and}$$

$$K(f^{-1}(w)|v) < K(w|g^{-1}(v)) + c_2.$$

Hence, $K(w|g(v)) < K(f(w)|v) + c_2$. Choosing $c = \max(c_1, c_2)$ completes the proof. □

Using this corollary, one finds that the complexity of the representation by binary strings of a mathematical object depends only on the mathematical object and not on its representation, provided that an algorithm can compute one representation from another. Hence, for any mathematical object o , we denote by notation abuse $K(o)$ the complexity of a representation of o with alphabet $\{0, 1\}$. In particular, we use this for integers, finite sets, graphs, etc.

However, for objects that cannot be represented by finite binary strings (for instance, real numbers, infinite strings), we cannot define a Kolmogorov complexity. In this case, a per-case definition has to be given since for infinite objects several reasonable definitions of complexity can be found. For instance, for infinite strings one can use the limit superior or the limit inferior of the conditional complexity of the prefix given its length. One can use the limit superior of the complexity of the prefix divided by its length, etc.

Now we state a result about the complexity of a pair compared to the complexity of each member.

Theorem 4 *There is constant c such that, for any words x , y , and v ,*

$$K(\langle x, y \rangle | v) < K(x|v) + K(y|\langle v, x \rangle) + 2 \log_2(K(x|v)) + c,$$

where $\langle \cdot, \cdot \rangle$ is any bijective computable function between \mathbb{N}^2 and \mathbb{N} .

Proof Let \widehat{x}, \widehat{y} be defined by $\widehat{x}, \widehat{y} = \ell(x)^{\times 2} 01xy$, where $\ell(x)$ is the length of x written in binary. Note that x and y can both be reconstructed from \widehat{x}, \widehat{y} and that

$$|\widehat{x}, \widehat{y}| = 2 + 2\lceil \log_2 |x| \rceil + |x| + |y|.$$

Let φ be the representation system defined by

$$\varphi(\widehat{x}, \widehat{y}, v) = \langle \varphi_0(x, v), \varphi_0(y, \langle v, \varphi_0(x, v) \rangle) \rangle.$$

As φ_0 is additively optimal, there is a constant c such that for all w and v

$$K(w|v) < K_\varphi(w|v) + c.$$

Let z be a shortest program for x knowing v , and let z' be a shortest program for y knowing $\langle v, x \rangle$. As $\widehat{z}, \widehat{z}'$ is a program for $\langle x, y \rangle$ knowing v with representation system φ , one has

$$\begin{aligned} K(\langle x, y \rangle | v) & < K_\varphi(w|v) + c \leq |\widehat{z}, \widehat{z}'| + c \\ & \leq K(x|v) + K(y|\langle v, x \rangle) + 2\log_2(K(x|v)) + 3 + c, \end{aligned}$$

which completes the proof. \square

Of course, this inequality is not an equality since if $x = y$, the complexity of $K(\langle x, y \rangle)$ is the same as $K(x)$ up to some constant. When the equality holds, it is said that x and y have no mutual information.

Note that if we can represent the combination of programs z and z' in a shorter way, the upper bound is decreased by the same amount. For instance, if we choose $\widehat{x}, \widehat{y} = \ell(\ell(x))^{\times 2} 01\ell(x)xy$, it becomes

$$2\log_2 \log_2 |x| + \log_2 |x| + |x| + |y|.$$

If we know that a program never contains a special word u , then with $\widehat{x}, \widehat{y} = xuy$ it becomes $|x| + |y|$.

Examples The properties seen so far allow one to prove most of the relations between complexities of words.

For instance, choosing $f: w \mapsto ww$ and g being the identity in Corollary 1, we obtain that there is a constant c such that $|K(ww|v) - K(w|v)| < c$.

In the same way, letting f be the identity and g be defined by $g(x) = \varepsilon$ in Theorem 3, we get that there is a constant c such that $K(w|v) < K(w) + c$.

By Theorem 4, Corollary 1, and choosing f as $\langle x, y \rangle \mapsto xy$, and g as the identity, we have that there is a constant c such that

$$K(xy) < K(x) + K(y) + 2 \min \log_2(K(x)), \log_2(K(y)) + c.$$

Incompressible Words Kolmogorov complexity gives a computer-science notion of randomness. By Theorem 2, we have an upper bound on Kolmogorov complexity: the length of the word. When this upper bound is reached, we say that the word is random. However, Kolmogorov complexity is difficult to use since it is not computable (Theorem 2.3.2 in [13]). The good news is that it is approximable from above.

The following definition formalizes the notion of incompressible word.

Definition 4 Let $c \in \mathbb{R}^+$. Word w is c -incompressible if

$$K(w) \geq |w| - c.$$

The next result proves the existence of incompressible words.

Proposition 2 For any $c \in \mathbb{R}^+$, there are at least $(2^{n+1} - 2^{n-c})$ c -incompressible words w such that $|w| \leq n$.

Proof Each program produces only one word; therefore there cannot be more words whose complexity is below n than the number of programs of size n . Hence, one finds

$$|\{w, K(w) < n\}| \leq 2^n - 1.$$

This implies that

$$\begin{aligned} |\{w, K(w) < |w| - c \text{ and } |w| \leq n\}| \\ \leq |\{w, K(w) < n\}| < 2^{n-c}, \end{aligned}$$

and, since

$$\begin{aligned} \{w, K(w) \geq |w| - c \text{ and } |w| \leq n\} \\ \subset \{w, |w| \leq n\} \setminus \{w, K(w) < |w| - c \text{ and } |w| \leq n\}, \end{aligned}$$

it holds that

$$|\{w, K(w) \geq |w| - c \text{ and } |w| \leq n\}| > 2^{n+1} - 2^{n-c}.$$

\square

From this proposition one deduces that half of the words whose size is less than or equal to n reach the upper bound of their Kolmogorov complexity. This incompressibility method relies on the fact that most words are incompressible.

Martin-Löf Tests

Martin-Löf tests give another equivalent definition of random sequences. The idea is simple: a sequence is random if it does not pass any computable test of singularity (i. e.,

a test that selects words from a “negligible” recursively enumerable set). As for Kolmogorov complexity, this notion needs a proper definition and implies a universal test. However, in this review we prefer to express tests in terms of Kolmogorov complexity. (We refer the reader to [13] for more on Martin–Löf tests.)

Let us start with an example. Consider the test “to have the same number of 0s and 1s.” Define the set

$$E = \{w, w \text{ has the same number of 0s and 1s}\}$$

and order it in military order (length first, then lexicographic order), $E = \{e_0, e_1, \dots\}$. Consider the computable function $f: x \mapsto e_x$ (which is in fact computable for any decidable set E). Note that there are $\binom{2n}{n}$ words in E of length $2n$. Thus, if e_x has length $2n$, x is less than $\binom{2n}{n}$, whose logarithm is $2n - \log_2 \sqrt{n} + O(1)$. Then, using Theorem 3 with function f , one finds that

$$K(e_x) < K(x) < \log_2 x < |e_x| - \log_2 \sqrt{\frac{|e_x|}{2}},$$

up to an additive constant. We conclude that all members e_x of E are not c -incompressible whenever they are long enough.

This notion of test corresponds to “belonging to a small set” in Kolmogorov complexity terms. The next proposition formalizes these ideas.

Proposition 3 *Let E be a recursively enumerable set such that $|E \cap \{0, 1\}^n| = o(2^n)$. Then, for all constants c , there is an integer M such that all words of length greater than M of E are not c -incompressible.*

Proof In this proof, we represent integers as binary strings. Let \widehat{x}, \widehat{y} be defined, for integers x and y , by

$$\widehat{x}, \widehat{y} = |x|^{\times 2} 01xy.$$

Let f be the computable function

$$f: \begin{cases} \{0, 1\}^* & \rightarrow \{0, 1\}^* , \\ \widehat{x}, \widehat{y} & \mapsto \text{the } y\text{th word of length } x \\ & \text{in the enumeration of } E. \end{cases}$$

From Theorems 2 and 3, there is a constant d such that for all words u of E

$$K(e) < |f^{-1}(e)| + d.$$

Note $u_n = |E \cap \{0, 1\}^n|$. From the definition of function f one has

$$|f^{-1}(e)| \leq 3 + 2 \log_2(|e|) + \log_2(u_{|e|}).$$

As $u_n = o(2^n)$, $\log_2(u_{|e|}) - |e|$ tends to $-\infty$, so there is an integer M such that when $|e| \geq M$, $K(e) < |f^{-1}(e)| + d < |e| - c$. This proves that no members of E whose length is greater than M are c -incompressible. \square

Dynamical Systems and Symbolic Factors

A (discrete-time) dynamical system is a structure $\langle X, f \rangle$ where X is the set of *states* of the system and f is a function from X to itself that, given a state, tells which is the next state of the system. In the literature, the state space is usually assumed to be a compact metric space and f is a continuous function.

The study of the asymptotic properties of a dynamical system is, in general, difficult. Therefore, it can be interesting to associate the studied system with a simpler one and deduce the properties of the original system from the simple one. Indeed, under the compactness assumption, one can associate each system $\langle X, f \rangle$ with its *symbolic factor* as follows.

Consider a finite open covering $\{\beta_0, \beta_1, \dots, \beta_k\}$ of X . Label each set β_i with a symbol α_i . For any orbit of initial condition $x \in X$, we build an infinite word w_x on $\{\alpha_0, \alpha_1, \dots, \alpha_k\}^*$ such that for all $i \in \mathbb{N}$, $w_x(i) = \alpha_j$ if $f^i(x) \in \beta_j$ for some $j \in \{0, 1, \dots, k\}$. If $f^i(x)$ belongs to $\beta_j \cap \beta_h$ (for $j \neq h$), then arbitrarily choose either α_j or α_h . Denote by S_x the set of infinite words associated with the initial condition $x \in X$ and set $S = \bigcup_{x \in X} S_x$. The system $\langle S, \sigma \rangle$ is the symbolic system associated with $\langle X, f \rangle$; σ is the *shift map* defined as $\forall x \in S \forall i \in \mathbb{N}$, $\sigma(x)_i = x_{i+1}$.

When $\{\beta_0, \beta_1, \dots, \beta_k\}$ is a clopen partition, then $\langle S, \sigma \rangle$ is a factor of $\langle X, f \rangle$ (see [12] for instance).

Dynamical systems theory has made great strides in recent decades, and a huge quantity of new systems have appeared. As a consequence, scientists have tried to classify them according to different criteria. From interactions among physicists was born the idea that in order to simulate a certain physical phenomenon, one should use a dynamical system whose complexity is not higher than that of the phenomenon under study. The problem is the meaning of the word “complexity”; in general it has a different meaning for each scientist. From a computer-science point of view, if we look at the state space of factor systems, they can be seen as sets of bi-infinite words on a fixed alphabet. Hence, each factor $\langle S, \sigma \rangle$ can be associated with a language as follows. For each pair of words u, w write $u < v$ if u is a factor of w . Given a finite word u and a bi-infinite word v , with an abuse of notation we write $u < v$ if u occurs as a factor in v . The language $L(v)$ associated with a bi-infinite word $v \in A^*$ is defined as

$$L(v) = \{u \in A^*, u < v\}.$$

Finally, the language $L(S)$ associated with the symbolic factor $\langle S, \sigma \rangle$ is given by

$$L(S) = \{u \in A^*, u < v\}.$$

The idea is that the complexity of the system $\langle X, f \rangle$ is proportional to the language complexity of its symbolic factor $\langle S, \sigma \rangle$ (see, for example, [► Topological Dynamics of Cellular Automata](#)).

In [4,5], Brudno proposes to evaluate the complexity of symbolic factors using Kolmogorov complexity. Indeed, the complexity of the orbit of initial condition $x \in X$ according to finite open covering $\{\beta_0, \beta_1, \dots, \beta_k\}$ is defined as

$$K(x, f, \{\beta_0, \beta_1, \dots, \beta_k\}) = \limsup_{n \rightarrow \infty} \min_{w \in S_x} \frac{K(w_{0:n})}{n}.$$

Finally, the complexity of the orbit x is given by

$$K(x, f) = \sup K(x, f, \beta),$$

where the supremum is taken w.r.t. all possible finite open coverings β of X . Brudno has proven the following result.

Theorem 5 ([4]) *Consider a dynamical system $\langle X, f \rangle$ and an ergodic measure μ . For μ -almost all $x \in X$, $K(x, f) = H_\mu(f)$, where $H_\mu(f)$ is the measure entropy of $\langle X, f \rangle$ (for more on the measure entropy see, for example, [► Ergodic Theory of Cellular Automata](#)).*

Cellular Automata

Cellular automata (CA) are (discrete-time) dynamical systems that have received increased attention in the last few decades as formal models of complex systems based on local interaction rules. This is mainly due to their great variety of distinct dynamical behavior and to the possibility of easily performing large-scale computer simulations. In this section we quickly review the definitions and useful results, which are necessary in the sequel.

Consider the set of configurations C , which consists of all functions from \mathbb{Z}^D into A . The space C is usually equipped with the Cantor metric d_C defined as

$$\forall a, b \in C, \quad d_C(a, b) = 2^{-n},$$

$$\text{with } n = \min_{\vec{v} \in \mathbb{Z}^D} \{ \|\vec{v}\|_\infty : a(\vec{v}) \neq b(\vec{v}) \}, \quad (3)$$

where $\|\vec{v}\|_\infty$ denotes the maximum of the absolute value of the components of \vec{v} . The topology induced by d_C coincides with the product topology induced by the discrete topology on A . With this topology, C is a compact, perfect, and totally disconnected space.

Let $N = \{\vec{u}_1, \dots, \vec{u}_s\}$ be an ordered set of vectors of \mathbb{Z}^D and $f: A^s \mapsto A$ be a function.

Definition 5 (CA) The D -dimensional CA based on the local rule δ and the neighborhood frame N is the pair

$\langle C, f \rangle$, where $f: C \mapsto C$ is the global transition rule defined as follows:

$$\forall c \in C, \quad \forall \vec{v} \in \mathbb{Z}^D, \\ f(c)(\vec{v}) = \delta(c(\vec{v} + \vec{u}_1), \dots, c(\vec{v} + \vec{u}_s)). \quad (4)$$

Note that the mapping f is (uniformly) continuous with respect to d_C . Hence, the pair $\langle C, f \rangle$ is a proper (discrete-time) dynamical system.

In [7], a new metric on the phase space is introduced to better match the intuitive idea of chaotic behavior with its mathematical formulation. More results in this research direction can be found in [1,2,3,15]. This volume dedicates a whole chapter to the subject ([► Dynamics of Cellular Automata in Non-compact Spaces](#)). Here we simply recall the definition of the Besicovitch distance since it will be used in Subsect. “Example 2”.

Consider the Hamming distance between two words uv on the same alphabet A , $\#(u, v) = |\{i \in \mathbb{N} \mid u_i \neq v_i\}|$. This distance can be easily extended to work on words that are factors of bi-infinite words as follows:

$$\#_{h,k}(u, v) = |\{i \in [h, k] \mid u_i \neq v_i\}|,$$

where $h, k \in \mathbb{Z}$ and $h < k$. Finally, the *Besicovitch pseudodistance* is defined for any pair of bi-infinite words u, v as

$$d_B(u, v) = \limsup_{n \rightarrow \infty} \frac{\#_{-n,n}(u, v)}{2n+1}.$$

The pseudodistance d_B can be turned into a distance when taking its restriction to $A^{\mathbb{Z}}|_{\equiv}$, where \equiv is the relation of “being at null d_B distance.” Roughly speaking, d_B measures the upper density of differences between two bi-infinite words.

The *space-time diagram* Γ is a graphical representation of a CA orbit. Formally, for a D -dimensional CA f with state set A , Γ is a function from $A^{\mathbb{Z}^D} \times \mathbb{N} \times \mathbb{Z}$ to A defined as $\Gamma(x, i) = f^i(x)_j$, for all $x \in A^{\mathbb{Z}^D}$, $i \in \mathbb{N}$ and $j \in \mathbb{Z}$.

The *limit set* Ω_f contains the long-term behavior of a CA on a given set U and is defined as follows:

$$\Omega_f(U) = \bigcap_{n \in \mathbb{N}} f^n(U).$$

Unfortunately, any nontrivial property on the CA limit set is undecidable [11]. Other interesting information on a CA’s long-term behavior is given by the *orbit limit set* Δ defined as follows:

$$\Delta_f(U) = \bigcup_{u \in U} \mathcal{O}'_f(u),$$

where H' is the set of adherence points of H . Note that in general the limit set and the orbit limit sets give different information. For example, consider a *rich* configuration c i. e., a configuration containing all possible finite patterns (here we adopt the terminology of [6]) and the shift map σ . Then, $\Delta_\sigma(\{c\}) = A^\mathbb{Z}$, while $\Omega_\sigma(\{c\})$ is countable.

Algorithmic Complexity as a Demonstration Tool

The incompressibility method is a valuable formal tool to decrease the combinatorial complexity of problems. It is essentially based on the following ideas (see also Chap. 6 in [13]):

- Incompressible words cannot be produced by short programs. Hence, if one has an incompressible (infinite) word, it cannot be algorithmically obtained.
- Most words are incompressible. Hence, a word taken at random can usually be considered to be incompressible without loss of generality.
- If one “proves” a recursive property on incompressible words, then, by Proposition 3, we have a contradiction.

Application to Cellular Automata

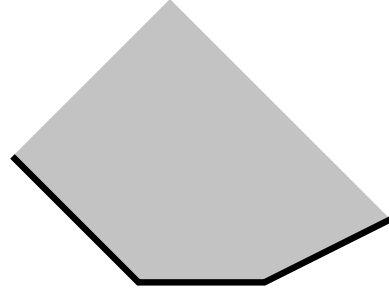
A portion of a space-time diagram of a CA can be computed given its local rule and the initial condition. The part of the diagram that depends only upon it has at most the complexity of this portion (Fig. 1). This fact often implies great computational dependencies between the initial part and the final part of the portion of the diagram: if the final part has high complexity, then the initial part must be at least as complex. Using this basic idea, proofs are structured as follows: assume that the final part has high complexity; use the hypothesis to prove that the initial part is not complex; then we have a contradiction. This technique provides a faster and clearer way to prove results that could be obtained by technical combinatorial arguments.

The first example illustrates this fact by rewriting a combinatorial proof in terms of Kolmogorov complexity. The second one is a result that was directly written in terms of Kolmogorov complexity.

Example 1

Consider the following result about languages recognizable by CA.

Proposition 4 ([17]) *The language $L = \{uvu, u, v \in \{0, 1\}^*, |u| > 1\}$ is not recognizable by a real-time one-way CA.*



Algorithmic Complexity and Cellular Automata, Figure 1

States in the *gray zone* can be computed for the states of the *black line*: $K(\text{gray zone}) \leq K(\text{black line})$ up to an additive constant

Before giving the proof in terms of Kolmogorov complexity, we must recall some concepts. A language L is accepted in real-time by a CA if it is accepted after $|x|$ transitions. An input word is accepted by a CA if after accepting state. A CA is *one-way* if the neighborhood is $\{0, 1\}$ (i. e., the central cell and the one on its right).

Proof One-way CA have some interesting properties. First, from a one-way CA recognizing L , a computer can build a one-way CA recognizing $L_\Sigma = \{uvu, u, v \in \Sigma^*, |u| > 1\}$, where Σ is any finite alphabet. The idea is to code the i th letter of Σ by $1u_01u_1 \dots 1u_k00$, where $u_0u_1 \dots u_k$ is i written in binary, and to apply the former CA.

Second, for any one-way CA of global rule f we have that

$$f^{|w|}(xwy) = f^{|w|}(xw)f^{|w|}(wy)$$

for all words x, y , and w .

Now, assume that a one-way CA recognizes L in real time. Then, there is an algorithm F that, for any integer n , computes a local rule for a one-way CA that recognizes $L_{\{0, \dots, n-1\}}$ in real time.

Fix an integer n and choose a 0-incompressible word w of length $n(n-1)$. Let A be the subset of Z of pairs (x, y) for $x, y \in \{0, \dots, n-1\}$ and let $x \neq y$ be defined by

$$A = \{(x, y), \text{ the } nx + y\text{th bit of } w \text{ is } 1\}.$$

Since A can be computed from w and vice versa, one finds that $K(A) = K(w) = |w| = n(n-1)$ up to an additive constant that does not depend on n .

Order the set $A = \{(x_0, y_0), (x_1, y_1), \dots\}$ and build a new word u as follows:

$$\begin{aligned} u_0 &= y_0x_0, \\ u_{i+1} &= u_iy_{i+1}x_{i+1}u_i, \\ u &= u_{|A|-1}, \end{aligned}$$

From Lemma 1 in [17], one finds that for all $x, y \in \{0, \dots, n-1\}$, $(x, y) \in A \Leftrightarrow xuy \in L$. Let δ be the local rule produced by $F(n)$, Q the set of final states, and f the associated global rule. Since

$$f^{|u|}(xuy) = f^{|u|}(xu)f^{|u|}(uy),$$

one has that

$$(x, y) \in A \Leftrightarrow xuy \in L \Leftrightarrow \delta(f^{|u|}(xu), f^{|u|}(uy)) \in Q.$$

Hence, from the knowledge of each $f^{|u|}(xu)$ and $f^{|u|}(ux)$ for $x \in \{0, \dots, n-1\}$, a list of $2n$ integers of $\{0, \dots, n-1\}$, one can compute A . We conclude that

$$K(A) \leq 2n \log_2(n)$$

up to an additive constant that does not depend on n . This contradicts $K(A) = n(n-1)$ for large enough n . \square

Example 2

Notation Given $x \in A^{\mathbb{Z}^D}$, we denote by $x_{\rightarrow n}$ the word $w \in S^{(2n+1)^D}$ obtained by taking all the states x_i for $i \in (2n+1)^D$ in the martial order.

The next example shows a proof that directly uses the incompressibility method.

Theorem 6 *In the Besicovitch topological space there is no transitive CA.*

Recall that a CA f is *transitive* if for all nonempty sets A, B there exists $n \in \mathbb{N}$ such that $f^n(A) \cap B \neq \emptyset$.

Proof By contradiction, assume that there exists a transitive CA f of radius r with $C = |S|$ states. Let x and y be two configurations such that

$$\forall n \in \mathbb{N}, K(x_{\rightarrow n} | y_{\rightarrow n}) \geq \frac{n}{2}.$$

A simple counting argument proves that such configurations x and y always exist. Since f is transitive, there are two configurations x' and y' such that for all $n \in \mathbb{N}$

$$\begin{aligned} \Delta(x_{\rightarrow n}, x'_{\rightarrow n}) &\leq 4\varepsilon n, \\ \Delta(y_{\rightarrow n}, y'_{\rightarrow n}) &\leq 4\delta n, \end{aligned} \quad (5)$$

and an integer u (which only depends on ε and δ) such that

$$f^u(y') = x', \quad (6)$$

where $\varepsilon = \delta = (4e^{10 \log_2 C})^{-1}$.

In what follows only n varies, while $C, u, x, y, x', y', \delta$, and ε are fixed and independent of n .

By Eq. (6), one may compute the word $x'_{\rightarrow n}$ from the following items:

- $y'_{\rightarrow n}, f, u$ and n ;
- The two words of y' of length ur that surround $y'_{\rightarrow n}$ and that are missing to compute $x'_{\rightarrow n}$ with Eq. (6).

We obtain that

$$K(x'_{\rightarrow n} | y'_{\rightarrow n}) \leq 2ur + K(u) + K(n) + K(f) + O(1) \leq o(n) \quad (7)$$

(the notations O and o are defined with respect to n). Note that n is fixed and hence $K(n)$ is a constant bounded by $\log_2 n$. Similarly, r and S are fixed, and hence $K(f)$ is constant and bounded by $C^{2r+1} \log_2 C + O(1)$.

Let us evaluate $K(y'_{\rightarrow n} | y_{\rightarrow n})$. Let $a_1, a_2, a_3, \dots, a_k$ be the positive positions that $y_{\rightarrow n}$ and $y'_{\rightarrow n}$ differ at, sorted in increasing order. Let $b_1 = a_1$ and $b_i = a_i - a_{i-1}$, for $2 \leq i \leq k$. By Eq. (5) we know that $k \leq 4\delta n$. Note that $\sum_{i=1}^k b_i = a_k \leq n$.

Symmetrically let $a'_1, a'_2, a'_3, \dots, a'_{k'}$ be the absolute values of the strictly negative positions that $y_{\rightarrow n}$ and $y'_{\rightarrow n}$ differ at, sorted in increasing order. Let $b'_1 = a'_1$ and $b'_i = a'_i - a'_{i-1}$, where $2 \leq i \leq k'$. Equation (5) states that $k' \leq 4\delta n$.

Since the logarithm is a concave function, one has

$$\sum \frac{\ln b_i}{k} \leq \ln \frac{\sum b_i}{k} \leq \ln \frac{n}{k},$$

and hence

$$\sum \ln b_i \leq k \ln \frac{n}{k}, \quad (8)$$

which also holds for b'_i and k' .

Knowledge of the b_i, b'_i , and $k + k'$ states of the cells of $y'_{\rightarrow n}$, where $y_{\rightarrow n}$ differs from $y'_{\rightarrow n}$, is enough to compute $y'_{\rightarrow n}$ from $y_{\rightarrow n}$. Hence,

$$\begin{aligned} K(y'_{\rightarrow n} | y_{\rightarrow n}) &\leq \sum \ln(b_i) \\ &\quad + \sum \ln(b'_i) + (k + k') \log_2 C + O(1). \end{aligned}$$

Equation (8) states that

$$K(y'_{\rightarrow n} | y_{\rightarrow n}) \leq k \ln \frac{n}{k} + k' \ln \frac{n}{k'} + (k + k') \log_2 C + O(1).$$

The function $k \mapsto k \ln \frac{n}{k}$ is increasing on $[0, \frac{n}{e}]$. As $k \leq 4\delta n \leq \frac{n}{e^{10 \log_2 C}}$, we have that

$$k \ln \frac{n}{k} \leq 4\delta n \ln \frac{n}{4\delta n} \leq \frac{n}{e^{10 \log_2 C}} \ln e^{10 \log_2 C} \leq \frac{10n}{e^{10 \log_2 C}}$$

and that

$$(k + k') \log_2 C \leq \frac{2n \log_2 C}{e^{10 \log_2 C}}.$$

Replacing a , b , and k by a' , b' , and k' , the same sequence of inequalities leads to a similar result. One deduces that

$$K(y'_{\rightarrow n} | y_{\rightarrow n}) \leq \frac{(2 \log_2 C + 20)n}{e^{10 \log_2 C}} + O(1). \quad (9)$$

Similarly, Eq. (9) is also true with $K(x_{\rightarrow n} | x'_{\rightarrow n})$.

The triangular mequality for Kolmogorov complexity $K(a|b) \leq K(a|c) + K(b|c)$ (consequence of theorems 3 and 4) gives:

$$K(x_{\rightarrow n} | y_{\rightarrow n}) \leq K(x_{\rightarrow n} | x'_{\rightarrow n}) + K(x'_{\rightarrow n} | y'_{\rightarrow n}) + K(y'_{\rightarrow n} | y_{\rightarrow n}) + O(1).$$

Equations (9) and (7) allow one to conclude that

$$K(x_{\rightarrow n} | y_{\rightarrow n}) \leq \frac{(2 \log_2 C + 20)n}{e^{10 \log_2 C}} + o(n).$$

The hypothesis on x and y was $K(x_{\rightarrow n} | y_{\rightarrow n}) \geq \frac{n}{2}$. This implies that

$$\frac{n}{2} \leq \frac{(2C + 20)n}{e^{10 \log_2 C}} + o(n).$$

The last inequality is false for big enough n . \square

Measuring CA Structural Complexity

Another use of Kolmogorov complexity in the study of CA is to understand what maximum complexity they can produce extracting examples of CA that show high complexity characteristics. The question is to define what is the meaning of “show high complexity characteristics” and, more precisely, what characteristic to consider. This section is devoted to structural characteristics of CA, that is to say, complexity that can be observed through static particularities.

The Case of Tilings

In this section, we give the original example that was given for tilings, often considered as a static version of CA.

In [10], Durand et al. construct a tile set whose tilings have maximum complexity. This paper contains two main results. The first one is an upper bound for the complexity of tilings, and the second one is an example of tiling that reaches this bound. First we recall the definitions about tilings of a plane by Wang tiles.

Definition 6 (Tilings with Wang tiles) Let C be a finite set of colors. A Wang tile is a quadruplet (n, s, e, w) of four colors from C corresponding to a square tile whose top color is n , left color is w , right color is e , and bottom color

is s . A Wang tile cannot be rotated but can be copied to infinity.

Given a set of Wang tiles T , we say that a plane can be tiled by T if one can place tiles from T on the square grid \mathbb{Z}^2 such that the adjacent border of neighboring tiles has the same color. A set of tiles T that can tile the plane is called a *palette*.

The notion of *local constraint* gives a point of view closer to CA than tilings. Roughly speaking, it gives the local constraints that a tiling of the plane using 0 and 1 must satisfy. Note that this notion can be defined on any alphabet, but we can equivalently code any letter with 0 and 1.

Definition 7 (Tilings by local constraints) Let r be a positive integer called *radius*. Let C be a set of square patterns of size $2r + 1$ made of 0 and 1 (formally a function from $\llbracket -r, r \rrbracket$ to $\{0, 1\}$). The set is said to tile the plane if there is a way to put zeros and ones on the 2-D grid (formally a function from \mathbb{Z}^2 to $\{0, 1\}$) whose patterns of size $2r + 1$ are all in C . The possible layouts of zeros and ones on the (2-D) grid are called the *tilings acceptable for C*.

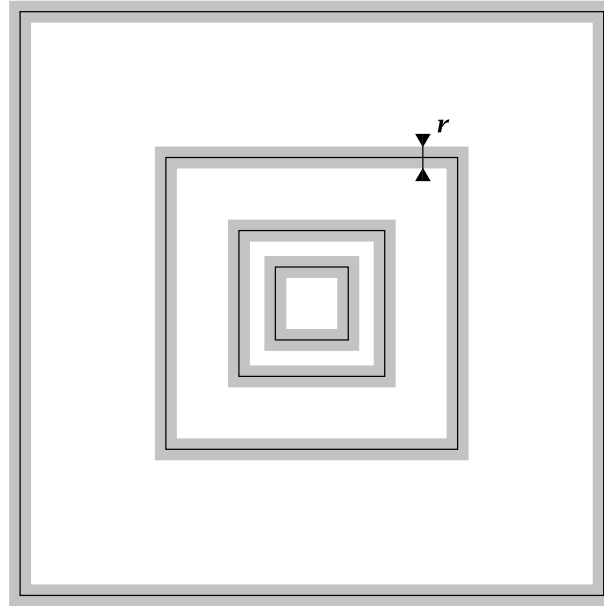
Seminal papers on this subject used Wang tiles. We translate these results in terms of local constraints in order to more smoothly apply them to CA.

Theorem 7 proves that, among tilings acceptable by a local constraint, there is always one that is not too complex. Note that this is a good notion since the use of a constraint of radius 1, $\{\llbracket 0, 1 \rrbracket\}$, which allows for all possible patterns, provides for acceptable high-complexity tilings since all tilings are acceptable.

Theorem 7 ([10]) Let C be a local constraint. There is tiling τ acceptable for C such that the Kolmogorov complexity of the central pattern of size n is $O(n)$ (recall that the maximal complexity for a square pattern $n \times n$ is $O(n^2)$).

Proof The idea is simple: if one knows the bits present in a border of width r of a pattern of size n , there are finitely many possibilities to fill the interior, so the first one in any computable order (for instance, lexicographically when putting all horizontal lines one after the other) has at most the complexity of the border since an algorithm can enumerate all possible fillings and take the first one in the chosen order.

Then, if one knows the bits in borders of width r of all central square patterns of size 2^n for all positive integers n (Fig. 2), one can recursively compute for each gap the first possible filling for a given computable order. The tiling obtained in this way (this actually defines a tiling since all cells are eventually assigned a value) has the required complexity: in order to compute the central pattern of size n ,



Algorithmic Complexity and Cellular Automata, Figure 2
Nested squares of size 2^k and border width r

the algorithm simply needs all the borders of size 2^k for $k \leq 2n$, which is of length at most $O(n)$. \square

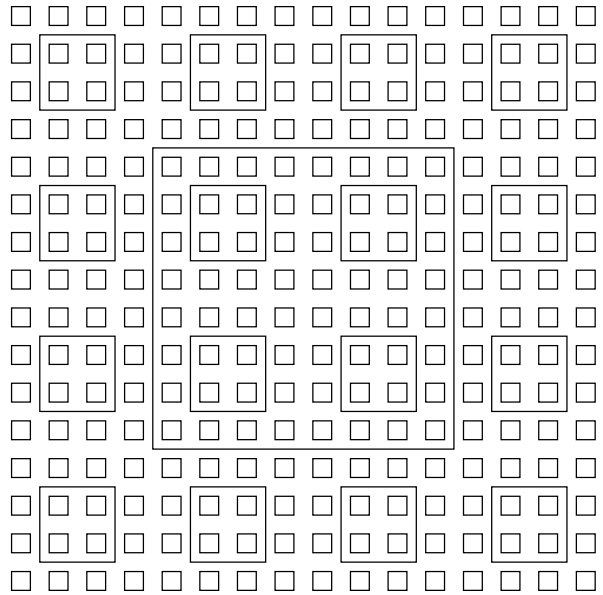
The next result proves that this bound is almost reachable.

Theorem 8 *Let r be any computable monotone and unbounded function. There exists a local constraint C_r such that for all tilings τ acceptable for C_r , the complexity of the central square pattern of τ is $O(n/r(n))$.*

The original statement does not have the O part. However, note that the simulation of Wang tiles by a local constraint uses a square of size $\ell = \lceil \sqrt{\log k} \rceil$ to simulate a single tile from a set of k Wang tiles; a square pattern of size n in the tiling corresponds to a square pattern of size $\frac{n}{\ell}$ in the original tiling.

Function r can grow very slowly (for instance, the inverse of the Ackermann function) provided it grows monotonously to infinity and is computable.

Proof The proof is rather technical, and we only give a sketch. The basic idea consists in taking a tiling constructed by Robinson [16] in order to prove that it is undecidable to test if a local constraint can tile the plane. This tiling is self-similar and is represented in Fig. 3. As it contains increasingly larger squares that occur periodically (note that the whole tiling is not periodic but only quasiperiodic), one can perform more and more computation steps within these squares (the periodicity is required to be sure that squares are present).



Algorithmic Complexity and Cellular Automata, Figure 3
Robinson's tiling

Using this tiling, Robinson can build a local constraint that simulates any Turing machine. Note that in the present case, the situation is more tricky than it seems since some technical features must be assured, like the fact that a square must deal with the smaller squares inside it or the constraint to add to make sure that smaller squares

have the same input as the bigger one. Using the constraint to forbid the occurrence of any final state, one gets that the compatible tilings will only simulate computations for inputs for which it does not halt.

To finish the proof, Durand et al. build a local constraint C that simulates a special Turing machine that halts on inputs whose complexity is small. Such a Turing machine exists since, though Kolmogorov complexity is not computable, testing all programs from ε to 1^n allows a Turing machine to compute all words whose complexity is below n and halt if it finds one. Then all tilings compatible with C contain in each square an input on which the Turing machine does not halt yet is of high complexity.

Function r occurs in the technical arguments since computable zones do not grow as fast as the length of squares. \square

The Case of Cellular Automata

As we have seen so far, one of the results of [10] is that a tile set always produces tilings whose central square patterns of size n have a Kolmogorov complexity of $O(n)$ and not n^2 (which is the maximal complexity).

In the case of CA, something similar holds for space-time diagrams. Indeed, if one knows the initial row. One can compute the triangular part of the space-time diagram which depends on it (see Fig. 1). Then, as with tilings, the complexity of an $n \times n$ -square is the same as the complexity of the first line, i. e., $O(n)$. However, unlike tilings, in CA, there is no restriction as to the initial configuration, hence CA have simple space-time diagrams.

Thus in this case Kolmogorov complexity is not of great help. One idea to improve the results could be to study how the complexity of configurations evolves during the application of the global rule. This aspect is particularly interesting with respect to dynamical properties. This is the subject of the next section.

Consider a CA F with radius r and local rule δ . Then, its orbit limit set cannot be empty. Indeed, let a be a state of f . Consider the configuration ${}^\omega a^\omega$. Let $s_a = \delta(a, \dots, a)$. Then, $f({}^\omega a^\omega) = {}^\omega s_a^\omega$. Consider now a graph whose vertices are the states of the CA and the edges are (a, s_a) . Since each vertex has an outgoing edge (actually exactly one), it must contain a cycle $a_0 \rightarrow a_1 \rightarrow \dots \rightarrow a_k \rightarrow a_0$. Then each of the configurations ${}^\omega a_i^\omega$ for $0 \leq i \leq k$ is in the limit set since $f^k({}^\omega a_i^\omega) = {}^\omega a_i^\omega$.

This simple fact proves that any orbit limit set (and any limit set) of a CA must contain at least a monochromatic configuration whose complexity is low (for any reasonable definition). However, one can build a CA whose orbit limit set contains only complex configurations, except

for the mandatory monochromatic one using the local-constraints technique discussed in the previous section.

Proposition 5 *There exists a CA whose orbit limit set contains only complex configurations.*

Proof Let r be any computable monotone and unbounded function and C_r the associated local constraint. Let A be the alphabet that C_r is defined on. Consider the 2-D CA f_r on the alphabet $A \cup \{\#\}$ (we assume $\{\#\} \neq A$). Let r be the radius of f_r (the same radius as C_r). Finally, the local rule δ of f_r is defined as follows:

$$\delta(P) = \begin{cases} P_0 & \text{if } P \in C_r, \\ \# & \text{otherwise.} \end{cases}$$

Using this local rule, one can verify the following fact. If the configuration c is not acceptable for C_r , then $(\mathcal{O}(c))' = \{\omega \#^\omega\}$; $(\mathcal{O}(c))' = \{c\}$ otherwise. Indeed, if c is acceptable for C_r , then $f(c) = c$; otherwise there is a position i such that $c(i)$ is not a valid pattern for C_r . Then $f(c)(i) = \#$. By simple induction, this means that all cells that are at a distance less than kr from position i become $\#$ after k steps. Hence, for all $n > 0$, after $k \geq \frac{n+2|i|}{r}$ steps, the Cantor distance between $f^k(c)$ and $\omega \#^\omega$ is less than 2^{-n} , i. e., $\mathcal{O}(c)$ tends to $\omega \#^\omega$. \square

Measuring the Complexity of CA Dynamics

The results of the previous section have limited range since they tell something about the quasicomplexity but nothing about the plain complexity of the limit set. To enhance our study we need to introduce some general concepts, namely, randomness spaces.

Randomness Spaces

Roughly speaking, a randomness space is a structure made of a topological space and a measure that helps in defining which points of the space are random. More formally we can give the following.

Definition 8 ([6]) A *randomness space* is a structure $\langle X, B, \mu \rangle$ where X is a topological space, $B: \mathbb{N} \rightarrow 2^X$ a total numbering of a subbase of X , and μ a Borel measure.

Given a numbering B for a subbase for an open set of X , one can produce a numbering B' for a base as follows:

$$B'(i) = \bigcap_{j \in D_{i+1}} B(j),$$

where $D: \mathbb{N} \rightarrow \{E \mid E \subseteq \mathbb{N} \text{ and } E \text{ is finite}\}$ is the bijection defined by $D^{-1}(E) = \sum_{i \in E} 2^i$. B' is called the *base derived* from the subbase B . Given two sequences of open sets (V_n)

and (U_n) of X , we say that (V_n) is *U-computable* if there exists a recursively enumerable set $H \in \mathbb{N}$ such that

$$\forall n \in \mathbb{N}, V_n = \bigcup_{i \in \mathbb{N}, \langle n, i \rangle \in H} U_i,$$

where $\langle i, j \rangle = (i + j)(i + j + 1) + j$ is the classical bijection between \mathbb{N}^2 and \mathbb{N} . Note that this bijection can be extended to \mathbb{N}^D (for $D > 1$) as follows: $\langle x_1, x_2, \dots, x_k \rangle = \langle x_1, \langle x_2, \dots, x_k \rangle \rangle$.

Definition 9 ([6]) Given a randomness space $\langle X, B, \mu \rangle$, a *randomness test* on X is a B' -computable sequence (U_n) of open sets such that $\forall n \in \mathbb{N}, \mu(U_n) \leq 2^{-n}$.

Given a randomness test (U_n) , a point $x \in X$ is said to *pass the test* (U_n) if $x \in \bigcap_{n \in \mathbb{N}} U_n$. In other words, tests select points belonging to sets of null measure. The computability of the tests assures the computability of the selected null measure sets.

Definition 10 ([6]) Given a randomness space $\langle X, B, \mu \rangle$, a point $x \in X$ is *nonrandom* if x passes some randomness test. The point $x \in X$ is *random* if it is not nonrandom.

Finally, note that for any $D \geq 1$, $\langle A^{\mathbb{Z}^D}, B, \mu \rangle$ is a randomness space when setting B as

$$B(j + D \cdot \langle i_1, \dots, i_D \rangle) = \left\{ c \in A^{\mathbb{Z}^D}, c_{i_1, \dots, i_D} = a_j \right\},$$

where $A = \{a_1, \dots, a_j, \dots, a_D\}$ and μ is the classical product measure built from the uniform Bernoulli measure over A .

Theorem 9 ([6]) Consider a D -dimensional CA f . Then, the following statements are equivalent:

1. f is surjective;
2. $\forall c \in A^{\mathbb{Z}^D}$, if c is rich (i.e., c contains all possible finite patterns), then $f(c)$ is rich;
3. $\forall c \in A^{\mathbb{Z}^D}$, if c is random, then $f(c)$ is random.

Theorem 10 ([6]) Consider a D -dimensional CA f . Then, $\forall c \in A^{\mathbb{Z}^D}$, if c is not rich, then $f(c)$ is not rich.

Theorem 11 ([6]) Consider a 1-D CA f . Then, $\forall c \in A^{\mathbb{Z}^D}$, if c is nonrandom, then $f(c)$ is nonrandom.

Note that the result in Theorem 11 is proved only for 1-D CA; its generalization to higher dimensions is still an open problem.

Open problem 1 Do D -dimensional CA for $D > 1$ preserve nonrandomness?

From Theorem 9 we know that the property of preserving randomness (resp. richness) is related to surjectivity.

Hence, randomness (resp. richness) preserving is decidable in one dimension and undecidable in higher dimensions. The opposite relations are still open.

Open problem 2 Is nonrichness (resp. nonrandomness) a decidable property?

Algorithmic Distance

In this section we review an approach to the study of CA dynamics from the point of view of algorithmic complexity that is completely different than the one reported in the previous section. For more details on the algorithmic distance see [► Chaotic Behavior of Cellular Automata](#).

In this new approach we are going to define a new distance using Kolmogorov complexity in such a way that two points x and y are near if it is “easy” to transform x into y or vice versa using a computer program. In this way, if a CA turns out to be sensitive to initial conditions, for example, then it means that it is able to create new information. Indeed, we will see that this is not the case.

Definition 11 The *algorithmic distance* between $x \in A^{\mathbb{Z}^D}$ and $y \in A^{\mathbb{Z}^D}$ is defined as follows:

$$d_a(x, y) = \limsup_{n \rightarrow \infty} \frac{K(x \rightarrow n | y \rightarrow n) + K(y \rightarrow n | x \rightarrow n)}{2(2n + 1)^D}.$$

It is not difficult to see that d_a is only a pseudodistance since there are many points that are at null distance (those that differ only on a finite number of cells, for example). Consider the relation \cong of “being at null d_a distance, i.e., $\forall x, y \in A^{\mathbb{Z}^D}, x \cong y$ if and only if $d_a(x, y) = 0$. Then, $\langle A^{\mathbb{Z}^D} / \cong, d_a \rangle$ is a metric space.

Note that the definition of d_a does not depend on the chosen additively optimal universal description mode since the additive constant disappears when dividing by $2(2n + 1)^D$ and taking the superior limit. Moreover, by Theorem 2, the distance is bounded by 1.

The following results summarize the main properties of this new metric space.

Theorem 12 (► Chaotic Behavior of Cellular Automata) The metric space $\langle A^{\mathbb{Z}^D} / \cong, d_a \rangle$ is perfect, pathwise connected, infinite dimensional, nonseparable, and noncompact.

Theorem 12 says that the new topological space has enough interesting properties to make worthwhile the study of CA dynamics on it.

The first interesting result obtained by this approach concerns surjective CA. Recall that surjectivity plays a central role in the study of chaotic behavior since it is a nec-

essary condition for many other properties used to define deterministic chaos like expansivity, transitivity, ergodicity, and so on (see ► [Topological Dynamics of Cellular Automata](#) and [8] for more on this subject). The following result proves that in the new topology $A^{\mathbb{Z}^D}/\cong$ the situation is completely different.

Proposition 6 (► Chaotic Behavior of Cellular Automata) *If f is a surjective CA, then $d_A(x, f(x)) = 0$ for any $x \in A^{\mathbb{Z}^D}/\cong$. In other words, every CA behaves like the identity in $A^{\mathbb{Z}^D}/\cong$.*

Proof In order to compute $f(x) \rightarrow_n$ from $x \rightarrow_n$, one need only know the index of f in the set of CA with radius r , state set S , and dimension d ; therefore

$$K(f(x) \rightarrow_n | x \rightarrow_n) \leq 2dr(2n + 2r + 1)^{d-1} \log_2 |S| \\ + K(f) + 2 \log_2 K(f) + c,$$

and similarly

$$K(x \rightarrow_n | f(x) \rightarrow_n) \leq gdr(2n + 1)^{d-1} \log_2 |S| \\ + K(f) + 2 \log_2 K(f) + c.$$

Dividing by $2(2n + 1)^d$ and taking the superior limit one finds that $d_a(x, f(x)) = 0$. \square

The result of Proposition 6 means that surjective CA can neither create new information nor destroy it. Hence, they have a high degree of stability from an algorithmic point of view. This contrasts with what happens in the Cantor topology. We conclude that the classical notion of deterministic chaos is orthogonal to “algorithmic chaos,” at least in what concerns the class of CA.

Proposition 7 (► Chaotic Behavior of Cellular Automata) *Consider a CA f that is neither surjective nor constant. Then, there exist two configurations $x, y \in A^{\mathbb{Z}^D}/\cong$ such that $d_a(x, y) = 0$ but $d_a(f(x), f(y)) \neq 0$.*

In other words, Proposition 7 says that nonsurjective, non-constant CA are not compatible with \cong and hence not continuous. This means that, for any pair of configurations x, y , this kind of CA either destroys completely the information content of x, y or preserves it in one, say x , and destroys it in y . However, the following result says that some weak form of continuity still persists (see ► [Topological Dynamics of Cellular Automata](#) and [8] for the definitions of equicontinuity point and sensibility to initial conditions).

Proposition 8 (► Chaotic Behavior of Cellular Automata) *Consider a CA f and let \underline{a} be the configuration made with all cells in state a . Then, \underline{a} is both a fixed point and an equicontinuity point for f .*

Even if CA were noncontinuous on $A^{\mathbb{Z}^D}/\cong$, one could still wonder what happens with respect to the usual properties used to define deterministic chaos. For instance, by Proposition 8, it is clear that no CA is sensitive to initial conditions. The following question is still open.

Open problem 3 *Is \underline{a} the only equicontinuity point for CA on $A^{\mathbb{Z}^D}/\cong$?*

Future Directions

In this paper we have illustrated how algorithmic complexity can help in the study of CA dynamics. We essentially used it as a powerful tool to decrease the combinatorial complexity of problems. These kinds of applications are only at their beginnings and much more are expected in the future. For example, in view of the results of Subsect. “[Example 1](#)”, we wonder if Kolmogorov complexity can help in proving the famous conjecture that languages recognizable in real time by CA are a strict subclass of linear-time recognizable languages (see [9, 14]).

Another completely different development would consist in finding how and if Theorem 11 extends to higher dimensions. How this property can be restated in the context of the algorithmic distance is also of great interest.

Finally, how to extend the results obtained for CA to other dynamical systems is a research direction that must be explored. We are rather confident that this can shed new light on the complexity behavior of such systems.

Acknowledgments

This work has been partially supported by the ANR Blanc Project “Sycamore”.

Bibliography

Primary Literature

1. Blanchard F, Formenti E, Kurka P (1999) Cellular automata in the Cantor, Besicovitch and Weyl topological spaces. *Complex Syst* 11:107–123
2. Blanchard F, Cervelle J, Formenti E (2003) Periodicity and transitivity for cellular automata in Besicovitch topologies. In: Rován B, Vojtas P (eds) (MFCS’2003), vol 2747. Springer, Bratislava, pp 228–238
3. Blanchard F, Cervelle J, Formenti E (2005) Some results about chaotic behavior of cellular automata. *Theor Comput Sci* 349(3):318–336
4. Brudno AA (1978) The complexity of the trajectories of a dynamical system. *Russ Math Surv* 33(1):197–198
5. Brudno AA (1983) Entropy and the complexity of the trajectories of a dynamical system. *Trans Moscow Math Soc* 44:127
6. Calude CS, Hertling P, Jürgensen H, Weihrauch K (2001) Randomness on full shift spaces. *Chaos, Solitons Fractals* 12(3):491–503

7. Cattaneo G, Formenti E, Margara L, Mazoyer J (1997) A shift-invariant metric on S^2 inducing a non-trivial topology. In: Privara I, Rusika P (eds) (MFCS'97), vol 1295. Springer, Bratislava, pp 179–188
8. Cerveille J, Durand B, Formenti E (2001) Algorithmic information theory and cellular automata dynamics. In: Mathematical Foundations of Computer Science (MFCS'01). Lectures Notes in Computer Science, vol 2136. Springer, Berlin pp 248–259
9. Delorme M, Mazoyer J (1999) Cellular automata as languages recognizers. In: Cellular automata: A parallel model. Kluwer, Dordrecht
10. Durand B, Levin L, Shen A (2001) Complex tilings. In: STOC '01: Proceedings of the 33rd annual ACM symposium on theory of computing, pp 732–739
11. Kari J (1994) Rice's theorem for the limit set of cellular automata. Theor Comput Sci 127(2):229–254
12. Kůrka P (1997) Languages, equicontinuity and attractors in cellular automata. Ergod Theory Dyn Syst 17:417–433
13. Li M, Vitányi P (1997) An introduction to Kolmogorov complexity and its applications, 2nd edn. Springer, Berlin
14. M Delorme EF, Mazoyer J (2000) Open problems. Research Report LIP 2000-25, Ecole Normale Supérieure de Lyon
15. Pivato M (2005) Cellular automata vs. quasisturmian systems. Ergod Theory Dyn Syst 25(5):1583–1632
16. Robinson RM (1971) Undecidability and nonperiodicity for tilings of the plane. Invent Math 12(3):177–209
17. Terrier V (1996) Language not recognizable in real time by one-way cellular automata. Theor Comput Sci 156:283–287
18. Wolfram S (2002) A new kind of science. Wolfram Media, Champaign. <http://www.wolframscience.com/>

Books and Reviews

- Batterman RW, White HS (1996) Chaos and algorithmic complexity. Fund Phys 26(3):307–336
- Bennet CH, Gács P, Li M, Vitányi P, Zurek W (1998) Information distance. IEEE Trans Inf Theory 44(4):1407–1423
- Calude CS (2002) Information and randomness. Texts in theoretical computer science, 2nd edn. Springer, Berlin
- Cover TM, Thomas JA (2006) Elements of information theory, 2nd edn. Wiley, New York
- White HS (1993) Algorithmic complexity of points in dynamical systems. Ergod Theory Dyn Syst 13:807–830

Amorphous Computing

HAL ABELSON, JACOB BEAL, GERALD JAY SUSSMAN
Computer Science and Artificial Intelligence Laboratory,
Massachusetts Institute of Technology, Cambridge, USA

Article Outline

Glossary

Definition of the Subject

Introduction

The Amorphous Computing Model
Programming Amorphous Systems

Amorphous Computing Paradigms
Primitives for Amorphous Computing
Means of Combination and Abstraction
Supporting Infrastructure and Services
Lessons for Engineering
Future Directions
Bibliography

Glossary

Amorphous computer A collection of computational particles dispersed irregularly on a surface or throughout a volume, where individual particles have no *a priori* knowledge of their positions or orientations.

Computational particle A (possibly faulty) individual device for an amorphous computer. Each particle has modest computing power and a modest amount of memory. The particles are not synchronized, although they are all capable of operating at similar speeds, since they are fabricated by the same process. All particles are programmed identically, although each particle has means for storing local state and for generating random numbers.

Field A function assigning a value to every particle in an amorphous computer.

Gradient A basic amorphous computing primitive that estimates the distance from each particle to the nearest particle designated as a source of the gradient.

Definition of the Subject

The goal of amorphous computing is to identify organizational principles and create programming technologies for obtaining intentional, pre-specified behavior from the cooperation of myriad unreliable parts that are arranged in unknown, irregular, and time-varying ways. The heightened relevance of amorphous computing today stems from the emergence of new technologies that could serve as substrates for information processing systems of immense power at unprecedentedly low cost, if only we could master the challenge of programming them.

Introduction

Even as the foundations of computer science were being laid, researchers could hardly help noticing the contrast between the robustness of natural organisms and the fragility of the new computing devices. As John von Neumann remarked in 1948 [53]:

With our artificial automata we are moving much more in the dark than nature appears to be with

its organisms. We are, and apparently, at least at present, have to be much more ‘scared’ by the occurrence of an isolated error and by the malfunction which must be behind it. Our behavior is clearly that of overcaution, generated by ignorance.

Amorphous computing emerged as a field in the mid-1990s, from the convergence of three factors:

- Inspiration from the cellular automata models for fundamental physics [13,34].
- Hope that understanding the robustness of biological development could both help overcome the brittleness typical of computer systems and also illuminate the mechanisms of developmental biology.
- The prospect of nearly free computers in vast quantities.

Microfabrication

One technology that has come to fruition over the past decade is micro-mechanical electronic component manufacture, which integrates logic circuits, micro-sensors, actuators, and communication on a single chip. Aggregates of these can be manufactured extremely inexpensively, provided that not all the chips need work correctly, and that there is no need to arrange the chips into precise geometrical configurations or to establish precise interconnections among them. A decade ago, researchers envisioned *smart dust* elements small enough to be borne on air currents to form clouds of communicating sensor particles [26].

Airborne sensor clouds are still a dream, but networks of millimeter-scale particles are now commercially available for environmental monitoring applications [40]. With low enough manufacturing costs, we could mix such particles into bulk materials to form coatings like “smart paint” that can sense data and communicate its actions to the outside world. A smart paint coating on a wall could sense vibrations, monitor the premises for intruders, or cancel noise. Bridges or buildings coated with smart paint could report on traffic and wind loads and monitor structural integrity. If the particles have actuators, then the paint could even heal small cracks by shifting the material around. Making the particles mobile opens up entire new classes of applications that are beginning to be explored by research in *swarm robotics* [35] and *modular robotics* [49].

Cellular Engineering

The second disruptive technology that motivates the study of amorphous computing is microbiology. Biological or-

ganisms have served as motivating metaphors for computing since the days of calculating engines, but it is only over the past decade that we have begun to see how biology could literally be a substrate for computing, through the possibility of constructing digital-logic circuits within individual living cells. In one technology logic signals are represented not by electrical voltages and currents, but by concentrations of DNA binding proteins, and logic elements are realized as binding sites where proteins interact through promotion and repression. As a simple example, if A and B are proteins whose concentrations represent logic levels, then an “inverter” can be implemented in DNA as a genetic unit through which A serves as a repressor that blocks the production of B [29,54,55,61]. Since cells can reproduce themselves and obtain energy from their environment, the resulting information processing units could be manufactured in bulk at a very low cost.

There is beginning to take shape a technology of cellular engineering that can tailor-make programmable cells to function as sensors or delivery vehicles for pharmaceuticals, or even as chemical factories for the assembly of nanoscale structures. Researchers in this emerging field of *synthetic biology* are starting to assemble registries of standard logical components implemented in DNA that can be inserted into *E. coli* [48]. The components have been engineered to permit standard means of combining them, so that biological logic designers can assemble circuits in a mix-and-match way, similar to how electrical logic designers create circuits from standard TTL parts. There’s even an *International Genetically Engineered Machine Competition* where student teams from universities around the world compete to create novel biological devices from parts in the registry [24].

Either of these technologies—microfabricated particles or engineered cells—provides a path to cheaply fabricate aggregates of massive numbers of computing elements. But harnessing these for computing is quite a different matter, because the aggregates are unstructured. Digital computers have always been constructed to behave as precise arrangements of reliable parts, and almost all techniques for organizing computations depend upon this precision and reliability. Amorphous computing seeks to discover new programming methods that do not require precise control over the interaction or arrangement of the individual computing elements and to instantiate these techniques in new programming languages.

The Amorphous Computing Model

Amorphous computing models the salient features of an unstructured aggregate through the notion of an *amor-*

phous computer, a collection of *computational particles* dispersed irregularly on a surface or throughout a volume, where individual particles have no *a priori* knowledge of their positions or orientations. The particles are possibly faulty, may contain sensors and effect actions, and in some applications might be mobile. Each particle has modest computing power and a modest amount of memory. The particles are not synchronized, although they are all capable of operating at similar speeds, since they are fabricated by the same process. All particles are programmed identically, although each particle has means for storing local state and for generating random numbers. There may also be several distinguished particles that have been initialized to particular states.

Each particle can communicate with a few nearby neighbors. In an electronic amorphous computer the particles might communicate via short-distance radio, whereas bioengineered cells might communicate by chemical signals. Although the details of the communication model can vary, the maximum distance over which two particles can communicate effectively is assumed to be small compared with the size of the entire amorphous computer. Communication is assumed to be unreliable and a sender has no assurance that a message has been received (Higher-level protocols with message acknowledgment can be built on such unreliable channels).

We assume that the number of particles may be very large (on the order of 10^6 to 10^{12}). Algorithms appropriate to run on an amorphous computer should be relatively independent of the number of particles: the performance should degrade gracefully as the number of particles decreases. Thus, the entire amorphous computer can be regarded as a massively parallel computing system, and previous investigations into massively parallel computing, such as research in cellular automata, are one source of ideas for dealing with amorphous computers. However, amorphous computing differs from investigations into cellular automata, because amorphous mechanisms must be independent of the detailed configuration, reliability, and synchronization of the particles.

Programming Amorphous Systems

A central theme in amorphous computing is the search for programming paradigms that work within the amorphous model. Here, biology has been a rich source of metaphors for inspiring new programming techniques. In embryonic development, even though the precise arrangements and numbers of the individual cells are highly variable, the genetic “programs” coded in DNA nevertheless produce well-defined intricate shapes and precise forms.

Amorphous computers should be able to achieve similar results.

One technique for programming an amorphous computer uses diffusion. One particle (chosen by some symmetry-breaking process) broadcasts a message. This message is received by each of its neighbors, which propagate it to their neighbors, and so on, to create a wave that spreads throughout the system. The message contains a count, and each particle stores the received count and increments it before re-broadcasting. Once a particle has stored its count, it stops re-broadcasting and ignores future count messages. This *count-up wave* gives each particle a rough measure of its distance from the original source. One can also produce regions of controlled size, by having the count message relayed only if the count is below a designated bound.

Two such count-up waves can be combined to identify a chain of particles between two given particles *A* and *B*. Particle *A* begins by generating a count-up wave as above. This time, however, each intermediate particle, when it receives its count, performs a handshake to identify its “predecessor”—the particle from which it received the count (and whose own count will therefore be one less). When the wave of count messages reaches *B*, *B* sends a “successor” message, informing its predecessor that it should become part of the chain and should send a message to *its* predecessor, and so on, all the way back to *A*. Note that this method works, even though the particles are irregularly distributed, provided there is a path from *A* to *B*.

The motivating metaphor for these two programs is chemical gradient diffusion, which is a foundational mechanism in biological development [60]. In nature, biological mechanisms not only generate elaborate forms, they can also maintain forms and repair them. We can modify the above amorphous line-drawing program so that it produces self-repairing line: first, particles keep re-broadcasting their count and successor messages. Second, the status of a particle as having a count or being in the chain decays over time unless it is refreshed by new messages. That is, a particle that stops hearing successor messages intended for it will eventually revert to not being in the chain and will stop broadcasting its own successor messages. A particle that stops hearing its count being broadcast will start acting as if it never had a count, pick up a new count from the messages it hears, and start broadcasting the count messages with the new count. Clement and Nagpal [12] demonstrated that this mechanism can be used to generate self-repairing lines and other patterns, and even re-route lines and patterns around “dead” regions where particles have stopped functioning.

The relationship with biology flows in the other direction as well: the amorphous algorithm for repair is a model which is not obviously inconsistent with the facts of angiogenesis in the repair of wounds. Although the existence of the algorithm has no bearing on the facts of the matter, it may stimulate systems-level thinking about models in biological research. For example, Patel et al. use amorphous ideas to analyze the growth of epithelial cells [43].

Amorphous Computing Paradigms

Amorphous computing is still in its infancy. Most of linguistic investigations based on the amorphous computing model have been carried out in simulation. Nevertheless, this work has yielded a rich variety of programming paradigms that demonstrate that one can in fact achieve robustness in face of the unreliability of individual particles and the absence of precise organization among them.

Marker Propagation for Amorphous Particles

Weiss's Microbial Colony Language [55] is a marker propagation language for programming the particles in an amorphous computer. The program to be executed, which is the same for each particle, is constructed as a set of rules. The state of each particle includes a set of binary markers, and rules are enabled by boolean combinations of the markers. The rules, which have the form (*trigger, condition, action*) are triggered by the receipt of labelled messages from neighboring particles. A rule may test conditions, set or clear various markers, and it broadcast further messages to its neighbors. Each message carries a count that determines how far it will diffuse, and each marker has a lifetime that determines how long its value lasts. Supporting these language's rules is a runtime system that

automatically propagates messages and manages the lifetimes of markers, so that the programmer need not deal with these operations explicitly.

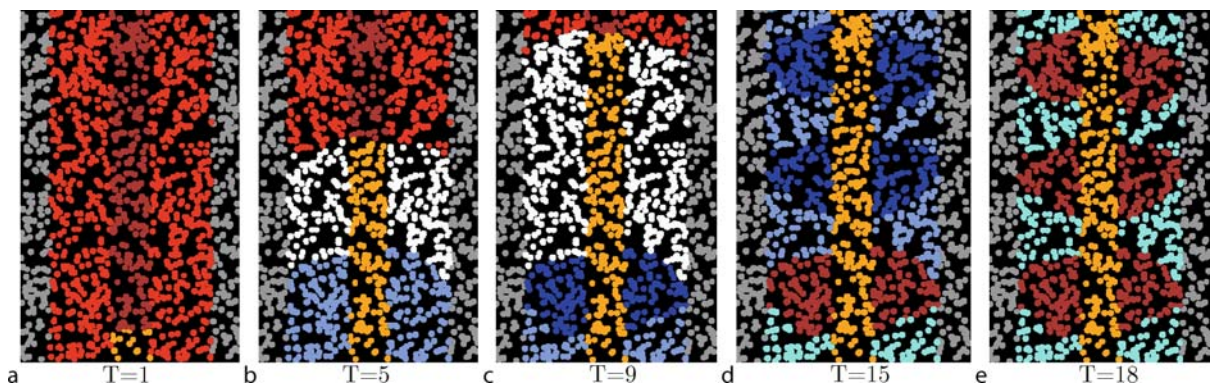
Weiss's system is powerful, but the level of abstraction is very low. This is because it was motivated by cellular engineering—as something that can be directly implemented by genetic regulatory networks. The language is therefore more useful as a tool set in which to implement higher-level languages such as GPL (see below), serving as a demonstration that in principle, these higher-level languages can be implemented by genetic regulatory networks as well.

Figure 1 shows an example simulation programmed in this language, that organizes an initially undifferentiated column of particles into a structure with band of two alternating colors: a caricature of somites in developing vertebræ.

The Growing Point Language

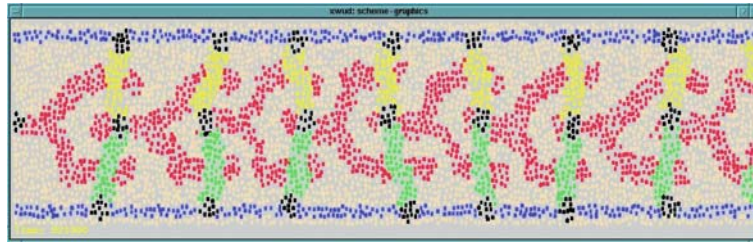
Coore's *Growing Point Language* (GPL) [15] demonstrates that an amorphous computer can be configured by a program that is common to all the computing elements to generate highly complex patterns, such as the pattern representing the interconnection structure of an arbitrary electrical circuit as shown in Fig. 2.

GPL is inspired by a botanical metaphor based on growing points and tropisms. A growing point is a locus of activity in an amorphous computer. A growing point propagates through the computer by transferring its activity from one computing element to a neighbor. As a growing point passes through the computer it effects the differentiation of the behaviors of the particles it visits. Particles secrete “chemical” signals whose count-up waves define gradients, and these attract or repel grow-



Amorphous Computing, Figure 1

A Microbial Colony Language program organizes a tube into a structure similar to that of somites in the developing vertebrate (from [55]).



Amorphous Computing, Figure 2

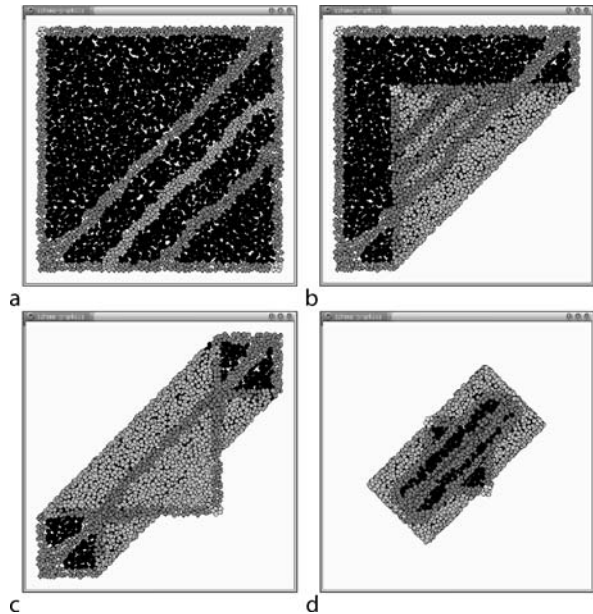
A pattern generated by GPL whose shape mimics a chain of CMOS inverters (from [15])

ing points as directed by programmer-specific “tropisms”. Coore demonstrated that these mechanisms are sufficient to permit amorphous computers to generate any arbitrary prespecified graph structure pattern, up to topology. Unlike real biology, however, once a pattern has been constructed, there is no clear mechanism to maintain it in the face of changes to the material. Also, from a programming linguistic point of view, there is no clear way to compose shapes by composing growing points. More recently, Gayle and Coore have shown how GPL may be extended to produce arbitrarily large patterns such as arbitrary text strings[21]. D’Hondt and D’Hondt have explored the use of GPL for geometrical constructions and its relations with computational geometry [18,19].

Origami-Based Self-Assembly

Nagpal [38] developed a prototype model for controlling programmable materials. She showed how to organize a program to direct an amorphous sheet of deformable particles to cooperate to construct a large family of globally-specified predetermined shapes. Her method, which is inspired by the folding of epithelial tissue, allows a programmer to specify a sequence of folds, where the set of available folds is sufficient to create any origami shape (as shown by Huzita’s axioms for origami [23]). Figure 3 shows a sheet of amorphous particles, where particles can cooperate to create creases and folds, assembling itself into the well-known origami “cup” structure.

Nagpal showed how this language of folds can be compiled into a low-level program that can be distributed to all of the particles of the amorphous sheet, similar to Coore’s GPL or Weiss’s MCL. With a few differences of initial state (for example, particles at the edges of the sheet know that they are edge particles) the particles run their copies of the program, interact with their neighbors, and fold up to make the predetermined shape. This technique is quite robust. Nagpal studied the range of shapes that can be constructed using her method, and on their sensitivity



Amorphous Computing, Figure 3

Folding an envelope structure (from [38]). A pattern of lines is constructed according to origami axioms. Elements then coordinate to fold the sheet using an actuation model based on epithelial cell morphogenesis. In the figure, black indicates the front side of the sheet, grey indicates the back side, and the various colored bands show the folds and creases that are generated by the amorphous process. The small white spots show gaps in the sheet cause by “dead” or missing cells—the process works despite these

to errors of communication, random cell death, and density of the cells.

As a programming framework, the origami language has more structure than the growing point language, because the origami methods allow composition of shape constructions. On the other hand, once a shape has been constructed, there is no clear mechanism to maintain existing patterns in the face of changes to the material.

Dynamic Recruitment

In Butera [10]’s “paintable computing”, processes dynamically recruit computational particles from an amorphous computer to implement their goals. As one of his examples Butera uses dynamic recruitment to implement a robust storage system for streaming audio and images (Fig. 4). Fragments of the image and audio stream circulate freely through the amorphous computer and are marshaled to a port when needed. The audio fragments also sort themselves into a playable stream as they migrate to the port.

To enhance robustness, there are multiple copies of the lower resolution image fragments and fewer copies of the higher resolution image fragments. Thus, the image is hard to destroy; with lost fragments the image is degraded but not destroyed.

Clement and Nagpal [12] also use dynamic recruitment in the development of active gradients, as described below.

Growth and Regeneration

Kondacs [30] showed how to synthesize arbitrary two-dimensional shapes by growing them. These computing units, or “cells”, are identically-programmed and decentralized, with the ability to engage in only limited, local communication. Growth is implemented by allowing cells to multiply. Each of his cells may create a child cell and place it randomly within a ring around the mother cell. Cells may also die, a fact which Kondacs puts to use for temporary scaffolding when building complex shapes. If a structure requires construction of a narrow neck between two other structures it can be built precisely by laying down a thick connection and later trimming it to size.

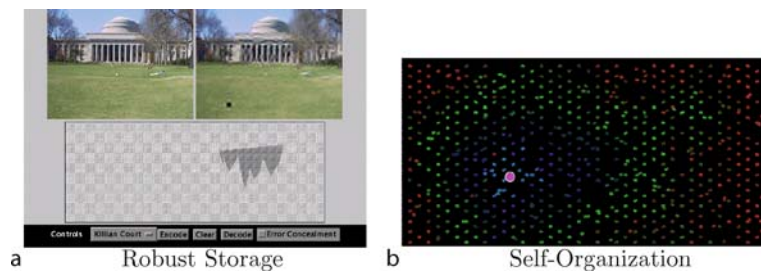
Attributes of this system include scalability, robustness, and the ability for self-repair. Just as a starfish can re-

generate its entire body from part of a limb, his system can self-repair in the event of agent death: his sphere-network representation allows the structure to be grown starting from any sphere, and every cell contains all necessary information for reproducing the missing structure.

Abstraction to Continuous Space and Time

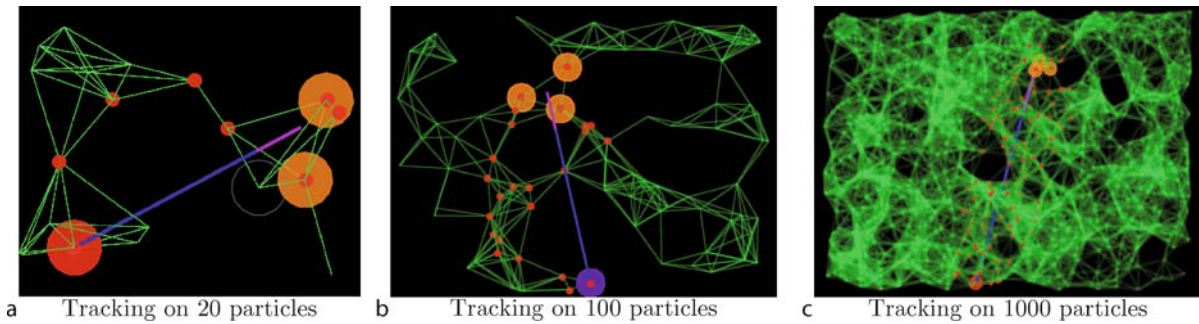
The amorphous model postulates computing particles distributed throughout a space. If the particles are dense, one can imagine the particles as actually filling the space, and create programming abstractions that view the space itself as the object being programmed, rather than the collection of particles. Beal and Bachrach [1,7] pursued this approach by creating a language, Proto, where programmers specify the behavior of an amorphous computer as though it were a continuous material filling the space it occupies. Proto programs manipulate fields of values spanning the entire space. Programming primitives are designed to make it simple to compile global operations to operations at each point of the continuum. These operations are approximated by having each device represent a nearby chunk of space. Programs are specified in space and time units that are independent of the distribution of particles and of the particulars of communication and execution on those particles (Fig. 5). Programs are composed functionally, and many of the details of communication and composition are made implicit by Proto’s runtime system, allowing complex programs to be expressed simply. Proto has been applied to applications in sensor networks like target tracking and threat avoidance, to swarm robotics and to modular robotics, e. g., generating a planar wave for coordinated actuation.

Newton’s language Regiment [41,42] also takes a continuous view of space and time. Regiment is organized in terms of stream operations, where each stream represents



Amorphous Computing, Figure 4

Butera dynamically controls the flow of information through an amorphous computer. In **a** image fragments spread through the computer so that a degraded copy can be recovered from any segment; the original image is on the left, the blurry copy on the right has been recovered from the small region shown below. In **b** audio fragments sort themselves into a playable stream as they migrate to an output port; cooler colors are earlier times (from [10]).



Amorphous Computing, Figure 5

A tracking program written in Proto sends the location of a target region (*orange*) to a listener (*red*) along a channel (*small red dots*) in the network (indicated by *green lines*). The continuous space and time abstraction allows the same program to run at different resolutions

a time-varying quantity over a part of space, for example, the average value of the temperature over a disc of a given radius centered at a designated point. Regiment, also a functional language, is designed to gather streams of data from regions of the amorphous computer and accumulate them at a single point. This assumption allows Regiment to provide region-wide summary functions that are difficult to implement in Proto.

Primitives for Amorphous Computing

The previous section illustrated some paradigms that have been developed for programming amorphous systems, each paradigm building on some organizing metaphor. But eventually, meeting the challenge of amorphous systems will require a more comprehensive linguistic framework. We can approach the task of creating such a framework following the perspective in [59], which views languages in terms of primitives, means of combination, and means of abstraction.

The fact that amorphous computers consist of vast numbers of unreliable and unsynchronized particles, arranged in space in ways that are locally unknown, constrains the primitive mechanisms available for organizing cooperation among the particles. While amorphous computers are naturally massively parallel, the kind of computation that they are most suited for is parallelism that does not depend on explicit synchronization and the use of atomic operations to control concurrent access to resources. However, there are large classes of useful behaviors that can be implemented without these tools. Primitive mechanisms that are appropriate for specifying behavior on amorphous computers include gossip, random choice, fields, and gradients.

Gossip

Gossip, also known as epidemic communication [17,20], is a simple communication mechanism. The goal of a gossip computation is to obtain an agreement about the value of some parameter. Each particle broadcasts its opinion of the parameter to its neighbors, and computation is performed by each particle combining the values that it receives from its neighbors, without consideration of the identification of the source. If the computation changes a particle's opinion of the value, it rebroadcasts its new opinion. The process concludes when there are no further broadcasts.

For example, an aggregate can agree upon the minimum of the values held by all the particles as follows. Each particle broadcasts its value. Each recipient compares its current value with the value that it receives. If the received value is smaller than its current value, it changes its current value to that minimum and rebroadcasts the new value.

The advantage of gossip is that it flows in all directions and is very difficult to disrupt. The disadvantage is that the lack of source information makes it difficult to revise a decision.

Random Choice

Random choice is used to break symmetry, allowing the particles to differentiate their behavior. The simplest use of random choice is to establish local identity of particles: each particle chooses a random number to identify itself to its neighbors. If the number of possible choices is large enough, then it is unlikely that any nearby particles will choose the same number, and this number can thus be used as an identifier for the particle to its neighbors. Random choice can be combined with gossip to elect

leaders, either for the entire system or for local regions. (If collisions in choice can be detected, then the number of choices need not be much higher than the number of neighbors. Also, using gossip to elect leaders makes sense only when we expect a leader to be long-lived, due to the difficulty of changing the decision to designate a replacement leader.)

To elect a single leader for the entire system, every particle chooses a value, then gossips to find the minimum. The particle with the minimum value becomes the leader. To elect regional leaders, we instead use gossip to carry the identity of the first leader a particle has heard of. Each particle uses random choice as a “coin flip” to decide when to declare itself a leader; if the flip comes up heads enough times before the particle hears of another leader, the particle declares itself a leader and broadcasts that fact to its neighbors. The entire system is thus broken up into contiguous domains of particles who first heard some particular particle declare itself a leader.

One challenge in using random choice on an amorphous computer is to ensure that the particulars of particle distribution do not have an unexpected effect on the outcome. For example, if we wish to control the expected size of the domain that each regional leader presides over, then the probability of becoming a leader must depend on the density of the particles.

Fields

Every component of the state of the computational particles in an amorphous computer may be thought of as a field over the discrete space occupied by those particles. If the density of particles is large enough this field of values may be thought of as an approximation of a field on the continuous space.

We can make amorphous models that approximate the solutions of the classical partial-differential equations of physics, given appropriate boundary conditions. The amorphous methods can be shown to be consistent, convergent and stable.

For example, the algorithm for solving the Laplace equation with Dirichlet conditions is analogous to the way it would be solved on a lattice. Each particle must repeatedly update the value of the solution to be the average of the solutions posted by its neighbors, but the boundary points must not change their values. This algorithm will eventually converge, although very slowly, independent of the order of the updates and the details of the local connectedness of the network. There are optimizations, such as over-relaxation, that are just as applicable in the amorphous context as on a regular grid.

Katzenelson [27] has shown similar results for the diffusion equation, complete with analytic estimates of the errors that arise from the discrete and irregularly connected network. In the diffusion equation there is a conserved quantity, the amount of material diffusing. Rauch [47] has shown how this can work with the wave equation, illustrating that systems that conserve energy and momentum can also be effectively modeled with an amorphous computer. The simulation of the wave equation does require that the communicating particles know their relative positions, but it is not hard to establish local coordinate systems.

Gradients

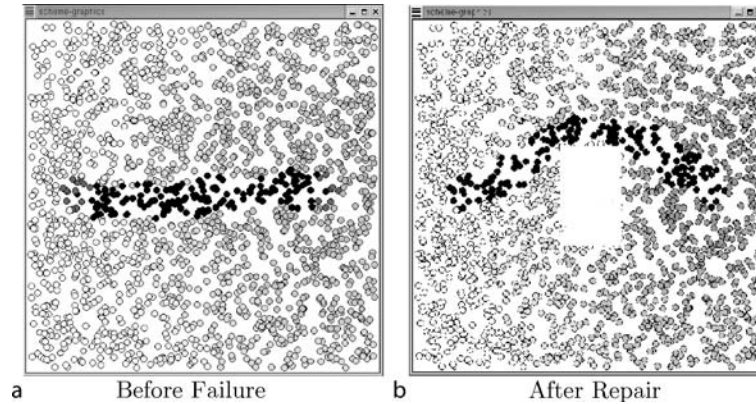
An important primitive in amorphous computing is the gradient, which estimates the distance from each particle to the nearest particle designated as a source. The gradient is inspired by the chemical-gradient diffusion process that is crucial to biological development. Amorphous computing builds on this idea, but does not necessarily compute the distance using diffusion because simulating diffusion can be expensive.

The common alternative is a linear-time mechanism that depends on active computation and relaying of information rather than passive diffusion. Calculation of a gradient starts with each source particle setting its distance estimate to zero, and every other particle setting its distance estimate to infinity. The sources then broadcast their estimates to their neighbors. When a particle receives a message from its neighbor, it compares its current distance estimate to the distance through its neighbor. If the distance through its neighbor is less, it chooses that to be its estimate, and broadcasts its new estimate onwards.

Although the basic form of the gradient is simple, there are several ways in which gradients can be varied to better match the context in which they are used. These choices may be made largely independently, giving a wide variety of options when designing a system. Variations which have been explored include:

Active Gradients An active gradient [12,14,16] monitors its validity in the face of changing sources and device failure, and maintains correct distance values. For example, if the supporting sources disappear, the gradient is deallocated. A gradient may also carry version information, allowing its source to change more smoothly.

Active gradients can provide self-repairing coordinate systems, as a foundation for robust construction of patterns. Figure 6 shows the “count-down wave” line described above in the introduction to this article. The line’s



Amorphous Computing, Figure 6

A line being maintained by active gradients, from [12]. A line (black) is constructed between two anchor regions (dark grey) based on the active gradient emitted by the right anchor region (light greys). The line is able to rapidly repair itself following failures because the gradient actively maintains itself

implementation in terms of active gradients provides for self-repair when the underlying amorphous computer is damaged.

Polarity A gradient may be set to count down from a positive value at the source, rather than to count up from zero. This bounds the distance that the gradient can span, which can help limit resource usage, but may limit the scalability of programs.

Adaptivity As described above, a gradient relaxes once to a distance estimate. If communication is expensive and precision unimportant, the gradient can take the first value that arrives and ignore all subsequent values. If we want the gradient to adapt to changes in the distribution of particles or sources, then the particles need to broadcast at regular intervals. We can then have estimates that converge smoothly to precise estimates by adding a restoring force which acts opposite to the relaxation, allowing the gradient to rise when unconstrained by its neighbors. If, on the other hand, we value adaptation speed over smoothness, then each particle can recalculate its distance estimate from scratch with each new batch of values.

Carrier Normally, the distance value calculated by the gradient is the signal we are interested in. A gradient may instead be used to carry an arbitrary signal outward from the source. In this case, the value at each particle is the most recently arrived value from the nearest source.

Distance Measure A gradient's distance measure is, of course, dependent on how much knowledge we have about the relative positions of neighbors. It is sometimes advan-

tageous to discard good information and use only hop-count values, since it is easier to make an adaptive gradient using hop-count values. Non-linear distance measures are also possible, such as a count-down gradient that decays exponentially from the source. Finally, the value of a gradient may depend on more sources than the nearest (this is the case for a chemical gradient), though this may be very expensive to calculate.

Coordinates and Clusters

Computational particles may be built with restrictions about what can be known about local geometry. A particle may know that it can reliably communicate with a few neighbors. If we assume that these neighbors are all within a disc of some approximate communication radius then distances to others may be estimated by minimum hop count [28]. However, it is possible that more elaborate particles can estimate distances to near neighbors. For example, the Cricket localization system [45] uses the fact that sound travels more slowly than radio, so the distance is estimated by the difference in time of arrival between simultaneously transmitted signals. McLurkin's swarmbots [35] use the ISIS communication system that gives bearing and range information. However, it is possible for a sufficiently dense amorphous computer to produce local coordinate systems for its particles with even the crudest method of determining distances. We can make an atlas of overlapping coordinate systems, using random symmetry breaking to make new starting baselines [3]. These coordinate systems can be combined and made consistent to form a manifold, even if the amorphous computer is not flat or simply connected.

One way to establish coordinates is to choose two initial particles that are a known distance apart. Each one serves as the source of a gradient. A pair of rectangular axes can be determined by the shortest path between them and by a bisector constructed where the two gradients are equal. These may be refined by averaging and calibrated using the known distance between the selected particles. After the axes are established, they may source new gradients that can be combined to make coordinates for the region near these axes. The coordinate system can be further refined using further averaging. Other natural coordinate constructions are bipolar elliptical. This kind of construction was pioneered by Coore [14] and Nagpal [37]. Katzenelson [27] did early work to determine the kind of accuracy that can be expected from such a construction.

Spatial clustering can be accomplished with any of a wide variety of algorithms, such as the clubs algorithm [16], LOCI [36], or persistent node partitioning [4]. Clusters can themselves be clustered, forming a hierarchical clustering of logarithmic height.

Means of Combination and Abstraction

A programming framework for amorphous systems requires more than primitive mechanisms. We also need suitable means of combination, so that programmers can combine behaviors to produce more complex behaviors, and means of abstraction so that the compound behaviors can be named and manipulated as units. Here are a few means of combination that have been investigated with amorphous computing.

Spatial and Temporal Sequencing

Several behaviors can be strung together in a sequence. The challenge in controlling such a sequence is to determine when one phase has completed and it is safe to move on to the next. Trigger rules can be used to detect completion locally.

In Coore's Growing Point Language [15], all of the sequencing decisions are made locally, with different growing points progressing independently. There is no difficulty of synchronization in this approach because the only time when two growing points need to agree is when they have become spatially coincident. When growing points merge, the independent processes are automatically synchronized.

Nagpal's origami language [38] has long-range operations that cannot overlap in time unless they are in non-interacting regions of the space. The implementation uses barrier synchronization to sequence the operations: when completion is detected locally, a signal is propagated

throughout a marked region of the sheet, and the next operation begins after a waiting time determined by the diameter of the sheet.

With adaptive gradients, we can use the presence of an inducing signal to run an operation. When the induction signal disappears, the operation ceases and the particles begin the next operation. This allows sequencing to be triggered by the last detection of completion rather than by the first.

Pipelining

If a behavior is self-stabilizing (meaning that it converges to a correct state from any arbitrary state) then we can use it in a sequence without knowing when the previous phase completes. The evolving output of the previous phase serves as the input of this next phase, and once the preceding behavior has converged, the self-stabilizing behavior will converge as well.

If the previous phase evolves smoothly towards its final state, then by the time it has converged, the next phase may have almost converged as well, working from its partial results. For example, the coordinate system mechanism described above can be pipelined; the final coordinates are being formed even as the farther particles learn that they are not on one of the two axes.

Restriction to Spatial Regions

Because the particles of an amorphous computer are distributed in space it is natural to assign particular behaviors to specific spatial regions. In Beal and Bachrach's work, restriction of a process to a region is a primitive [7]. As another example, when Nagpal's system folds an origami construction, regions on different faces may differentiate so that they fold in different patterns. These folds may, if the physics permits, be performed simultaneously. It may be necessary to sequence later construction that depends on the completion of the substructures.

Regions of space can be named using coordinates, clustering, or implicitly through calculations on fields. Indeed, one could implement solid modelling on an amorphous computer. Once a region is identified, a particle can test whether it is a member of that region when deciding whether to run a behavior. It is also necessary to specify how a particle should change its behavior if its membership in a region may vary with time.

Modularity and Abstraction

Standard means of abstraction may be applied in an amorphous computing context, such as naming procedures,

data structures, and processes. The question for amorphous computing is what collection of entities is useful to name.

Because geometry is essential in an amorphous computing context, it becomes appropriate to describe computational processes in terms of geometric entities. Thus there are new opportunities for combining geometric structures and naming the combinations. For example, it is appropriate to compute with, combine, and name regions of space, intervals of time, and fields defined on them [7,42]. It may also be useful to describe the propagation of information through the amorphous computer in terms of the light cone of an event [2].

Not all traditional abstractions extend nicely to an amorphous computing context because of the challenges of scale and the fallibility of parts and interconnect. For example, atomic transactions may be excessively expensive in an amorphous computing context. And yet, some of the goals that a programmer might use atomic transactions to accomplish, such as the approximate enforcement of conservation laws, can be obtained using techniques that are compatible with an amorphous environment, as shown by Rauch [47].

Supporting Infrastructure and Services

Amorphous computing languages, with their primitives, means of combination, and means of abstraction, rest on supporting services. One example, described above, is the automatic message propagation and decay in Weiss's Microbial Colony Language [55]. MCL programs do not need to deal with this explicitly because it is incorporated into the operating system of the MCL machine. Experience with amorphous computing is beginning to identify other key services that amorphous machines must supply.

Particle Identity

Particles must be able to choose identifiers for communicating with their neighbors. More generally, there are many operations in an amorphous computation where the particles may need to choose numbers, with the property that individual particles choose different numbers.

If we are willing to pay the cost, it is possible to build unique identifiers into the particles, as is done with current macroscopic computers. We need only locally unique identifiers, however, so we can obtain them using pseudorandom-number generators. On the surface of it, this may seem problematic, since the particles in the amorphous are assumed to be manufactured identically, with identical programs. There are, however, ways to obtain individualized random numbers. For example, the particles

are not synchronized, and they are not really physically identical, so they will run at slightly different rates. This difference is enough to allow pseudorandom-number generators to get locally out of synch and produce different sequences of numbers. Amorphous computing particles that have sensors may also get seeds for their pseudorandom-number generators from sensor noise.

Local Geometry and Gradients

Particles must maintain connections with their neighbors, tracking who they can reliably communicate with, and whatever local geometry information is available. Because particles may fail or move, this information needs to be maintained actively. The geometry information may include distance and bearing to each neighbor, as well as the time it takes to communicate with each neighbor. But many implementations will not be able to give significant distance or bearing information. Since all of this information may be obsolete or inaccurately measured, the particles must also maintain information on how reliable each piece of information is.

An amorphous computer must know the dimension of the space it occupies. This will generally be a constant—either the computer covers a surface or fills a volume. In rare cases, however, the effective dimension of a computer may change: for example, paint is three-dimensional in a bucket and two-dimensional once applied. Combining this information with how the number of accessible correspondents changes with distance, an amorphous process can derive curvature and local density information.

An amorphous computer should also support gradient propagation as part of the infrastructure: a programmer should not have to explicitly deal with the propagation of gradients (or other broadcast communications) in each particle. A process may explicitly initiate a gradient, or explicitly react to one that it is interested in, but the propagation of the gradient through a particle should be automatically maintained by the infrastructure.

Implementing Communication

Communication between neighbors can occur through any number of mechanisms, each with its own set of properties: amorphous computing systems have been built that communicate through directional infrared [35], RF broadcast [22], and low-speed serial cables [9,46]. Simulated systems have also included other mechanisms such as signals superimposed on the power connections [11] and chemical diffusion [55].

Communication between particles can be made implicit with a neighborhood shared memory. In this arrangement, each particle designates some of its internal state to be shared with its neighbors. The particles regularly communicate, giving each particle a best-effort view of the exposed portions of the states of its neighbors. The contents of the exposed state may be specified explicitly [10,35,58] or implicitly [7]. The shared memory allows the system to be tuned by trading off communication rate against the quality of the synchronization, and decreasing transmission rates when the exposed state is not changing.

Lessons for Engineering

As von Neumann remarked half a century ago, biological systems are strikingly robust when compared with our artificial systems. Even today software is fragile. Computer science is currently built on a foundation that largely assumes the existence of a perfect infrastructure. Integrated circuits are fabricated in clean-room environments, tested deterministically, and discarded if even a single defect is uncovered. Entire software systems fail with single-line errors. In contrast, biological systems rely on local computation, local communication, and local state, yet they exhibit tremendous resilience.

Although this contrast is most striking in computer science, amorphous computing can provide lessons throughout engineering. Amorphous computing concentrates on making systems flexible and adaptable at the expense of efficiency. Amorphous computing requires an engineer to work under extreme conditions. The engineer must arrange the cooperation of vast numbers of identical computational particles to accomplish prespecified goals, but may not depend upon the numbers. We may not depend on any prespecified interconnect of the particles. We may not depend on synchronization of the particles. We may not depend on the stability of the communications system. We may not depend on the long-term survival of any individual particles. The combination of these obstacles forces us to abandon many of the comforts that are available in more typical engineering domains.

By restricting ourselves in this way we obtain some robustness and flexibility, at the cost of potentially inefficient use of resources, because the algorithms that are appropriate are ones that do not take advantage of these assumptions. Algorithms that work well in an amorphous context depend on the average behavior of participating particles. For example, in Nagpal's origami system a fold that is specified will be satisfactory if it is approximately in the right place and if most of the particles on the specified fold

line agree that they are part of the fold line: dissenters will be overruled by the majority. In Proto a programmer can address only regions of space, assumed to be populated by many particles. The programmer may not address individual particles, so failures of individual particles are unlikely to make major perturbations to the behavior of the system.

An amorphous computation can be quite immune to details of the macroscopic geometry as well as to the interconnectedness of the particles. Since amorphous computations make their own local coordinate systems, they are relatively independent of coordinate distortions. In an amorphous computation we accept a wide range of outcomes that arise from variations of the local geometry. Tolerance of local variation can lead to surprising flexibility: the mechanisms which allow Nagpal's origami language to tolerate local distortions allow programs to distort globally as well, and Nagpal shows how such variations can account for the variations in the head shapes of related species of *Drosophila* [38]. In Coore's language one specifies the topology of the pattern to be constructed, but only limited information about the geometry. The topology will be obtained, regardless of the local geometry, so long as there is sufficient density of particles to support the topology. Amorphous computations based on a continuous model of space (as in Proto) are naturally scale independent.

Since an amorphous computer is composed of unsynchronized particles, a program may not depend upon a priori timing of events. The sequencing of phases of a process must be determined by either explicit termination signals or with times measured dynamically. So amorphous computations are time-scale independent by construction.

A program for an amorphous computer may not depend on the reliability of the particles or the communication paths. As a consequence it is necessary to construct the program so as to dynamically compensate for failures. One way to do this is to specify the result as the satisfaction of a set of constraints, and to build the program as a homeostatic mechanism that continually drives the system toward satisfaction of those constraints. For example, an active gradient continually maintains each particle's estimate of the distance to the source of the gradient. This can be used to establish and maintain connections in the face of failures of particles or relocation of the source. If a system is specified in this way, repair after injury is a continuation of the development process: an injury causes some constraints to become unsatisfied, and the development process builds new structure to heal the injury.

By restricting the assumptions that a programmer can rely upon we increase the flexibility and reliability of the

programs that are constructed. However, it is not yet clear how this limits the range of possible applications of amorphous computing.

Future Directions

Computer hardware is almost free, and in the future it will continue to decrease in price and size. Sensors and actuators are improving as well. Future systems will have vast numbers of computing mechanisms with integrated sensors and actuators, to a degree that outstrips our current approaches to system design. When the numbers become large enough, the appropriate programming technology will be amorphous computing. This transition has already begun to appear in several fields:

Sensor networks The success of sensor network research has encouraged the planning and deployment of ever-larger numbers of devices. The ad-hoc, time-varying nature of sensor networks has encouraged amorphous approaches, such as communication through directed diffusion [25] and Newton's Regiment language [42].

Robotics Multi-agent robotics is much like sensor networks, except that the devices are mobile and have actuators. *Swarm robotics* considers independently mobile robots working together as a team like ants or bees, while *modular robotics* consider robots that physically attach to one another in order to make shapes or perform actions, working together like the cells of an organism. Gradients are being used to create "flocking" behaviors in swarm robotics [35,44]. In modular robotics, Stoy uses gradients to create shapes [51] while De Rosa et al. form shapes through stochastic growth and decay [49].

Pervasive computing Pervasive computing seeks to exploit the rapid proliferation of wireless computing devices throughout our everyday environment. Mamei and Zambonelli's TOTA system [32] is an amorphous computing implementation supporting a model of programming using fields and gradients [33]. Servat and Drogoul have suggested combining amorphous computing and reactive agent-based systems to produce something they call "pervasive intelligence" [50].

Multicore processors As it becomes more difficult to increase processor speed, chip manufacturers are looking for performance gains through increasing the number of processing cores per chip. Butera's work [10] looks toward a future in which there are thousands of cores per chip and it is no longer reasonable to assume they are all working or have them communicate all-to-all.

While much of amorphous computing research is inspired by biological observations, it is also likely that insights and lessons learned from programming amorphous computers will help elucidate some biological problems [43]. Some of this will be stimulated by the emerging engineering of biological systems. Current work in synthetic biology [24,48,61] is centered on controlling the molecular biology of cells. Soon synthetic biologists will begin to engineer biofilms and perhaps direct the construction of multicellular organs, where amorphous computing will become an essential technological tool.

Bibliography

Primary Literature

1. Bachrach J, Beal J (2006) Programming a sensor network as an amorphous medium. In: DCOSS 2006 Posters, June 2006
2. Bachrach J, Beal J, Fujiwara T (2007) Continuous space-time semantics allow adaptive program execution. In: IEEE International Conference on Self-Adaptive and Self-Organizing Systems, 2007
3. Bachrach J, Nagpal R, Salib M, Shrobe H (2003) Experimental results and theoretical analysis of a self-organizing global coordinate system for ad hoc sensor networks. *Telecommun Syst J*, Special Issue on Wireless System Networks 26(2-4):213-233
4. Beal J (2003) A robust amorphous hierarchy from persistent nodes. In: *Commun Syst Netw*
5. Beal J (2004) Programming an amorphous computational medium. In: *Unconventional Programming Paradigms International Workshop*, September 2004
6. Beal J (2005) Amorphous medium language. In: *Large-Scale Multi-Agent Systems Workshop (LSMAS)*. Held in Conjunction with AAMAS-05
7. Beal J, Bachrach J (2006) Infrastructure for engineered emergence on sensor/actuator networks. In: *IEEE Intelligent Systems*, 2006
8. Beal J, Sussman G (2005) Biologically-inspired robust spatial programming. Technical Report AI Memo 2005-001, MIT, January 2005
9. Beebe W M68hc11 gunk api book. <http://www.swiss.ai.mit.edu/projects/amorphous/Hc11/api.html>. Accessed 31 May 2007
10. Butera W (2002) Programming a Paintable Computer. Ph D thesis, MIT
11. Campbell J, Pillai P, Goldstein SC (2005) The robot is the tether: Active, adaptive power routing for modular robots with unary inter-robot connectors. In: *IROS 2005*
12. Clement L, Nagpal R (2003) Self-assembly and self-repairing topologies. In: *Workshop on Adaptability in Multi-Agent Systems, RoboCup Australian Open*, January 2003
13. Codd EF (1968) *Cellular Automata*. Academic Press, New York
14. Coore D (1998) Establishing a coordinate system on an amorphous computer. In: *MIT Student Workshop on High Performance Computing*, 1998
15. Coore D (1999) *Botanical Computing: A Developmental Approach to Generating Interconnect Topologies on an Amorphous Computer*. Ph D thesis, MIT

16. Coore D, Nagpal R, Weiss R (1997) Paradigms for structure in an amorphous computer. Technical Report AI Memo 1614, MIT
17. Demers A, Greene D, Hauser C, Irish W, Larson J, Shenker S, Stuygis H, Swinehart D, Terry D (1987) Epidemic algorithms for replicated database maintenance. In: 7th ACM Symposium on Operating Systems Principles, 1987
18. D'Hondt E, D'Hondt T (2001) Amorphous geometry. In: ECAL 2001
19. D'Hondt E, D'Hondt T (2001) Experiments in amorphous geometry. In: 2001 International Conference on Artificial Intelligence
20. Ganesan D, Krishnamachari B, Woo A, Culler D, Estrin D, Wicker S (2002) An empirical study of epidemic algorithms in large scale multihop wireless networks. Technical Report IRB-TR-02-003, Intel Research Berkeley
21. Gayle O, Coore D (2006) Self-organizing text in an amorphous environment. In: ICCS 2006
22. Hill J, Szwedcyk R, Woo A, Culler D, Hollar S, Pister K (2000) System architecture directions for networked sensors. In: ASPLOS, November 2000
23. Huzita H, Scimemi B (1989) The algebra of paper-folding. In: First International Meeting of Origami Science and Technology, 1989
24. igem 2006: international genetically engineered machine competition (2006) <http://www.igem2006.com>. Accessed 31 May 2007
25. Intanagonwiwat C, Govindan R, Estrin D (2000) Directed diffusion: a scalable and robust communication paradigm for sensor networks. In: Mobile Computing and Networking, pp 56–67
26. Kahn JM, Katz RH, Pister KSJ (1999) Mobile networking for smart dust. In: ACM/IEEE Int. Conf. on Mobile Computing and Networking (MobiCom 99), August 1999
27. Katzenelson J (1999) Notes on amorphous computing. (Unpublished Draft)
28. Kleinrock L, Sylvester J (1978) Optimum transmission radii for packet radio networks or why six is a magic number. In: IEEE Natl Telecommun. Conf, December 1978, pp 4.3.1–4.3.5
29. Knight TF, Sussman GJ (1998) Cellular gate technology. In: First International Conference on Unconventional Models of Computation (UMC98)
30. Kondacs A (2003) Biologically-inspired self-assembly of 2d shapes, using global-to-local compilation. In: International Joint Conference on Artificial Intelligence (IJCAI)
31. Mamei M, Zambonelli F (2003) Spray computers: Frontiers of self-organization for pervasive computing. In: WOA 2003
32. Mamei M, Zambonelli F (2004) Spatial computing: the tota approach. In: WOA 2004, pp 126–142
33. Mamei M, Zambonelli F (2005) Physical deployment of digital pheromones through rfid technology. In: AAMAS 2005, pp 1353–1354
34. Margolus N (1988) Physics and Computation. Ph D thesis, MIT
35. McLurkin J (2004) Stupid robot tricks: A behavior-based distributed algorithm library for programming swarms of robots. Master's thesis, MIT
36. Mittal V, Demirbas M, Arora A (2003) Loci: Local clustering service for large scale wireless sensor networks. Technical Report OSU-CISRC-2/03-TR07, Ohio State University
37. Nagpal R (1999) Organizing a global coordinate system from local information on an amorphous computer. Technical Report AI Memo 1666, MIT
38. Nagpal R (2001) Programmable Self-Assembly: Constructing Global Shape using Biologically-inspired Local Interactions and Origami Mathematics. Ph D thesis, MIT
39. Nagpal R, Mamei M (2004) Engineering amorphous computing systems. In: Bergenti F, Gleizes MP, Zambonelli F (eds) Methodologies and Software Engineering for Agent Systems, The Agent-Oriented Software Engineering Handbook. Kluwer, New York, pp 303–320
40. Dust Networks. <http://www.dust-inc.com>. Accessed 31 May 2007
41. Newton R, Morrisett G, Welsh M (2007) The regiment macro-programming system. In: International Conference on Information Processing in Sensor Networks (IPSN'07)
42. Newton R, Welsh M (2004) Region streams: Functional macro-programming for sensor networks. In: First International Workshop on Data Management for Sensor Networks (DMSN), August 2004
43. Patel A, Nagpal R, Gibson M, Perrimon N (2006) The emergence of geometric order in proliferating metazoan epithelia. *Nature* 442:1038–1041
44. Payton D, Daily M, Estowski R, Howard M, Lee C (2001) Pheromone robotics. *Autonomous Robotics* 11:319–324
45. Priyantha N, Chakraborty A, Balakrishnan H (2000) The cricket location-support system. In: ACM International Conference on Mobile Computing and Networking (ACM MOBICOM), August 2000
46. Raffle H, Parkes A, Ishii H (2004) Topobo: A constructive assembly system with kinetic memory. CHI
47. Rauch E (1999) Discrete, amorphous physical models. Master's thesis, MIT
48. Registry of standard biological parts. <http://parts.mit.edu>. Accessed 31 May 2007
49. De Rosa M, Goldstein SC, Lee P, Campbell J, Pillai P (2006) Scalable shape sculpting via hole motion: Motion planning in lattice-constrained module robots. In: Proceedings of the 2006 IEEE International Conference on Robotics and Automation (ICRA '06), May 2006
50. Servat D, Drogoul A (2002) Combining amorphous computing and reactive agent-based systems: a paradigm for pervasive intelligence? In: AAMAS 2002
51. Stoy K (2003) Emergent Control of Self-Reconfigurable Robots. Ph D thesis, University of Southern Denmark
52. Sutherland A (2003) Towards rseam: Resilient serial execution on amorphous machines. Master's thesis, MIT
53. von Neumann J (1951) The general and logical theory of automata. In: Jeffress L (ed) *Cerebral Mechanisms for Behavior*. Wiley, New York, p 16
54. Weiss R, Knight T (2000) Engineered communications for microbial robotics. In: Sixth International Meeting on DNA Based Computers (DNA6)
55. Weiss R (2001) Cellular Computation and Communications using Engineered Genetic Regular Networks. Ph D thesis, MIT
56. Welsh M, Mainland G (2004) Programming sensor networks using abstract regions. In: Proceedings of the First USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI '04), March 2004
57. Werfel J, Bar-Yam Y, Nagpal R (2005) Building patterned structures with robot swarms. In: IJCAI
58. Whitehouse K, Sharp C, Brewer E, Culler D (2004) Hood: a neighborhood abstraction for sensor networks. In: Proceedings of the 2nd international conference on Mobile systems, applications, and services.

59. Abelson H, Sussman GJ, Sussman J (1996) *Structure and Interpretation of Computer Programs*, 2nd edn. MIT Press, Cambridge
60. Ashe HL, Briscoe J (2006) The interpretation of morphogen gradients. *Development* 133:385–94
61. Weiss R, Basu S, Hooshangi S, Kalmbach A, Karig D, Mehreja R, Netravali I (2003) Genetic circuit building blocks for cellular computation, communications, and signal processing. *Natural Computing* 2(1):47–84

Books and Reviews

- Abelson H, Allen D, Coore D, Hanson C, Homsy G, Knight T, Nagpal R, Rauch E, Sussman G, Weiss R (1999) *Amorphous computing*. Technical Report AIM-1665, MIT

Analog Computation

BRUCE J. MACLENNAN

Department of Electrical Engineering & Computer Science, University of Tennessee, Knoxville, USA

Article Outline

Glossary
 Definition of the Subject
 Introduction
 Fundamentals of Analog Computing
 Analog Computation in Nature
 General-Purpose Analog Computation
 Analog Computation and the Turing Limit
 Analog Thinking
 Future Directions
 Bibliography

Glossary

- Accuracy** The closeness of a computation to the corresponding primary system.
- BSS** The theory of computation over the real numbers defined by Blum, Shub, and Smale.
- Church–Turing (CT) computation** The model of computation based on the Turing machine and other equivalent abstract computing machines; commonly accepted as defining the limits of digital computation.
- EAC** Extended analog computer defined by Rubel.
- GPAC** General-purpose analog computer.
- Nomograph** A device for the graphical solution of equations by means of a family of curves and a straightedge.
- ODE** Ordinary differential equation.
- PDE** Partial differential equation.

Potentiometer A variable resistance, adjustable by the computer operator, used in electronic analog computing as an attenuator for setting constants and parameters in a computation.

Precision The quality of an analog representation or computation, which depends on both resolution and stability.

Primary system The system being simulated, modeled, analyzed, or controlled by an analog computer, also called the *target system*.

Scaling The adjustment, by constant multiplication, of variables in the primary system (including time) so that the corresponding variables in the analog systems are in an appropriate range.

TM Turing machine.

Definition of the Subject

Although analog computation was eclipsed by digital computation in the second half of the twentieth century, it is returning as an important alternative computing technology. Indeed, as explained in this article, theoretical results imply that analog computation can escape from the limitations of digital computation. Furthermore, analog computation has emerged as an important theoretical framework for discussing computation in the brain and other natural systems.

Analog computation gets its name from an *analogy*, or systematic relationship, between the physical processes in the computer and those in the system it is intended to model or simulate (the *primary system*). For example, the electrical quantities voltage, current, and conductance might be used as analogs of the fluid pressure, flow rate, and pipe diameter. More specifically, in traditional analog computation, physical quantities in the computation obey the same mathematical laws as physical quantities in the primary system. Thus the computational quantities are proportional to the modeled quantities. This is in contrast to *digital computation*, in which quantities are represented by strings of symbols (e.g., binary digits) that have no direct physical relationship to the modeled quantities. According to the *Oxford English Dictionary* (2nd ed., s.vv. analogue, digital), these usages emerged in the 1940s.

However, in a fundamental sense all computing is based on an analogy, that is, on a systematic relationship between the states and processes in the computer and those in the primary system. In a digital computer, the relationship is more abstract and complex than simple proportionality, but even so simple an analog computer as a slide rule goes beyond strict proportion (i.e., distance on

the rule is proportional to the logarithm of the number). In both analog and digital computation—indeed in all computation—the relevant abstract mathematical structure of the problem is realized in the physical states and processes of the computer, but the realization may be more or less direct [40,41,46].

Therefore, despite the etymologies of the terms “analog” and “digital”, in modern usage the principal distinction between digital and analog computation is that the former operates on discrete representations in discrete steps, while the later operated on continuous representations by means of continuous processes (e.g., MacLennan [46], Siegelmann [p. 147 in 78], Small [p. 30 in 82], Weyrick [p. 3 in 89]).

That is, the primary distinction resides in the *topologies* of the states and processes, and it would be more accurate to refer to *discrete* and *continuous computation* [p. 39 in 25]. (Consider so-called analog and digital clocks. The principal difference resides in the continuity or discreteness of the representation of time; the motion of the two (or three) hands of an “analog” clock do not mimic the motion of the rotating earth or the position of the sun relative to it.)

Introduction

History

Pre-electronic Analog Computation Just like digital calculation, analog computation was originally performed by hand. Thus we find several analog computational procedures in the “constructions” of Euclidean geometry (Euclid, fl. 300 BCE), which derive from techniques used in ancient surveying and architecture. For example, Problem II.51 is “to divide a given straight line into two parts, so that the rectangle contained by the whole and one of the parts shall be equal to the square of the other part”. Also, Problem VI.13 is “to find a mean proportional between two given straight lines”, and VI.30 is “to cut a given straight line in extreme and mean ratio”. These procedures do not make use of measurements in terms of any fixed unit or of digital calculation; the lengths and other continuous quantities are manipulated directly (via compass and straightedge). On the other hand, the techniques involve discrete, precise operational steps, and so they can be considered *algorithms*, but over continuous magnitudes rather than discrete numbers.

It is interesting to note that the ancient Greeks distinguished continuous *magnitudes* (Grk., *megethoi*), which have physical dimensions (e.g., length, area, rate), from discrete *numbers* (Grk., *arithmoi*), which do not [49]. Euclid axiomatizes them separately (magnitudes in Book V,

numbers in Book VII), and a mathematical system comprising both discrete and continuous quantities was not achieved until the nineteenth century in the work of Weierstrass and Dedekind.

The earliest known mechanical analog computer is the “Antikythera mechanism”, which was found in 1900 in a shipwreck under the sea near the Greek island of Antikythera (between Kythera and Crete). It dates to the second century BCE and appears to be intended for astronomical calculations. The device is sophisticated (at least 70 gears) and well engineered, suggesting that it was not the first of its type, and therefore that other analog computing devices may have been used in the ancient Mediterranean world [22]. Indeed, according to Cicero (*Rep.* 22) and other authors, Archimedes (c. 287–c. 212 BCE) and other ancient scientists also built analog computers, such as armillary spheres, for astronomical simulation and computation. Other antique mechanical analog computers include the astrolabe, which is used for the determination of longitude and a variety of other astronomical purposes, and the *torquetum*, which converts astronomical measurements between equatorial, ecliptic, and horizontal coordinates.

A class of special-purpose analog computer, which is simple in conception but may be used for a wide range of purposes, is the *nomograph* (also, *nomogram*, *alignment chart*). In its most common form, it permits the solution of quite arbitrary equations in three real variables, $f(u, v, w) = 0$. The nomograph is a chart or graph with scales for each of the variables; typically these scales are curved and have non-uniform numerical markings. Given values for any two of the variables, a straightedge is laid across their positions on their scales, and the value of the third variable is read off where the straightedge crosses the third scale. Nomographs were used to solve many problems in engineering and applied mathematics. They improve intuitive understanding by allowing the relationships among the variables to be visualized, and facilitate exploring their variation by moving the straightedge. Lipka (1918) is an example of a course in graphical and mechanical methods of analog computation, including nomographs and slide rules.

Until the introduction of portable electronic calculators in the early 1970s, the *slide rule* was the most familiar analog computing device. Slide rules use logarithms for multiplication and division, and they were invented in the early seventeenth century shortly after John Napier’s description of logarithms.

The mid-nineteenth century saw the development of the *field analogy method* by G. Kirchhoff (1824–1887) and others [33]. In this approach an electrical field in an elec-

trolytic tank or conductive paper was used to solve two-dimensional boundary problems for temperature distributions and magnetic fields [p. 34 in 82]. It is an early example of *analog field computation* (see ► [Field Computation in Natural and Artificial Intelligence](#)).

In the nineteenth century a number of mechanical analog computers were developed for integration and differentiation (e. g., Litka 1918, pp. 246–256; Clymer [15]). For example, the *planimeter* measures the area under a curve or within a closed boundary. While the operator moves a pointer along the curve, a rotating wheel accumulates the area. Similarly, the *integrator* is able to draw the integral of a given function as its shape is traced. Other mechanical devices can draw the derivative of a curve or compute a tangent line at a given point.

In the late nineteenth century William Thomson, Lord Kelvin, constructed several analog computers, including a “tide predictor” and a “harmonic analyzer”, which computed the Fourier coefficients of a tidal curve [85,86]. In 1876 he described how the mechanical integrators invented by his brother could be connected together in a feedback loop in order to solve second and higher order differential equations (Small [pp. 34–35, 42 in 82], Thomson [84]). He was unable to construct this *differential analyzer*, which had to await the invention of the torque amplifier in 1927.

The torque amplifier and other technical advancements permitted Vannevar Bush at MIT to construct the first practical differential analyzer in 1930 [pp. 42–45 in 82]. It had six integrators and could also do addition, subtraction, multiplication, and division. Input data were entered in the form of continuous curves, and the machine automatically plotted the output curves continuously as the equations were integrated. Similar differential analyzers were constructed at other laboratories in the US and the UK.

Setting up a problem on the MIT differential analyzer took a long time; gears and rods had to be arranged to define the required dependencies among the variables. Bush later designed a much more sophisticated machine, the Rockefeller Differential Analyzer, which became operational in 1947. With 18 integrators (out of a planned 30), it provided programmatic control of machine setup, and permitted several jobs to be run simultaneously. Mechanical differential analyzers were rapidly supplanted by electronic analog computers in the mid-1950s, and most were disassembled in the 1960s (Bowles [10], Owens [61], Small [pp. 50–45 in 82]).

During World War II, and even later wars, an important application of optical and mechanical analog computation was in “gun directors” and “bomb sights”, which

performed ballistic computations to accurately target artillery and dropped ordnance.

Electronic Analog Computation in the 20th Century It is commonly supposed that electronic analog computers were superior to mechanical analog computers, and they were in many respects, including speed, cost, ease of construction, size, and portability [pp. 54–56 in 82]. On the other hand, mechanical integrators produced higher precision results (0.1% vs. 1% for early electronic devices) and had greater mathematical flexibility (they were able to integrate with respect to any variable, not just time). However, many important applications did not require high precision and focused on dynamic systems for which time integration was sufficient.

Analog computers (non-electronic as well as electronic) can be divided into *active-element* and *passive-element* computers; the former involve some kind of amplification, the latter do not [pp. 2-1–4 in 87]. Passive-element computers included the *network analyzers*, which were developed in the 1920s to analyze electric power distribution networks, and which continued in use through the 1950s [pp. 35–40 in 82]. They were also applied to problems in thermodynamics, aircraft design, and mechanical engineering. In these systems networks or grids of resistive elements or reactive elements (i. e., involving capacitance and inductance as well as resistance) were used to model the spatial distribution of physical quantities such as voltage, current, and power (in electric distribution networks), electrical potential in space, stress in solid materials, temperature (in heat diffusion problems), pressure, fluid flow rate, and wave amplitude [p. 2-2 in 87]. That is, network analyzers dealt with partial differential equations (PDEs), whereas active-element computers, such as the differential analyzer and its electronic successors, were restricted to ordinary differential equations (ODEs) in which time was the independent variable. Large network analyzers are early examples of *analog field computers* (see ► [Field Computation in Natural and Artificial Intelligence](#)).

Electronic analog computers became feasible after the invention of the *DC operational amplifier* (“op amp”) c. 1940 [pp. 64, 67–72 in 82]. Already in the 1930s scientists at Bell Telephone Laboratories (BTL) had developed the DC-coupled feedback-stabilized amplifier, which is the basis of the op amp. In 1940, as the USA prepared to enter World War II, DL Parkinson at BTL had a dream in which he saw DC amplifiers being used to control an anti-aircraft gun. As a consequence, with his colleagues CA Lovell and BT Weber, he wrote a series of papers on “electrical mathematics”, which described electrical circuits to

“operationalize” addition, subtraction, integration, differentiation, etc. The project to produce an electronic gun-director led to the development and refinement of DC op amps suitable for analog computation.

The war-time work at BTL was focused primarily on control applications of analog devices, such as the gun-director. Other researchers, such as E. Lakatos at BTL, were more interested in applying them to general-purpose analog computation for science and engineering, which resulted in the design of the General Purpose Analog Computer (GPAC), also called “Gypsy”, completed in 1949 [pp. 69–71 in 82]. Building on the BTL op amp design, fundamental work on electronic analog computation was conducted at Columbia University in the 1940s. In particular, this research showed how analog computation could be applied to the simulation of dynamic systems and to the solution of nonlinear equations.

Commercial general-purpose analog computers (GPACs) emerged in the late 1940s and early 1950s [pp. 72–73 in 82]. Typically they provided several dozen integrators, but several GPACs could be connected together to solve larger problems. Later, large-scale GPACs might have up to 500 amplifiers and compute with 0.01%–0.1% precision [p. 2-33 in 87].

Besides integrators, typical GPACs provided adders, subtracters, multipliers, fixed function generators (e.g., logarithms, exponentials, trigonometric functions), and variable function generators (for user-defined functions) [Chaps. 1.3, 2.4 in 87]. A GPAC was programmed by connecting these components together, often by means of a patch panel. In addition, parameters could be entered by adjusting potentiometers (attenuators), and arbitrary functions could be entered in the form of graphs [pp. 1-72–81, 2-154–156 in 87]. Output devices plotted data continuously or displayed it numerically [pp. 3-1–30 in 87].

The most basic way of using a GPAC was in *single-shot mode* [pp. 168–170 in 89]. First, parameters and initial values were entered into the potentiometers. Next, putting a master switch in “reset” mode controlled relays to apply the initial values to the integrators. Turning the switch to “operate” or “compute” mode allowed the computation to take place (i.e., the integrators to integrate). Finally, placing the switch in “hold” mode stopped the computation and stabilized the values, allowing them to be read from the computer (e.g., on voltmeters). Although single-shot operation was also called “slow operation” (in comparison to “repetitive operation”, discussed next), it was in practice quite fast. Because all of the devices computed in parallel and at electronic speeds, analog computers usually solved problems in real-time but often much faster (Truitt and Rogers [pp. 1-30–32 in 87], Small [p. 72 in 82]).

One common application of GPACs was to explore the effect of one or more parameters on the behavior of a system. To facilitate this exploration of the parameter space, some GPACs provided a *repetitive operation mode*, which worked as follows (Weyrick [p. 170 in 89], Small [p. 72 in 82]). An electronic clock switched the computer between reset and compute modes at an adjustable rate (e.g., 10–1000 cycles per second) [p. 280, n. 1 in 2]. In effect the simulation was rerun at the clock rate, but if any parameters were adjusted, the simulation results would vary along with them. Therefore, within a few seconds, an entire family of related simulations could be run. More importantly, the operator could acquire an intuitive understanding of the system’s dependence on its parameters.

The Eclipse of Analog Computing A common view is that electronic analog computers were a primitive predecessor of the digital computer, and that their use was just a historical episode, or even a digression, in the inevitable triumph of digital technology. It is supposed that the current digital hegemony is a simple matter of technological superiority. However, the history is much more complicated, and involves a number of social, economic, historical, pedagogical, and also technical factors, which are outside the scope of this article (see Small [81] and Small [82], especially Chap. 8, for more information). In any case, beginning after World War II and continuing for twenty-five years, there was lively debate about the relative merits of analog and digital computation.

Speed was an oft-cited advantage of analog computers [Chap. 8 in 82]. While early digital computers were much faster than mechanical differential analyzers, they were slower (often by several orders of magnitude) than electronic analog computers. Furthermore, although digital computers could perform individual arithmetic operations rapidly, complete problems were solved sequentially, one operation at a time, whereas analog computers operated in parallel. Thus it was argued that increasingly large problems required more time to solve on a digital computer, whereas on an analog computer they might require more hardware but not more time. Even as digital computing speed was improved, analog computing retained its advantage for several decades, but this advantage eroded steadily.

Another important issue was the comparative *precision* of digital and analog computation [Chap. 8 in 82]. Analog computers typically computed with three or four digits of precision, and it was very expensive to do much better, due to the difficulty of manufacturing the parts and other factors. In contrast, digital computers could perform

arithmetic operations with many digits of precision, and the hardware cost was approximately proportional to the number of digits. Against this, analog computing advocates argued that many problems did not require such high precision, because the measurements were known to only a few significant figures and the mathematical models were approximations. Further, they distinguished between precision and *accuracy*, which refers to the conformity of the computation to physical reality, and they argued that digital computation was often less accurate than analog, due to numerical limitations (e.g., truncation, cumulative error in numerical integration). Nevertheless, some important applications, such as the calculation of missile trajectories, required greater precision, and for these, digital computation had the advantage. Indeed, to some extent precision was viewed as inherently desirable, even in applications where it was unimportant, and it was easily mistaken for accuracy. (See Sect. “Precision” for more on precision and accuracy.)

There was even a social factor involved, in that the written programs, precision, and exactness of digital computation were associated with mathematics and science, but the hands-on operation, parameter variation, and approximate solutions of analog computation were associated with engineers, and so analog computing inherited “the lower status of engineering *vis-à-vis* science” [p. 251 in 82]. Thus the status of digital computing was further enhanced as engineering became more mathematical and scientific after World War II [pp. 247–251 in 82].

Already by the mid-1950s the competition between analog and digital had evolved into the idea that they were complementary technologies. This resulted in the development of a variety of *hybrid* analog/digital computing systems [pp. 251–253, 263–266 in 82]. In some cases this involved using a digital computer to control an analog computer by using digital logic to connect the analog computing elements, set parameters, and gather data. This improved the accessibility and usability of analog computers, but had the disadvantage of distancing the user from the physical analog system. The intercontinental ballistic missile program in the USA stimulated the further development of hybrid computers in the late 1950s and 1960s [81]. These applications required the speed of analog computation to simulate the closed-loop control systems and the precision of digital computation for accurate computation of trajectories. However, by the early 1970s hybrids were being displaced by all digital systems. Certainly part of the reason was the steady improvement in digital technology, driven by a vibrant digital computer industry, but contemporaries also pointed to an inaccurate perception that

analog computing was obsolete and to a lack of education about the advantages and techniques of analog computing.

Another argument made in favor of digital computers was that they were general-purpose, since they could be used in business data processing and other application domains, whereas analog computers were essentially special-purpose, since they were limited to scientific computation [pp. 248–250 in 82]. Against this it was argued that *all* computing is essentially computing by analogy, and therefore analog computation was general-purpose because the class of analog computers included digital computers! (See also Sect. “Definition of the Subject” on computing by analogy.) Be that as it may, analog computation, as normally understood, is restricted to continuous variables, and so it was not immediately applicable to discrete data, such as that manipulated in business computing and other nonscientific applications. Therefore business (and eventually consumer) applications motivated the computer industry’s investment in digital computer technology at the expense of analog technology.

Although it is commonly believed that analog computers quickly disappeared after digital computers became available, this is inaccurate, for both general-purpose and special-purpose analog computers have continued to be used in specialized applications to the present time. For example, a general-purpose *electrical* (vs. electronic) analog computer, the Anacom, was still in use in 1991. This is not technological atavism, for “there is no doubt considerable truth in the fact that Anacom continued to be used because it effectively met a need in a historically neglected but nevertheless important computer application area” [3]. As mentioned, the reasons for the eclipse of analog computing were not simply the technological superiority of digital computation; the conditions were much more complex. Therefore a change in conditions has necessitated a reevaluation of analog technology.

Analog VLSI In the mid-1980s, Carver Mead, who already had made important contributions to digital VLSI technology, began to advocate for the development of analog VLSI [51,52]. His motivation was that “the nervous system of even a very simple animal contains computing paradigms that are orders of magnitude more effective than are those found in systems made by humans” and that they “can be realized in our most commonly available technology—silicon integrated circuits” [pp. xi in 52]. However, he argued, since these natural computation systems are analog and highly non-linear, progress would require understanding neural information processing in animals and applying it in a new analog VLSI technology.

Because analog computation is closer to the physical laws by which all computation is realized (which are continuous), analog circuits often use fewer devices than corresponding digital circuits. For example, a four-quadrant adder (capable of adding two signed numbers) can be fabricated from four transistors [pp. 87–88 in 52], and a four-quadrant multiplier from nine to seventeen, depending on the required range of operation [pp. 90–96 in 52]. Intuitions derived from digital logic about what is simple or complex to compute are often misleading when applied to analog computation. For example, two transistors are sufficient to compute the logarithm or exponential, five for the hyperbolic tangent (which is very useful in neural computation), and three for the square root [pp. 70–71, 97–99 in 52]. Thus analog VLSI is an attractive approach to “post-Moore’s Law computing” (see Sect. “[Future Directions](#)” below). Mead and his colleagues demonstrated a number of analog VLSI devices inspired by the nervous system, including a “silicon retina” and an “electronic cochlea” [Chaps. 15–16 in 52], research that has led to a renaissance of interest in electronic analog computing.

Non-Electronic Analog Computation As will be explained in the body of this article, analog computation suggests many opportunities for future computing technologies. Many physical phenomena are potential media for analog computation provided they have useful mathematical structure (i. e., the mathematical laws describing them are mathematical functions useful for general- or special-purpose computation), and they are sufficiently controllable for practical use.

Article Roadmap

The remainder of this article will begin by summarizing the fundamentals of analog computing, starting with the continuous state space and the various processes by which analog computation can be organized in time. Next it will discuss analog computation in nature, which provides models and inspiration for many contemporary uses of analog computation, such as neural networks. Then we consider general-purpose analog computing, both from a theoretical perspective and in terms of practical general-purpose analog computers. This leads to a discussion of the theoretical power of analog computation and in particular to the issue of whether analog computing is in some sense more powerful than digital computing. We briefly consider the cognitive aspects of analog computing, and whether it leads to a different approach to computation than does digital computing. Finally, we conclude with

some observations on the role of analog computation in “post-Moore’s Law computing”.

Fundamentals of Analog Computing

Continuous State Space

As discussed in Sect. “[Introduction](#)”, the fundamental characteristic that distinguishes analog from digital computation is that the state space is continuous in analog computation and discrete in digital computation. Therefore it might be more accurate to call analog and digital computation *continuous* and *discrete computation*, respectively. Furthermore, since the earliest days there have been *hybrid computers* that combine continuous and discrete state spaces and processes. Thus, there are several respects in which the state space may be continuous.

In the simplest case the state space comprises a finite (generally modest) number of variables, each holding a continuous quantity (e. g., voltage, current, charge). In a traditional GPAC they correspond to the variables in the ODEs defining the computational process, each typically having some independent meaning in the analysis of the problem. Mathematically, the variables are taken to contain bounded real numbers, although complex-valued variables are also possible (e. g., in AC electronic analog computers). In a practical sense, however, their precision is limited by noise, stability, device tolerance, and other factors (discussed below, Sect. “[Characteristics of Analog Computation](#)”).

In typical analog neural networks the state space is larger in dimension but more structured than in the former case. The artificial neurons are organized into one or more layers, each composed of a (possibly large) number of artificial neurons. Commonly each layer of neurons is densely connected to the next layer. In general the layers each have some meaning in the problem domain, but the individual neurons constituting them do not (and so, in mathematical descriptions, the neurons are typically numbered rather than named).

The individual artificial neurons usually perform a simple computation such as this:

$$y = \sigma(s), \quad \text{where } s = b + \sum_{i=1}^n w_i x_i,$$

and where y is the activity of the neuron, x_1, \dots, x_n are the activities of the neurons that provide its inputs, b is a bias term, and w_1, \dots, w_n are the *weights* or strengths of the connections. Often the *activation function* σ is a real-valued *sigmoid* (“S-shaped”) function, such as the *logistic*

sigmoid,

$$\sigma(s) = \frac{1}{1 + e^{-s}},$$

in which case the neuron activity y is a real number, but some applications use a discontinuous *threshold function*, such as the *Heaviside function*,

$$U(s) = \begin{cases} +1 & \text{if } s \geq 0 \\ 0 & \text{if } s < 0 \end{cases}$$

in which case the activity is a discrete quantity. The *saturated-linear* or *piecewise-linear* sigmoid is also used occasionally:

$$\sigma(s) = \begin{cases} +1 & \text{if } s > 1 \\ s & \text{if } 0 \leq s \leq 1 \\ 0 & \text{if } s < 0. \end{cases}$$

Regardless of whether the activation function is continuous or discrete, the bias b and connection weights w_1, \dots, w_n are real numbers, as is the “net input” $s = \sum_i w_i x_i$ to the activation function. Analog computation may be used to evaluate the linear combination s and the activation function $\sigma(s)$, if it is real-valued. The biases and weights are normally determined by a learning algorithm (e.g., back-propagation), which is also a good candidate for analog implementation.

In summary, the continuous state space of a neural network includes the bias values and net inputs of the neurons and the interconnection strengths between the neurons. It also includes the activity values of the neurons, if the activation function is a real-valued sigmoid function, as is often the case. Often large groups (“layers”) of neurons (and the connections between these groups) have some intuitive meaning in the problem domain, but typically the individual neuron activities, bias values, and interconnection weights do not.

If we extrapolate the number of neurons in a layer to the continuum limit, we get a *field*, which may be defined as a continuous distribution of continuous quantity (see ► [Field Computation in Natural and Artificial Intelligence](#)). Treating a group of artificial or biological neurons as a continuous mass is a reasonable mathematical approximation if their number is sufficiently large and if their spatial arrangement is significant (as it generally is in the brain). Fields are especially useful in modeling *cortical maps*, in which information is represented by the pattern of activity over a region of neural cortex.

In field computation the state space is continuous in two ways: it is continuous in variation but also in space. Therefore, field computation is especially applicable to

solving PDEs and to processing spatially extended information such as visual images. Some early analog computing devices were capable of field computation [pp. 1-14–17, 2-2–16 in 87]. For example, as previously mentioned (Sect. “[Introduction](#)”), large resistor and capacitor networks could be used for solving PDEs such as diffusion problems. In these cases a discrete ensemble of resistors and capacitors was used to approximate a continuous field, while in other cases the computing medium was spatially continuous. The latter made use of conductive sheets (for two-dimensional fields) or electrolytic tanks (for two- or three-dimensional fields). When they were applied to steady-state spatial problems, these analog computers were called *field plotters* or *potential analyzers*.

The ability to fabricate very large arrays of analog computing devices, combined with the need to exploit massive parallelism in realtime computation and control applications, creates new opportunities for field computation [37,38,43]. There is also renewed interest in using physical fields in analog computation. For example, Rubel [73] defined an abstract *extended analog computer* (EAC), which augments Shannon’s [77] general purpose analog computer with (unspecified) facilities for field computation, such as PDE solvers (see Sects. [Shannon’s Analysis – Rubel’s Extended Analog Computer](#) below). JW. Mills has explored the practical application of these ideas in his *artificial neural field networks* and VLSI EACs, which use the diffusion of electrons in bulk silicon or conductive gels and plastics for 2D and 3D field computation [53,54].

Computational Process

We have considered the continuous state space, which is the basis for analog computing, but there are a variety of ways in which analog computers can operate on the state. In particular, the state can change continuously in time or be updated at distinct instants (as in digital computation).

Continuous Time Since the laws of physics on which analog computing is based are differential equations, many analog computations proceed in continuous real time. Also, as we have seen, an important application of analog computers in the late 19th and early 20th centuries was the integration of ODEs in which time is the independent variable. A common technique in analog simulation of physical systems is *time scaling*, in which the differential equations are altered systematically so the simulation proceeds either more slowly or more quickly than the primary system (see Sect. “[Characteristics of Analog Computation](#)” for more on time scaling). On the other hand, because analog computations are close to the physical processes that

realize them, analog computing is rapid, which makes it very suitable for real-time control applications.

In principle, any mathematically describable physical process operating on time-varying physical quantities can be used for analog computation. In practice, however, analog computers typically provide familiar operations that scientists and engineers use in differential equations [70,87]. These include basic arithmetic operations, such as algebraic sum and difference ($u(t) = v(t) \pm w(t)$), constant multiplication or scaling ($u(t) = cv(t)$), variable multiplication and division ($u(t) = v(t)w(t)$, $u(t) = v(t)/w(t)$), and inversion ($u(t) = -v(t)$). Transcendental functions may be provided, such as the exponential ($u(t) = \exp v(t)$), logarithm ($u(t) = \ln v(t)$), trigonometric functions ($u(t) = \sin v(t)$, etc.), and *resolvers* for converting between polar and rectangular coordinates. Most important, of course, is definite integration ($u(t) = v_0 + \int_0^t v(\tau) d\tau$), but differentiation may also be provided ($u(t) = \dot{v}(t)$). Generally, however, direct differentiation is avoided, since noise tends to have a higher frequency than the signal, and therefore differentiation amplifies noise; typically problems are reformulated to avoid direct differentiation [pp. 26–27 in 89]. As previously mentioned, many GPACs include (*arbitrary*) *function generators*, which allow the use of functions defined only by a graph and for which no mathematical definition might be available; in this way empirically defined functions can be used [pp. 32–42 in 70]. Thus, given a graph ($x, f(x)$), or a sufficient set of samples, ($x_k, f(x_k)$), the function generator approximates $u(t) = f(v(t))$. Rather less common are generators for arbitrary functions of two variables, $u(t) = f(v(t), w(t))$, in which the function may be defined by a surface, ($x, y, f(x, y)$), or by sufficient samples from it.

Although analog computing is primarily continuous, there are situations in which discontinuous behavior is required. Therefore some analog computers provide *comparators*, which produce a discontinuous result depending on the relative value of two input values. For example,

$$u = \begin{cases} k & \text{if } v \geq w, \\ 0 & \text{if } v < w. \end{cases}$$

Typically, this would be implemented as a Heaviside (unit step) function applied to the difference of the inputs, $w = kU(v - w)$. In addition to allowing the definition of discontinuous functions, comparators provide a primitive decision making ability, and may be used, for example to terminate a computation (switching the computer from “operate” to “hold” mode).

Other operations that have proved useful in analog computation are time delays and noise generators [Chap. 7 in 31]. The function of a *time delay* is simply to retard the signal by an adjustable delay $T > 0$: $u(t + T) = v(t)$. One common application is to model delays in the primary system (e. g., human response time).

Typically a *noise generator* produces time-invariant Gaussian-distributed noise with zero mean and a flat power spectrum (over a band compatible with the analog computing process). The standard deviation can be adjusted by scaling, the mean can be shifted by addition, and the spectrum altered by filtering, as required by the application. Historically noise generators were used to model noise and other random effects in the primary system, to determine, for example, its sensitivity to effects such as turbulence. However, noise can make a positive contribution in some analog computing algorithms (e. g., for symmetry breaking and in simulated annealing, weight perturbation learning, and stochastic resonance).

As already mentioned, some analog computing devices for the direct solution of PDEs have been developed. In general a PDE solver depends on an analogous physical process, that is, on a process obeying the same class of PDEs that it is intended to solve. For example, in Mill’s EAC, diffusion of electrons in conductive sheets or solids is used to solve diffusion equations [53,54]. Historically, PDEs were solved on electronic GPACs by discretizing all but one of the independent variables, thus replacing the differential equations by difference equations [70], pp. 173–193. That is, computation over a field was approximated by computation over a finite real array.

Reaction-diffusion computation is an important example of continuous-time analog computing ► [Reaction-Diffusion Computing](#). The state is represented by a set of time-varying chemical concentration fields, c_1, \dots, c_n . These fields are distributed across a one-, two-, or three-dimensional space Ω , so that, for $\mathbf{x} \in \Omega$, $c_k(\mathbf{x}, t)$ represents the concentration of chemical k at location \mathbf{x} and time t . Computation proceeds in continuous time according to reaction-diffusion equations, which have the form:

$$\partial \mathbf{c} / \partial t = D \nabla^2 \mathbf{c} + \mathbf{F}(\mathbf{c}),$$

where $\mathbf{c} = (c_1, \dots, c_n)^T$ is the vector of concentrations, $D = \text{diag}(d_1, \dots, d_n)$ is a diagonal matrix of positive diffusion rates, and \mathbf{F} is nonlinear vector function that describes how the chemical reactions affect the concentrations.

Some neural net models operate in continuous time and thus are examples of continuous-time analog computation. For example, Grossberg [26,27,28] defines the activity of a neuron by differential equations such as this:

$$\dot{x}_i = -a_i x_i + \sum_{j=1}^n b_{ij} w_{ij}^{(+)} f_j(x_j) - \sum_{j=1}^n c_{ij} w_{ij}^{(-)} g_j(x_j) + I_i.$$

This describes the continuous change in the activity of neuron i resulting from passive decay (first term), positive feedback from other neurons (second term), negative feedback (third term), and input (last term). The f_j and g_j are nonlinear activation functions, and the $w_{ij}^{(+)}$ and $w_{ij}^{(-)}$ are adaptable excitatory and inhibitory connection strengths, respectively.

The continuous Hopfield network is another example of continuous-time analog computation [30]. The output y_i of a neuron is a nonlinear function of its internal state x_i , $y_i = \sigma(x_i)$, where the hyperbolic tangent is usually used as the activation function, $\sigma(x) = \tanh x$, because its range is $[-1, 1]$. The internal state is defined by a differential equation,

$$\tau_i \dot{x}_i = -a_i x_i + b_i + \sum_{j=1}^n w_{ij} y_j,$$

where τ_i is a time constant, a_i is the decay rate, b_i is the bias, and w_{ij} is the connection weight to neuron i from neuron j . In a Hopfield network every neuron is symmetrically connected to every other ($w_{ij} = w_{ji}$) but not to itself ($w_{ii} = 0$).

Of course analog VLSI implementations of neural networks also operate in continuous time (e. g., [20,52]).

Concurrent with the resurgence of interest in analog computation have been innovative reconceptualizations of continuous-time computation. For example, Brockett [12] has shown that dynamical systems can perform a number of problems normally considered to be intrinsically sequential. In particular, a certain system of ODEs (a *non-periodic finite Toda lattice*) can sort a list of numbers by continuous-time analog computation. The system is started with the vector \mathbf{x} equal to the values to be sorted and a vector \mathbf{y} initialized to small nonzero values; the \mathbf{y} vector converges to a sorted permutation of \mathbf{x} .

Sequential Time computation refers to computation in which discrete computational operations take place in succession but at no definite interval [88]. Ordinary digital computer programs take place in sequential time, for the operations occur one after another, but the individual operations are not required to have any specific duration, so long as they take finite time.

One of the oldest examples of sequential analog computation is provided by the compass-and-straightedge constructions of traditional Euclidean geometry (Sect. “[Introduction](#)”). These computations proceed by a sequence

of discrete operations, but the individual operations involve continuous representations (e. g., compass settings, straightedge positions) and operate on a continuous state (the figure under construction). Slide rule calculation might seem to be an example of sequential analog computation, but if we look at it, we see that although the operations are performed by an analog device, the intermediate results are recorded digitally (and so this part of the state space is discrete). Thus it is a kind of hybrid computation.

The familiar digital computer automates sequential digital computations that once were performed manually by human “computers”. Sequential analog computation can be similarly automated. That is, just as the control unit of an ordinary digital computer sequences digital computations, so a digital control unit can sequence analog computations. In addition to the analog computation devices (adders, multipliers, etc.), such a computer must provide variables and registers capable of holding continuous quantities between the sequential steps of the computation (see also Sect. “Discrete Time” below).

The primitive operations of sequential-time analog computation are typically similar to those in continuous-time computation (e. g., addition, multiplication, transcendental functions), but integration and differentiation with respect to sequential time do not make sense. However, continuous-time integration within a single step, and space-domain integration, as in PDE solvers or field computation devices, are compatible with sequential analog computation.

In general, any model of digital computation can be converted to a similar model of sequential analog computation by changing the discrete state space to a continuum, and making appropriate changes to the rest of the model. For example, we can make an analog Turing machine by allowing it to write a bounded real number (rather than a symbol from a finite alphabet) onto a tape cell. The Turing machine’s finite control can be altered to test for tape markings in some specified range.

Similarly, in a series of publications Blum, Shub, and Smale developed a theory of computation over the reals, which is an abstract model of sequential-time analog computation [6,7]. In this “BSS model” programs are represented as flowcharts, but they are able to operate on real-valued variables. Using this model they were able to prove a number of theorems about the complexity of sequential analog algorithms.

The BSS model, and some other sequential analog computation models, assume that it is possible to make exact comparisons between real numbers (analogous to exact comparisons between integers or discrete symbols in digital computation) and to use the result of the comparison

to control the path of execution. Comparisons of this kind are problematic because they imply infinite precision in the comparator (which may be defensible in a mathematical model but is impossible in physical analog devices), and because they make the execution path a discontinuous function of the state (whereas analog computation is usually continuous). Indeed, it has been argued that this is not “true” analog computation [p. 148 in 78].

Many artificial neural network models are examples of sequential-time analog computation. In a simple feed-forward neural network, an input vector is processed by the layers in order, as in a pipeline. That is, the output of layer n becomes the input of layer $n + 1$. Since the model does not make any assumptions about the amount of time it takes a vector to be processed by each layer and to propagate to the next, execution takes place in sequential time. Most *recurrent* neural networks, which have feedback, also operate in sequential time, since the activities of all the neurons are updated synchronously (that is, signals propagate through the layers, or back to earlier layers, in lock-step).

Many artificial neural-net learning algorithms are also sequential-time analog computations. For example, the back-propagation algorithm updates a network’s weights, moving sequentially backward through the layers.

In summary, the correctness of sequential time computation (analog or digital) depends on the *order* of operations, not on their *duration*, and similarly the efficiency of sequential computations is evaluated in terms of the *number* of operations, not on their *total duration*.

Discrete Time analog computation has similarities to both continuous-time and sequential analog computation. Like the latter, it proceeds by a sequence of discrete (analog) computation steps; like the former, these steps occur at a constant rate in real time (e.g., some “frame rate”). If the real-time rate is sufficient for the application, then discrete-time computation can approximate continuous-time computation (including integration and differentiation).

Some electronic GPACs implemented discrete-time analog computation by a modification of repetitive operation mode, called *iterative analog computation* [Chap. 9 in 2]. Recall (Sect. “Electronic Analog Computation in the 20th Century”) that in repetitive operation mode a clock rapidly switched the computer between reset and compute modes, thus repeating the same analog computation, but with different parameters (set by the operator). However, each repetition was independent of the others. Iterative operation was different in that analog values computed by one iteration could be used as initial values in the

next. This was accomplished by means of an analog memory circuit (based on an op amp) that sampled an analog value at the end of one compute cycle (effectively during hold mode) and used it to initialize an integrator during the following reset cycle. (A modified version of the memory circuit could be used to retain a value over several iterations.) Iterative computation was used for problems such as determining, by iterative search or refinement, the initial conditions that would lead to a desired state at a future time. Since the analog computations were iterated at a fixed clock rate, iterative operation is an example of discrete-time analog computation. However, the clock rate is not directly relevant in some applications (such as the iterative solution of boundary value problems), in which case iterative operation is better characterized as sequential analog computation.

The principal contemporary examples of discrete-time analog computing are in neural network applications to time-series analysis and (discrete-time) control. In each of these cases the input to the neural net is a sequence of discrete-time samples, which propagate through the net and generate discrete-time output signals. Many of these neural nets are recurrent, that is, values from later layers are fed back into earlier layers, which allows the net to remember information from one sample to the next.

Analog Computer Programs

The concept of a *program* is central to digital computing, both practically, for it is the means for programming general-purpose digital computers, and theoretically, for it defines the limits of what can be computed by a universal machine, such as a universal Turing machine. Therefore it is important to discuss means for describing or specifying analog computations.

Traditionally, analog computers were used to solve ODEs (and sometimes PDEs), and so in one sense a mathematical differential equation is one way to represent an analog computation. However, since the equations were usually not suitable for direct solution on an analog computer, the process of *programming* involved the translation of the equations into a schematic diagram showing how the analog computing devices (integrators etc.) should be connected to solve the problem. These diagrams are the closest analogies to digital computer programs and may be compared to flowcharts, which were once popular in digital computer programming. It is worth noting, however, that flowcharts (and ordinary computer programs) represent sequences among operations, whereas analog computing diagrams represent functional relationships among variables, and therefore a kind of parallel data flow.

Differential equations and schematic diagrams are suitable for continuous-time computation, but for sequential analog computation something more akin to a conventional digital program can be used. Thus, as previously discussed (Sect. “Sequential Time”), the BSS system uses flowcharts to describe sequential computations over the reals. Similarly, C. Moore [55] defines recursive functions over the reals by means of a notation similar to a programming language.

In principle any sort of analog computation might involve constants that are arbitrary real numbers, which therefore might not be expressible in finite form (e.g., as a finite string of digits). Although this is of theoretical interest (see Sect. “Real-valued Inputs, Outputs, and Constants” below), from a practical standpoint these constants could be set with about at most four digits of precision [p. 11 in 70]. Indeed, automatic potentiometer-setting devices were constructed that read a series of decimal numerals from punched paper tape and used them to set the potentiometers for the constants [pp. 3–58–60 in 87]. Nevertheless it is worth observing that analog computers do allow continuous inputs that need not be expressed in digital notation, for example, when the parameters of a simulation are continuously varied by the operator. In principle, therefore, an analog program can incorporate constants that are represented by a real-valued physical quantity (e.g., an angle or a distance), which need not be expressed digitally. Further, as we have seen (Sect. “Electronic Analog Computation in the 20th Century”), some electronic analog computers could compute a function by means of an arbitrarily drawn curve, that is, not represented by an equation or a finite set of digitized points. Therefore, in the context of analog computing it is natural to expand the concept of a program beyond discrete symbols to include continuous representations (scalar magnitudes, vectors, curves, shapes, surfaces, etc.).

Typically such continuous representations would be used as adjuncts to conventional discrete representations of the analog computational process, such as equations or diagrams. However, in some cases the most natural static representation of the process is itself continuous, in which case it is more like a “guiding image” than a textual prescription [42]. A simple example is a potential surface, which defines a continuum of trajectories from initial states (possible inputs) to fixed-point attractors (the results of the computations). Such a “program” may define a deterministic computation (e.g., if the computation proceeds by gradient descent), or it may constrain a non-deterministic computation (e.g., if the computation may proceed by any potential-decreasing trajectory). Thus ana-

log computation suggests a broadened notion of programs and programming.

Characteristics of Analog Computation

Precision Analog computation is evaluated in terms of both accuracy and precision, but the two must be distinguished carefully [pp. 25–28 in 2, pp. 12–13 in 89, pp. 257–261 in 82]. *Accuracy* refers primarily to the relationship between a simulation and the primary system it is simulating or, more generally, to the relationship between the results of a computation and the mathematically correct result. Accuracy is a result of many factors, including the mathematical model chosen, the way it is set up on a computer, and the precision of the analog computing devices. *Precision*, therefore, is a narrower notion, which refers to the quality of a representation or computing device. In analog computing, precision depends on *resolution* (finesse of operation) and *stability* (absence of drift), and may be measured as a fraction of the represented value. Thus a precision of 0.01% means that the representation will stay within 0.01% of the represented value for a reasonable period of time. For purposes of comparing analog devices, the precision is usually expressed as a fraction of *full-scale variation* (i. e., the difference between the maximum and minimum representable values).

It is apparent that the precision of analog computing devices depends on many factors. One is the choice of physical process and the way it is utilized in the device. For example a linear mathematical operation can be realized by using a linear region of a nonlinear physical process, but the realization will be approximate and have some inherent imprecision. Also, associated, unavoidable physical effects (e.g., loading, and leakage and other losses) may prevent precise implementation of an intended mathematical function. Further, there are fundamental physical limitations to resolution (e.g., quantum effects, diffraction). Noise is inevitable, both intrinsic (e.g., thermal noise) and extrinsic (e.g., ambient radiation). Changes in ambient physical conditions, such as temperature, can affect the physical processes and decrease precision. At slower time scales, materials and components age and their physical characteristics change. In addition, there are always technical and economic limits to the control of components, materials, and processes in analog device fabrication.

The precision of analog and digital computing devices depend on very different factors. The precision of a (binary) digital device depends on the number of bits, which influences the amount of hardware, but not its quality. For example, a 64-bit adder is about twice the size of a 32-bit adder, but can be made out of the same components. At worst, the size of a digital device might increase with the

square of the number of bits of precision. This is because binary digital devices only need to represent two states, and therefore they can operate in saturation. The fabrication standards sufficient for the first bit of precision are also sufficient for the 64th bit. Analog devices, in contrast, need to be able to represent a continuum of states precisely. Therefore, the fabrication of high-precision analog devices is much more expensive than low-precision devices, since the quality of components, materials, and processes must be much more carefully controlled. Doubling the precision of an analog device may be expensive, whereas the cost of each additional bit of digital precision is incremental; that is, the cost is proportional to the logarithm of the precision expressed as a fraction of full range.

The forgoing considerations might seem to be a convincing argument for the superiority of digital to analog technology, and indeed they were an important factor in the competition between analog and digital computers in the middle of the twentieth century [pp. 257–261 in 82]. However, as was argued at that time, many computer applications do not require high precision. Indeed, in many engineering applications, the input data are known to only a few digits, and the equations may be approximate or derived from experiments. In these cases the very high precision of digital computation is unnecessary and may in fact be misleading (e. g., if one displays all 14 digits of a result that is accurate to only three). Furthermore, many applications in image processing and control do not require high precision. More recently, research in artificial neural networks (ANNs) has shown that low-precision analog computation is sufficient for almost all ANN applications. Indeed, neural information processing in the brain seems to operate with very low precision (perhaps less than 10% [p. 378 in 50]), for which it compensates with massive parallelism. For example, by *coarse coding* a population of low-precision devices can represent information with relatively high precision [pp. 91–96 in 74,75].

Scaling An important aspect of analog computing is *scaling*, which is used to adjust a problem to an analog computer. First is *time scaling*, which adjusts a problem to the characteristic time scale at which a computer operates, which is a consequence of its design and the physical processes by which it is realized [pp. 37–44 in 62, pp. 262–263 in 70, pp. 241–243 in 89]. For example, we might want a simulation to proceed on a very different time scale from the primary system. Thus a weather or economic simulation should proceed faster than real time in order to get useful predictions. Conversely, we might want to slow down a simulation of protein folding so that we can observe the stages in the process. Also, for accurate results

it is necessary to avoid exceeding the maximum response rate of the analog devices, which might dictate a slower simulation speed. On the other hand, too slow a computation might be inaccurate as a consequence of instability (e. g., drift and leakage in the integrators).

Time scaling affects only time-dependent operations such as integration. For example, suppose t , time in the primary system or “problem time”, is related to τ , time in the computer, by $\tau = \beta t$. Therefore, an integration $u(t) = \int_0^t v(t')dt'$ in the primary system is replaced by the integration $u(\tau) = \beta^{-1} \int_0^\tau v(\tau')d\tau'$ on the computer. Thus time scaling may be accomplished simply by decreasing the input gain to the integrator by a factor of β .

Fundamental to analog computation is the representation of a continuous quantity in the primary system by a continuous quantity in the computer. For example, a displacement x in meters might be represented by a potential V in volts. The two are related by an *amplitude* or *magnitude scale factor*, $V = \alpha x$, (with units volts/meter), chosen to meet two criteria [pp. 103–106 in 2, Chap. 4 in 62, pp. 127–128 in 70, pp. 233–240 in 89]. On the one hand, α must be sufficiently small so that the range of the problem variable is accommodated within the range of values supported by the computing device. Exceeding the device’s intended operating range may lead to inaccurate results (e. g., forcing a linear device into nonlinear behavior). On the other hand, the scale factor should not be too small, or relevant variation in the problem variable will be less than the resolution of the device, also leading to inaccuracy. (Recall that precision is specified as a fraction of full-range variation.)

In addition to the explicit variables of the primary system, there are implicit variables, such as the time derivatives of the explicit variables, and scale factors must be chosen for them too. For example, in addition to displacement x , a problem might include velocity \dot{x} and acceleration \ddot{x} . Therefore, scale factors α , α' , and α'' must be chosen so that αx , $\alpha' \dot{x}$, and $\alpha'' \ddot{x}$ have an appropriate range of variation (neither too large nor too small).

Once a scale factor has been chosen, the primary system equations are adjusted to obtain the analog computing equations. For example, if we have scaled $u = \alpha x$ and $v = \alpha' \dot{x}$, then the integration $x(t) = \int_0^t \dot{x}(t')dt'$ would be computed by scaled equation:

$$u(t) = \frac{\alpha}{\alpha'} \int_0^t v(t')dt'.$$

This is accomplished by simply setting the input gain of the integrator to α/α' .

In practice, time scaling and magnitude scaling are not independent [p. 262 in 70]. For example, if the derivatives

of a variable can be large, then the variable can change rapidly, and so it may be necessary to slow down the computation to avoid exceeding the high-frequency response of the computer. Conversely, small derivatives might require the computation to be run faster to avoid integrator leakage etc. Appropriate scale factors are determined by considering both the physics and the mathematics of the problem [pp. 40–44 in 62]. That is, first, the physics of the primary system may limit the ranges of the variables and their derivatives. Second, analysis of the mathematical equations describing the system can give additional information on the ranges of the variables. For example, in some cases the natural frequency of a system can be estimated from the coefficients of the differential equations; the maximum of the n th derivative is then estimated as the n power of this frequency [p. 42 in 62, pp. 238–240 in 89]. In any case, it is not necessary to have accurate values for the ranges; rough estimates giving orders of magnitude are adequate.

It is tempting to think of magnitude scaling as a problem unique to analog computing, but before the invention of floating-point numbers it was also necessary in digital computer programming. In any case it is an essential aspect of analog computing, in which physical processes are more directly used for computation than they are in digital computing. Although the necessity of scaling has been a source of criticism, advocates for analog computing have argued that it is a blessing in disguise, because it leads to improved understanding of the primary system, which was often the goal of the computation in the first place [5, Chap. 8 in 82]. Practitioners of analog computing are more likely to have an intuitive understanding of both the primary system and its mathematical description (see Sect. “Analog Thinking”).

Analog Computation in Nature

Computational processes—that is to say, information processing and control—occur in many living systems, most obviously in nervous systems, but also in the self-organized behavior of groups of organisms. In most cases natural computation is analog, either because it makes use of continuous natural processes, or because it makes use of discrete but stochastic processes. Several examples will be considered briefly.

Neural Computation

In the past neurons were thought of binary computing devices, something like digital logic gates. This was a consequence of the “all or nothing” response of a neuron, which refers to the fact that it does or does not generate an *ac-*

tion potential (voltage spike) depending, respectively, on whether its total input exceeds a threshold or not (more accurately, it generates an action potential if the membrane depolarization at the axon hillock exceeds the threshold and the neuron is not in its refractory period). Certainly some neurons (e. g., so-called “command neurons”) do act something like logic gates. However, most neurons are analyzed better as analog devices, because the *rate* of impulse generation represents significant information. In particular, an *amplitude code*, the membrane potential near the axon hillock (which is a summation of the electrical influences on the neuron), is translated into a *rate code* for more reliable long-distance transmission along the axons. Nevertheless, the code is low precision (about one digit), since information theory shows that it takes at least N milliseconds (and probably more like 5N ms) to discriminate N values [39]. The rate code is translated back to an amplitude code by the synapses, since successive impulses release neurotransmitter from the axon terminal, which diffuses across the synaptic cleft to receptors. Thus a synapse acts as a leaky integrator to time-average the impulses.

As previously discussed (Sect. “Continuous State Space”), many artificial neural net models have real-valued neural activities, which correspond to rate-encoded axonal signals of biological neurons. On the other hand, these models typically treat the input connections as simple real-valued weights, which ignores the analog signal processing that takes place in the dendritic trees of biological neurons. The dendritic trees of many neurons are complex structures, which often have thousand of synaptic inputs. The binding of neurotransmitters to receptors causes minute voltage fluctuations, which propagate along the membrane, and ultimately cause voltage fluctuations at the axon hillock, which influence the impulse rate. Since the dendrites have both resistance and capacitance, to a first approximation the signal propagation is described by the “cable equations”, which describe passive signal propagation in cables of specified diameter, capacitance, and resistance [Chap. 1 in 1]. Therefore, to a first approximation, a neuron’s dendritic net operates as an adaptive linear analog filter with thousands of inputs, and so it is capable of quite complex signal processing. More accurately, however, it must be treated as a *nonlinear* analog filter, since voltage-gated ion channels introduce nonlinear effects. The extent of analog signal processing in dendritic trees is still poorly understood.

In most cases, then, neural information processing is treated best as low-precision analog computation. Although individual neurons have quite broadly tuned responses, accuracy in perception and sensorimotor control is achieved through coarse coding, as already discussed

(Sect. “[Characteristics of Analog Computation](#)”). Further, one widely used neural representation is the *cortical map*, in which neurons are systematically arranged in accord with one or more dimensions of their stimulus space, so that stimuli are represented by patterns of activity over the map. (Examples are *tonotopic maps*, in which pitch is mapped to cortical location, and *retinotopic maps*, in which cortical location represents retinal location.) Since neural density in the cortex is at least 146 000 neurons per square millimeter [p. 51 in [14](#)], even relatively small cortical maps can be treated as fields and information processing in them as analog field computation (see ► [Field Computation in Natural and Artificial Intelligence](#)). Overall, the brain demonstrates what can be accomplished by massively parallel analog computation, even if the individual devices are comparatively slow and of low precision.

Adaptive Self-Organization in Social Insects

Another example of analog computation in nature is provided by the self-organizing behavior of social insects, microorganisms, and other populations [[13](#)]. Often such organisms respond to concentrations, or gradients in the concentrations, of chemicals produced by other members of the population. These chemicals may be deposited and diffuse through the environment. In other cases, insects and other organisms communicate by contact, but may maintain estimates of the relative proportions of different kinds of contacts. Because the quantities are effectively continuous, all these are examples of analog control and computation.

Self-organizing populations provide many informative examples of the use of natural processes for analog information processing and control. For example, diffusion of pheromones is a common means of self-organization in insect colonies, facilitating the creation of paths to resources, the construction of nests, and many other functions [[13](#)]. Real diffusion (as opposed to sequential simulations of it) executes, in effect, a massively parallel search of paths from the chemical’s source to its recipients and allows the identification of near-optimal paths. Furthermore, if the chemical degrades, as is generally the case, then the system will be adaptive, in effect continually searching out the shortest paths, so long as source continues to function [[13](#)]. Simulated diffusion has been applied to robot path planning [[32,69](#)].

Genetic Circuits

Another example of natural analog computing is provided by the *genetic regulatory networks* that control the behavior of cells, in multicellular organisms as well as single-celled

ones [[16](#)]. These networks are defined by the mutually interdependent regulatory genes, promoters, and repressors that control the internal and external behavior of a cell. The interdependencies are mediated by proteins, the synthesis of which is governed by genes, and which in turn regulate the synthesis of other gene products (or themselves). Since it is the quantities of these substances that is relevant, many of the regulatory motifs can be described in computational terms as adders, subtracters, integrators, etc. Thus the genetic regulatory network implements an analog control system for the cell [[68](#)].

It might be argued that the number of intracellular molecules of a particular protein is a (relatively small) discrete number, and therefore that it is inaccurate to treat it as a continuous quantity. However, the molecular processes in the cell are stochastic, and so the relevant quantity is the *probability* that a regulatory protein will bind to a regulatory site. Further, the processes take place in continuous real time, and so the rates are generally the significant quantities. Finally, although in some cases gene activity is either on or off (more accurately: very low), in other cases it varies continuously between these extremes [pp. 388–390 in [29](#)].

Embryological development combines the analog control of individual cells with the sort of self-organization of populations seen in social insects and other colonial organisms. Locomotion of the cells and the expression of specific genes is controlled by chemical signals, among other mechanisms [[16,17](#)]. Thus PDEs have proved useful in explaining some aspects of development; for example *reaction-diffusion equations* have been used to describe the formation of hair-coat patterns and related phenomena [[13,48,57](#)]; see ► [Reaction-Diffusion Computing](#). Therefore the developmental process is governed by naturally occurring analog computation.

Is Everything a Computer?

It might seem that any continuous physical process could be viewed as analog computation, which would make the term almost meaningless. As the question has been put, is it meaningful (or useful) to say that the solar system is *computing* Kepler’s laws? In fact, it is possible and worthwhile to make a distinction between computation and other physical processes that happen to be described by mathematical laws [[40,41,44,46](#)].

If we recall the original meaning of analog computation (Sect. “[Definition of the Subject](#)”), we see that the computational system is used to solve some mathematical problem with respect to a primary system. What makes this possible is that the computational system and the pri-

mary system have the same, or systematically related, abstract (mathematical) structures. Thus the computational system can inform us about the primary system, or be used to control it, etc. Although from a practical standpoint some analogs are better than others, in principle any physical system can be used that obeys the same equations as the primary system.

Based on these considerations we may define computation as a physical process the purpose of which is the abstract manipulation of abstract objects (i. e., information processing); this definition applies to analog, digital, and hybrid computation [40,41,44,46]. Therefore, to determine if a natural system is computational we need to look to its purpose or function within the context of the living system of which it is a part. One test of whether its function is the abstract manipulation of abstract objects is to ask whether it could still fulfill its function if realized by different physical processes, a property called *multiple realizability*. (Similarly, in artificial systems, a simulation of the economy might be realized equally accurately by a hydraulic analog computer or an electronic analog computer [5].) By this standard, the majority of the nervous system is purely computational; in principle it could be replaced by electronic devices obeying the same differential equations. In the other cases we have considered (self-organization of living populations, genetic circuits) there are instances of both pure computation and computation mixed with other functions (for example, where the specific substances used have other – e. g. metabolic – roles in the living system).

General-Purpose Analog Computation

The Importance of General-Purpose Computers

Although special-purpose analog and digital computers have been developed, and continue to be developed, for many purposes, the importance of general-purpose computers, which can be adapted easily for a wide variety of purposes, has been recognized since at least the nineteenth century. Babbage's plans for a general-purpose digital computer, his *analytical engine* (1835), are well known, but a general-purpose differential analyzer was advocated by Kelvin [84]. Practical general-purpose analog and digital computers were first developed at about the same time: from the early 1930s through the war years. General-purpose computers of both kinds permit the prototyping of special-purpose computers and, more importantly, permit the flexible reuse of computer hardware for different or evolving purposes.

The concept of a general-purpose computer is useful also for determining the limits of a computing paradigm.

If one can design—theoretically or practically—a *universal computer*, that is, a general-purpose computer capable of simulating any computer in a relevant class, then anything uncomputable by the universal computer will also be uncomputable by any computer in that class. This is, of course, the approach used to show that certain functions are uncomputable by any Turing machine because they are uncomputable by a universal Turing machine. For the same reason, the concept of general-purpose analog computers, and in particular of *universal analog computers* are theoretically important for establishing limits to analog computation.

General-Purpose Electronic Analog Computers

Before taking up these theoretical issues, it is worth recalling that a typical electronic GPAC would include linear elements, such as adders, subtractors, constant multipliers, integrators, and differentiators; nonlinear elements, such as variable multipliers and function generators; other computational elements, such as comparators, noise generators, and delay elements (Sect. “Electronic Analog Computation in the 20th Century”). These are, of course, in addition to input/output devices, which would not affect its computational abilities.

Shannon's Analysis

Claude Shannon did an important analysis of the computational capabilities of the differential analyzer, which applies to many GPACs [76,77]. He considered an abstract differential analyzer equipped with an unlimited number of integrators, adders, constant multipliers, and function generators (for functions with only a finite number of finite discontinuities), with at most one source of drive (which limits possible interconnections between units). This was based on prior work that had shown that almost all the generally used elementary functions could be generated with addition and integration. We will summarize informally a few of Shannon's results; for details, please consult the original paper.

First Shannon offers proofs that, by setting up the correct ODEs, a GPAC with the mentioned facilities can generate any function if and only if it is not hypertranscendental (Theorem II); thus the GPAC can generate any function that is algebraic transcendental (a very large class), but not, for example, Euler's gamma function and Riemann's zeta function. He also shows that the GPAC can generate functions derived from generable functions, such as the integrals, derivatives, inverses, and compositions of generable functions (Thms. III, IV). These results can be generalized to functions of any number of variables, and to their com-

positions, partial derivatives, and inverses with respect to any one variable (Thms. VI, VII, IX, X).

Next Shannon shows that a function of any number of variables that is continuous over a closed region of space can be approximated arbitrarily closely over that region with a finite number of adders and integrators (Thms. V, VIII).

Shannon then turns from the generation of functions to the solution of ODEs and shows that the GPAC can solve any system of ODEs defined in terms of non-hypertranscendental functions (Thm. XI).

Finally, Shannon addresses a question that might seem of limited interest, but turns out to be relevant to the computational power of analog computers (see Sect. “[Analog Computation and the Turing Limit](#)” below). To understand it we must recall that he was investigating the differential analyzer—a mechanical analog computer—but similar issues arise in other analog computing technologies. The question is whether it is possible to perform an arbitrary constant multiplication, $u = kv$, by means of gear ratios. He shows that if we have just two gear ratios a and b ($a, b \neq 0, 1$), such that b is not a rational power of a , then by combinations of these gears we can approximate k arbitrarily closely (Thm. XII). That is, to approximate multiplication by arbitrary real numbers, it is sufficient to be able to multiply by a , b , and their inverses, provided a and b are not related by a rational power.

Shannon mentions an alternative method of constant multiplication, which uses integration, $kv = \int_0^v k dv$, but this requires setting the integrand to the constant function k . Therefore, multiplying by an arbitrary real number requires the ability to input an arbitrary real as the integrand. The issue of real-valued inputs and outputs to analog computers is relevant both to their theoretical power and to practical matters of their application (see Sect. “[Real-valued Inputs, Output, and Constants](#)”).

Shannon’s proofs, which were incomplete, were eventually refined by M. Pour-El [63] and finally corrected by L. Lipshitz and L.A. Rubel [35]. Rubel [72] proved that Shannon’s GPAC cannot solve the Dirichlet problem for Laplace’s equation on the disk; indeed, it is limited to initial-value problems for algebraic ODEs. Specifically, the *Shannon–Pour-El Thesis* is that the outputs of the GPAC are exactly the solutions of the *algebraic differential equations*, that is, equations of the form

$$P[x, y(x), y'(x), y''(x), \dots, y^{(n)}(x)] = 0,$$

where P is a polynomial that is not identically vanishing in any of its variables (these are the *differentially algebraic functions*) [71]. (For details please consult the cited

papers.) The limitations of Shannon’s GPAC motivated Rubel’s definition of the Extended Analog Computer.

Rubel’s Extended Analog Computer

The combination of Rubel’s [71] conviction that the brain is an analog computer together with the limitations of Shannon’s GPAC led him to propose the *Extended Analog Computer* (EAC) [73].

Like Shannon’s GPAC (and the Turing machine), the EAC is a conceptual computer intended to facilitate theoretical investigation of the limits of a class of computers. The EAC extends the GPAC in a number of respects. For example, whereas the GPAC solves equations defined over a single variable (time), the EAC can generate functions over any finite number of real variables. Further, whereas the GPAC is restricted to initial-value problems for ODEs, the EAC solves both initial- and boundary-value problems for a variety of PDEs.

The EAC is structured into a series of levels, each more powerful than the ones below it, from which it accepts inputs. The inputs to the lowest level are a finite number of real variables (“settings”). At this level it operates on real polynomials, from which it is able to generate the differentially algebraic functions. The computing on each level is accomplished by conceptual analog devices, which include constant real-number generators, adders, multipliers, differentiators, “substituters” (for function composition), devices for analytic continuation, and inverters, which solve systems of equations defined over functions generated by the lower levels. Most characteristic of the EAC is the “boundary-value-problem box”, which solves systems of PDEs and ODEs subject to boundary conditions and other constraints. The PDEs are defined in terms of functions generated by the lower levels. Such PDE solvers may seem implausible, and so it is important to recall field-computing devices for this purpose were implemented in some practical analog computers (see Sect. “[History](#)”) and more recently in Mills’ EAC [54]. As Rubel observed, PDE solvers could be implemented by physical processes that obey the same PDEs (heat equation, wave equation, etc.). (See also Sect. “[Future Directions](#)” below.)

Finally, the EAC is required to be “extremely well-posed”, which means that each level is relatively insensitive to perturbations in its inputs; thus “all the outputs depend in a strongly deterministic and stable way on the initial settings of the machine” [73].

Rubel [73] proves that the EAC can compute everything that the GPAC can compute, but also such functions as the gamma and zeta, and that it can solve the Dirichlet problem for Laplace’s equation on the disk, all of which

are beyond the GPAC's capabilities. Further, whereas the GPAC can compute differentially algebraic functions of time, the EAC can compute differentially algebraic functions of any finite number of real variables. In fact, Rubel did not find any real-analytic (C^∞) function that is *not* computable on the EAC, but he observes that if the EAC can indeed generate every real-analytic function, it would be too broad to be useful as a model of analog computation.

Analog Computation and the Turing Limit

Introduction

The Church–Turing Thesis asserts that anything that is effectively computable is computable by a Turing machine, but the Turing machine (and equivalent models, such as the lambda calculus) are models of discrete computation, and so it is natural to wonder how analog computing compares in power, and in particular whether it can compute beyond the “Turing limit”. Superficial answers are easy to obtain, but the issue is subtle because it depends upon choices among definitions, none of which is obviously correct, it involves the foundations of mathematics and its philosophy, and it raises epistemological issues about the role of models in scientific theories. Nevertheless this is an active research area, but many of the results are apparently inconsistent due to the differing assumptions on which they are based. Therefore this section will be limited to a mention of a few of the interesting results, but without attempting a comprehensive, systematic, or detailed survey; Siegelmann [78] can serve as an introduction to the literature.

A Sampling of Theoretical Results

Continuous-Time Models P. Orponen's [59] 1997 survey of continuous-time computation theory is a good introduction to the literature as of that time; here we give a sample of these and more recent results.

There are several results showing that—under various assumptions—analog computers have at least the power of Turing machines (TMs). For example, M.S. Branicky [11] showed that a TM could be simulated by ODEs, but he used non-differentiable functions; O. Bournez et al. [8] provide an alternative construction using only analytic functions. They also prove that the GPAC computability coincides with (Turing-)computable analysis, which is surprising, since the gamma function is Turing-computable but, as we have seen, the GPAC cannot generate it. The paradox is resolved by a distinction between *generating* a function and *computing* it, with the latter,

broader notion permitting convergent computation of the function (that is, as $t \rightarrow \infty$). However, the computational power of general ODEs has not been determined in general [p. 149 in 78]. MB Pour-El and I Richards exhibit a Turing-computable ODE that does not have a Turing-computable solution [64,66]. M. Stannett [83] also defined a continuous-time analog computer that could solve the halting problem.

C. Moore [55] defines a class of continuous-time recursive functions over the reals, which includes a zero-finding operator μ . Functions can be classified into a hierarchy depending on the number of uses of μ , with the lowest level (no μ s) corresponding approximately to Shannon's GPAC. Higher levels can compute non-Turing-computable functions, such as the decision procedure for the halting problem, but he questions whether this result is relevant in the physical world, which is constrained by “noise, quantum effects, finite accuracy, and limited resources”. O. Bournez and M. Cosnard [9] have extended these results and shown that many dynamical systems have super-Turing power.

S. Omohundro [58] showed that a system of ten coupled nonlinear PDEs could simulate an arbitrary cellular automaton (see ► [Mathematical Basis of Cellular Automata, Introduction to](#)), which implies that PDEs have at least Turing power. Further, D. Wolpert and B.J. MacLennan [90,91] showed that any TM can be simulated by a field computer with linear dynamics, but the construction uses Dirac delta functions. Pour-El and Richards exhibit a wave equation in three-dimensional space with Turing-computable initial conditions, but for which the unique solution is Turing-uncomputable [65,66].

Sequential-Time Models We will mention a few of the results that have been obtained concerning the power of sequential-time analog computation.

Although the BSS model has been investigated extensively, its power has not been completely determined [6,7]. It is known to depend on whether just rational numbers or arbitrary real numbers are allowed in its programs [p. 148 in 78].

A *coupled map lattice* (CML) is a cellular automaton with real-valued states ► [Mathematical Basis of Cellular Automata, Introduction to](#); it is a sequential-time analog computer, which can be considered a discrete-space approximation to a simple sequential-time field computer. P. Orponen and M. Matamala [60] showed that a finite CML can simulate a universal Turing machine. However, since a CML can simulate a BSS program or a recurrent neural network (see Sect. “Recurrent Neural Networks” below), it actually has super-Turing power [p. 149 in 78].

Recurrent neural networks are some of the most important examples of sequential analog computers, and so the following section is devoted to them.

Recurrent Neural Networks With the renewed interest in neural networks in the mid-1980s, many investigators wondered if recurrent neural nets have super-Turing power. M. Garzon and S. Franklin showed that a sequential-time net with a countable infinity of neurons could exceed Turing power [21,23,24]. Indeed, Siegelmann and E.D. Sontag [80] showed that finite neural nets with real-valued weights have super-Turing power, but W. Maass and Sontag [36] showed that recurrent nets with Gaussian or similar noise had *sub*-Turing power, illustrating again the dependence on these results on assumptions about what is a reasonable mathematical idealization of analog computing.

For recent results on recurrent neural networks, we will restrict our attention to the work of Siegelmann [78], who addresses the computational power of these networks in terms of the classes of languages they can recognize. Without loss of generality the languages are restricted to sets of binary strings. A string to be tested is fed to the network one bit at a time, along with an input that indicates when the end of the input string has been reached. The network is said to *decide* whether the string is in the language if it correctly indicates whether it is in the set or not, after some finite number of sequential steps since input began.

Siegelmann shows that, if exponential time is allowed for recognition, finite recurrent neural networks with real-valued weights (and saturated-linear activation functions) can compute *all* languages, and thus they are more powerful than Turing machines. Similarly, stochastic networks with rational weights also have super-Turing power, although less power than the deterministic nets with real weights. (Specifically, they compute $P/POLY$ and BPP/\log^* respectively; see Siegelmann [Chaps. 4, 9 in 78] for details.) She further argues that these neural networks serve as a “standard model” of (sequential) analog computation (comparable to Turing machines in Church-Turing computation), and therefore that the limits and capabilities of these nets apply to sequential analog computation generally.

Siegelmann [p. 156 in 78] observes that the super-Turing power of recurrent neural networks is a consequence of their use of non-rational real-valued weights. In effect, a real number can contain an infinite number of bits of information. This raises the question of how the non-rational weights of a network can ever be set, since it is not possible to define a physical quantity with infinite

precision. However, although non-rational weights may not be able to be set from outside the network, they can be computed within the network by learning algorithms, which are analog computations. Thus, Siegelmann suggests, the fundamental distinction may be between *static computational models*, such as the Turing machine and its equivalents, and *dynamically evolving computational models*, which can tune continuously variable parameters and thereby achieve super-Turing power.

Dissipative Models Beyond the issue of the power of analog computing relative to the Turing limit, there are also questions of its relative efficiency. For example, could analog computing solve NP-hard problems in polynomial or even linear time? In traditional computational complexity theory, efficiency issues are addressed in terms of the asymptotic number of computation steps to compute a function as the size of the function’s input increases. One way to address corresponding issues in an analog context is by treating an analog computation as a *dissipative system*, which in this context means a system that decreases some quantity (analogous to energy) so that the system state converges to an *point attractor*. From this perspective, the initial state of the system incorporates the input to the computation, and the attractor represents its output. Therefore, HT Siegelmann, S Fishman, and A Ben-Hur have developed a complexity theory for dissipative systems, in both sequential and continuous time, which addresses the rate of convergence in terms of the underlying rates of the system [4,79]. The relation between dissipative complexity classes (e.g., P_d , NP_d) and corresponding classical complexity classes (P , NP) remains unclear [p. 151 in 78].

Real-Valued Inputs, Outputs, and Constants

A common argument, with relevance to the theoretical power of analog computation, is that an input to an analog computer must be determined by setting a dial to a number or by typing a number into digital-to-analog conversion device, and therefore that the input will be a rational number. The same argument applies to any internal constants in the analog computation. Similarly, it is argued, any output from an analog computer must be measured, and the accuracy of measurement is limited, so that the result will be a rational number. Therefore, it is claimed, real numbers are irrelevant to analog computing, since any practical analog computer computes a function from the rationals to the rationals, and can therefore be simulated by a Turing machine. (See related arguments by Martin Davis [18,19].)

There are a number of interrelated issues here, which may be considered briefly. First, the argument is couched in terms of the input or output of *digital representations*, and the numbers so represented are necessarily rational (more generally, computable). This seems natural enough when we think of an analog computer as a calculating device, and in fact many historical analog computers were used in this way and had digital inputs and outputs (since this is our most reliable way of recording and reproducing quantities).

However, in many analog *control systems*, the inputs and outputs are continuous physical quantities that vary continuously in time (also a continuous physical quantity); that is, according to current physical theory, these quantities are real numbers, which vary according to differential equations. It is worth recalling that physical quantities are neither rational nor irrational; they can be so classified only in comparison with each other or with respect to a unit, that is, only if they are measured and digitally represented. Furthermore, physical quantities are neither computable nor uncomputable (in a Church-Turing sense); these terms apply only to discrete representations of these quantities (i. e., to numerals or other digital representations).

Therefore, in accord with ordinary mathematical descriptions of physical processes, analog computations can be treated as having arbitrary real numbers (in some range) as inputs, outputs, or internal states; like other continuous processes, continuous-time analog computations pass through all the reals in some range, including non-Turing-computable reals. Paradoxically, however, these same physical processes can be simulated on digital computers.

The Issue of Simulation by Turing Machines and Digital Computers

Theoretical results about the computational power, relative to Turing machines, of neural networks and other analog models of computation raise difficult issues, some of which are epistemological rather than strictly technical. On the one hand, we have a series of theoretical results proving the super-Turing power of analog computation models of various kinds. On the other hand, we have the obvious fact that neural nets are routinely simulated on ordinary digital computers, which have at most the power of Turing machines. Furthermore, it is reasonable to suppose that any physical process that might be used to realize analog computation—and certainly the known processes—could be simulated on a digital computer, as is done routinely in computational science. This would seem

to be incontrovertible proof that analog computation is no more powerful than Turing machines. The crux of the paradox lies, of course, in the non-Turing-computable reals. These numbers are a familiar, accepted, and necessary part of standard mathematics, in which physical theory is formulated, but from the standpoint of Church-Turing (CT) computation they do not exist. This suggests that the paradox is not a contradiction, but reflects a divergence between the goals and assumptions of the two models of computation.

The Problem of Models of Computation

These issues may be put in context by recalling that the Church-Turing (CT) model of computation is in fact a *model*, and therefore that it has the limitations of all models. A model is a cognitive tool that improves our ability to understand some class of phenomena by preserving relevant characteristics of the phenomena while altering other, irrelevant (or less relevant) characteristics. For example, a *scale model* alters the size (taken to be irrelevant) while preserving shape and other characteristics. Often a model achieves its purposes by making *simplifying* or *idealizing assumptions*, which facilitate analysis or simulation of the system. For example, we may use a linear mathematical model of a physical process that is only approximately linear. For a model to be effective it must preserve characteristics and make simplifying assumptions that are appropriate to the domain of questions it is intended to answer, its *frame of relevance* [46]. If a model is applied to problems outside of its frame of relevance, then it may give answers that are misleading or incorrect, because they depend more on the simplifying assumptions than on the phenomena being modeled. Therefore we must be especially cautious applying a model outside of its frame of relevance, or even at the limits of its frame, where the simplifying assumptions become progressively less appropriate. The problem is aggravated by the fact that often the frame of relevance is not explicitly defined, but resides in a tacit background of practices and skills within some discipline.

Therefore, to determine the applicability of the CT model of computation to analog computing, we must consider the frame of relevance of the CT model. This is easiest if we recall the domain of issues and questions it was originally developed to address: issues of effective calculability and derivability in formalized mathematics. This frame of relevance determines many of the assumptions of the CT model, for example, that information is represented by finite discrete structures of symbols from a finite alphabet, that information processing proceeds by the application of definite formal rules at discrete instants of time, and

that a computational or derivational process must be completed in a finite number of these steps.¹ Many of these assumptions are incompatible with analog computing and with the frames of relevance of many models of analog computation.

Relevant Issues for Analog Computation

Analog computation is often used for control. Historically, analog computers were used in control systems and to simulate control systems, but contemporary analog VLSI is also frequently applied in control. Natural analog computation also frequently serves a control function, for example, sensorimotor control by the nervous system, genetic regulation in cells, and self-organized cooperation in insect colonies. Therefore, control systems provide one frame of relevance for models of analog computation.

In this frame of relevance real-time response is a critical issue, which models of analog computation, therefore, ought to be able to address. Thus it is necessary to be able to relate the speed and frequency response of analog computation to the rates of the physical processes by which the computation is realized. Traditional methods of algorithm analysis, which are based on sequential time and asymptotic behavior, are inadequate in this frame of relevance. On the one hand, the constants (time scale factors), which reflect the underlying rate of computation are absolutely critical (but ignored in asymptotic analysis); on the other hand, in control applications the asymptotic behavior of algorithm is generally irrelevant, since the inputs are typically fixed in size or of a limited range of sizes.

The CT model of computation is oriented around the idea that the purpose of a computation is to evaluate a mathematical function. Therefore the basic criterion of adequacy for a computation is *correctness*, that is, that given a precise representation of an input to the function, it will produce (after finitely many steps) a precise representation of the corresponding output of the function. In the context of natural computation and control, however, other criteria may be equally or even more relevant. For example, *robustness* is important: how well does the system respond in the presence of noise, uncertainty, imprecision, and error, which are unavoidable in real natural and artificial control systems, and how well does it respond to defects and damage, which arise in many natural and artificial contexts. Since the real world is unpredictable, *flexibility* is also important: how well does an artificial system respond to inputs for which it was not designed, and how well does a natural system behave in situations outside the

range of those to which it is evolutionarily adapted. Therefore, *adaptability* (through learning and other means) is another important issue in this frame of relevance.²

Transcending Turing Computability

Thus we see that many applications of analog computation raise different questions from those addressed by the CT model of computation; the most useful models of analog computing will have a different frame of relevance. In order to address traditional questions such as whether analog computers can compute “beyond the Turing limit”, or whether they can solve NP-hard problems in polynomial time, it is necessary to construct models of analog computation within the CT frame of relevance. Unfortunately, constructing such models requires making commitments about many issues (such as the representation of reals and the discretization of time), that may affect the answers to these questions, but are fundamentally unimportant in the frame of relevance of the most useful applications of the concept of analog computation. Therefore, being overly focused on traditional problems in the theory of computation (which was formulated for a different frame of relevance) may distract us from formulating models of analog computation that can address important issues in its own frame of relevance.

Analog Thinking

It will be worthwhile to say a few words about the *cognitive implications* of analog computing, which are a largely forgotten aspect of analog vs. digital debates of the late 20th century. For example, it was argued that analog computing provides a deeper intuitive understanding of a system than the alternatives do [5, Chap. 8 in 82]. On the one hand, analog computers afforded a means of understanding analytically intractable systems by means of “dynamic models”. By setting up an analog simulation, it was possible to vary the parameters and explore interactively the behavior of a dynamical system that could not be analyzed mathematically. Digital simulations, in contrast, were orders of magnitude slower and did not permit this kind of interactive investigation. (Performance has improved sufficiently in contemporary digital computers so that in many cases digital simulations can be used as dynamic models, sometimes with an interface that mimics an analog computer; see [5].)

Analog computing is also relevant to the cognitive distinction between *knowing how* (*procedural knowledge*)

¹See MacLennan [45,46] for a more detailed discussion of the frame of relevance of the CT model.

²See MacLennan [45,46] for a more detailed discussion of the frames of relevance of natural computation and control.

and *knowing that* (*factual knowledge*) [Chap. 8 in 82]. The latter (“know-that”) is more characteristic of scientific culture, which strives for generality and exactness, often by designing experiments that allow phenomena to be studied in isolation, whereas the former (“know-how”) is more characteristic of engineering culture; at least it was so through the first half of the twentieth century, before the development of “engineering science” and the widespread use of analytic techniques in engineering education and practice. Engineers were faced with analytically intractable systems, with inexact measurements, and with empirical relationships (characteristic curves, etc.), all of which made analog computers attractive for solving engineering problems. Furthermore, because analog computing made use of physical phenomena that were mathematically analogous to those in the primary system, the engineer’s intuition and understanding of one system could be transferred to the other. Some commentators have mourned the loss of hands-on intuitive understanding attendant on the increasingly scientific orientation of engineering education and the disappearance of analog computer [5,34,61,67].

I will mention one last cognitive issue relevant to the differences between analog and digital computing. As already discussed Sect. “[Characteristics of Analog Computation](#)”, it is generally agreed that it is less expensive to achieve high precision with digital technology than with analog technology. Of course, high precision may not be important, for example when the available data are inexact or in natural computation. Further, some advocates of analog computing argue that high precision digital result are often misleading [p. 261 in 82]. Precision does not imply accuracy, and the fact that an answer is displayed with 10 digits does not guarantee that it is accurate to 10 digits; in particular, engineering data may be known to only a few significant figures, and the accuracy of digital calculation may be limited by numerical problems. Therefore, on the one hand, users of digital computers might fall into the trap of trusting their apparently exact results, but users of modest-precision analog computers were more inclined to healthy skepticism about their computations. Or so it was claimed.

Future Directions

Certainly there are many purposes that are best served by digital technology; indeed there is a tendency nowadays to think that everything is done better digitally. Therefore it will be worthwhile to consider whether analog computation should have a role in future computing technologies. I will argue that the approaching end of *Moore’s Law* [56],

which has predicted exponential growth in digital logic densities, will encourage the development of new analog computing technologies.

Two avenues present themselves as ways toward greater computing power: faster individual computing elements and greater densities of computing elements. Greater density increases power by facilitating parallel computing, and by enabling greater computing power to be put into smaller packages. Other things being equal, the fewer the layers of implementation between the computational operations and the physical processes that realize them, that is to say, the more directly the physical processes implement the computations, the more quickly they will be able to proceed. Since most physical processes are continuous (defined by differential equations), analog computation is generally faster than digital. For example, we may compare analog addition, implemented directly by the additive combination of physical quantities, with the sequential process of digital addition. Similarly, other things being equal, the fewer physical devices required to implement a computational element, the greater will be the density of these elements. Therefore, in general, the closer the computational process is to the physical processes that realize it, the fewer devices will be required, and so the continuity of physical law suggests that analog computation has the potential for greater density than digital. For example, four transistors can realize analog addition, whereas many more are required for digital addition. Both considerations argue for an increasing role of analog computation in post-Moore’s Law computing.

From this broad perspective, there are many physical phenomena that are potentially usable for future analog computing technologies. We seek phenomena that can be described by well-known and useful mathematical functions (e. g., addition, multiplication, exponential, logarithm, convolution). These descriptions do not need to be exact for the phenomena to be useful in many applications, for which limited range and precision are adequate. Furthermore, in some applications speed is not an important criterion; for example, in some control applications, small size, low power, robustness, etc. may be more important than speed, so long as the computer responds quickly enough to accomplish the control task. Of course there are many other considerations in determining whether given physical phenomena can be used for practical analog computation in a given application [47]. These include stability, controllability, manufacturability, and the ease of interfacing with input and output transducers and other devices. Nevertheless, in the post-Moore’s Law world, we will have to be willing to consider all physical phenomena as potential computing technologies, and in many cases we

will find that analog computing is the most effective way to utilize them.

Natural computation provides many examples of effective analog computation realized by relatively slow, low-precision operations, often through massive parallelism. Therefore, post-Moore's Law computing has much to learn from the natural world.

Bibliography

Primary Literature

- Anderson JA (1995) *An Introduction to Neural Networks*. MIT Press, Cambridge
- Ashley JR (1963) *Introduction to Analog Computing*. Wiley, New York
- Aspray W (1993) Edwin L. Harder and the Anacom: Analog computing at Westinghouse. *IEEE Ann Hist of Comput* 15(2):35–52
- Ben-Hur A, Siegelmann HT, Fishman S (2002) A theory of complexity for continuous time systems. *J Complex* 18:51–86
- Bissell CC (2004) A great disappearing act: The electronic analogue computer. In: *IEEE Conference on the History of Electronics*, Bletchley, June 2004. pp 28–30
- Blum L, Cucker F, Shub M, Smale S (1998) *Complexity and Real Computation*. Springer, Berlin
- Blum L, Shub M, Smale S (1988) On a theory of computation and complexity over the real numbers: NP completeness, recursive functions and universal machines. *Bulletin Am Math Soc* 21:1–46
- Bournez O, Campagnolo ML, Graça DS, Hairny E. The General Purpose Analog Computer and computable analysis are two equivalent paradigms of analog computation. In: *Theory and Applications of Models of Computation (TAMC 2006)*. *Lectures Notes in Computer Science*, vol 3959. Springer, Berlin, pp 631–643
- Bournez O, Cosnard M (1996) On the computational power of dynamical systems and hybrid systems. *Theor Comput Sci* 168(2):417–59
- Bowles MD (1996) US technological enthusiasm and British technological skepticism in the age of the analog brain. *Ann Hist Comput* 18(4):5–15
- Branicky MS (1994) Analog computation with continuous ODEs. In: *Proceedings IEEE Workshop on Physics and Computation*, Dallas, pp 265–274
- Brockett RW (1988) Dynamical systems that sort lists, diagonalize matrices and solve linear programming problems. In: *Proceedings 27th IEEE Conference Decision and Control*, Austin, December 1988, pp 799–803
- Camazine S, Deneubourg J-L, Franks NR, Sneyd G, Theraulaz J, Bonabeau E (2001) *Self-organization in Biological Systems*. Princeton Univ. Press, New York
- Changeux J-P (1985) *Neuronal Man: The Biology of Mind* (trans: Garey LL). Oxford University Press, Oxford
- Clymer AB (1993) The mechanical analog computers of Hannibal Ford and William Newell. *IEEE Ann Hist Comput* 15(2): 19–34
- Davidson EH (2006) *The Regulatory Genome: Gene Regulatory Networks in Development and Evolution*. Academic Press, Amsterdam
- Davies JA (2005) *Mechanisms of Morphogenesis*. Elsevier, Amsterdam
- Davis M (2004) The myth of hypercomputation. In: Teuscher C (ed) *Alan Turing: Life and Legacy of a Great Thinker*. Springer, Berlin, pp 195–212
- Davis M (2006) Why there is no such discipline as hypercomputation. *Appl Math Comput* 178:4–7
- Fakhraie SM, Smith KC (1997) *VLSI-Compatible Implementation for Artificial Neural Networks*. Kluwer, Boston
- Franklin S, Garzon M (1990) Neural computability. In: Omidvar OM (ed) *Progress in Neural Networks*, vol 1. Ablex, Norwood, pp 127–145
- Freeth T, Bitsakis Y, Moussas X, Seiradakis JH, Tselikas A, Mangou H, Zafeiropoulou M, Hadland R, Bate D, Ramsey A, Allen M, Crawley A, Hockley P, Malzbender T, Gelb D, Ambrisco W, Edmunds MG (2006) Decoding the ancient Greek astronomical calculator known as the Antikythera mechanism. *Nature* 444:587–591
- Garzon M, Franklin S (1989) Neural computability ii (extended abstract). In: *Proceedings, IJCNN International Joint Conference on Neural Networks*, vol 1. Institute of Electrical and Electronic Engineers, New York, pp 631–637
- Garzon M, Franklin S (1990) Computation on graphs. In: Omidvar OM (ed) *Progress in Neural Networks*, vol 2. Ablex, Norwood
- Goldstine HH (1972) *The Computer from Pascal to von Neumann*. Princeton, Princeton
- Grossberg S (1967) Nonlinear difference-differential equations in prediction and learning theory. *Proc Nat Acad Sci USA* 58(4):1329–1334
- Grossberg S (1973) Contour enhancement, short term memory, and constancies in reverberating neural networks. *Stud Appl Math* 11:213–257
- Grossberg S (1976) Adaptive pattern classification and universal recoding: I. parallel development and coding of neural feature detectors. *Biol Cybern* 23:121–134
- Hartl DL (1994) *Genetics*, 3rd edn. Jones & Bartlett, Boston
- Hopfield JJ (1984) Neurons with graded response have collective computational properties like those of two-state neurons. *Proc Nat Acad Sci USA* 81:3088–92
- Howe RM (1961) *Design Fundamentals of Analog Computer Components*. Van Nostrand, Princeton
- Khatib O (1986) Real-time obstacle avoidance for manipulators and mobile robots. *Int J Robot Res* 5:90–99
- Kirchhoff G (1845) Ueber den Durchgang eines elektrischen Stromes durch eine Ebene, insbesondere durch eine kreisförmige. *Ann Phys Chem* 140(64)(4):497–514
- Lang GF (2000) Analog was *not* a computer trademark! Why would anyone write about analog computers in year 2000? *Sound Vib* 34(8):16–24
- Lipshitz L, Rubel LA (1987) A differentially algebraic replacement theorem. *Proc Am Math Soc* 99(2):367–72
- Maass W, Sontag ED (1999) Analog neural nets with Gaussian or other common noise distributions cannot recognize arbitrary regular languages. *Neural Comput* 11(3):771–782
- MacLennan BJ (1987) Technology-independent design of neurocomputers: The universal field computer. In: Caudill M, Butler C (eds) *Proceedings of the IEEE First International Conference on Neural Networks*, vol 3, IEEE Press, pp 39–49

38. MacLennan BJ (1990) Field computation: A theoretical framework for massively parallel analog computation, parts I–IV. Technical Report CS-90-100, Department of Computer Science, University of Tennessee, Knoxville. Available from <http://www.cs.utk.edu/~mclennan>. Accessed 20 May 2008
39. MacLennan BJ (1991) Gabor representations of spatiotemporal visual images. Technical Report CS-91-144, Department of Computer Science, University of Tennessee, Knoxville. Available from <http://www.cs.utk.edu/~mclennan>. Accessed 20 May 2008
40. MacLennan BJ (1994) Continuous computation and the emergence of the discrete. In: Pribram KH (ed) *Origins: Brain & Self-Organization*, Lawrence Erlbaum, Hillsdale, pp 121–151.
41. MacLennan BJ (1994) “Words lie in our way”. *Minds Mach* 4(4):421–437
42. MacLennan BJ (1995) Continuous formal systems: A unifying model in language and cognition. In: *Proceedings of the IEEE Workshop on Architectures for Semiotic Modeling and Situation Analysis in Large Complex Systems*, Monterey, August 1995. pp 161–172. Also available from <http://www.cs.utk.edu/~mclennan>. Accessed 20 May 2008
43. MacLennan BJ (1999) Field computation in natural and artificial intelligence. *Inf Sci* 119:73–89
44. MacLennan BJ (2001) Can differential equations compute? Technical Report UT-CS-01-459, Department of Computer Science, University of Tennessee, Knoxville. Available from <http://www.cs.utk.edu/~mclennan>. Accessed 20 May 2008
45. MacLennan BJ (2003) Transcending Turing computability. *Minds Mach* 13:3–22
46. MacLennan BJ (2004) Natural computation and non-Turing models of computation. *Theor Comput Sci* 317:115–145
47. MacLennan BJ (in press) Super-Turing or non-Turing? Extending the concept of computation. *Int J Unconv Comput*, in press
48. Maini PK, Othmer HG (eds) (2001) *Mathematical Models for Biological Pattern Formation*. Springer, New York
49. Maziarz EA, Greenwood T (1968) *Greek Mathematical Philosophy*. Frederick Ungar, New York
50. McClelland JL, Rumelhart DE, the PDP Research Group. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol 2: Psychological and Biological Models. MIT Press, Cambridge
51. Mead C (1987) Silicon models of neural computation. In: Caudill M, Butler C (eds) *Proceedings, IEEE First International Conference on Neural Networks*, vol I. IEEE Press, Piscataway, pp 91–106
52. Mead C (1989) *Analog VLSI and Neural Systems*. Addison-Wesley, Reading
53. Mills JW (1996) The continuous retina: Image processing with a single-sensor artificial neural field network. In: *Proceedings IEEE Conference on Neural Networks*. IEEE Press, Piscataway
54. Mills JW, Himebaugh B, Kopecky B, Parker M, Shue C, Weilemann C (2006) “Empty space” computes: The evolution of an unconventional supercomputer. In: *Proceedings of the 3rd Conference on Computing Frontiers*, New York, May 2006. ACM Press, pp 115–126
55. Moore C (1996) Recursion theory on the reals and continuous-time computation. *Theor Comput Sci* 162:23–44
56. Moore GE (1965) Cramming more components onto integrated circuits. *Electronics* 38(8):114–117
57. Murray JD (1977) *Lectures on Nonlinear Differential-Equation Models in Biology*. Oxford, Oxford
58. Omohundro S (1984) Modeling cellular automata with partial differential equations. *Physica D* 10:128–34, 1984.
59. Orponen P (1997) A survey of continuous-time computation theory. In: *Advances in Algorithms, Languages, and Complexity*, Kluwer, Dordrecht, pp 209–224
60. Orponen P, Matamala M (1996) Universal computation by finite two-dimensional coupled map lattices. In: *Proceedings, Physics and Computation 1996*, New England Complex Systems Institute, Cambridge, pp 243–7
61. Owens L (1986) Vannevar Bush and the differential analyzer: The text and context of an early computer. *Technol Culture* 27(1):63–95
62. Peterson GR (1967) *Basic Analog Computation*. Macmillan, New York
63. Pour-El MB (1974) Abstract computability and its relation to the general purpose analog computer (some connections between logic, differential equations and analog computers). *Trans Am Math Soc* 199:1–29
64. Pour-El MB, Richards I (1979) A computable ordinary differential equation which possesses no computable solution. *Ann Math Log* 17:61–90
65. Pour-El MB, Richards I (1981) The wave equation with computable initial data such that its unique solution is not computable. *Adv Math* 39:215–239
66. Pour-El MB, Richards I (1982) Noncomputability in models of physical phenomena. *Int J Theor Phys*, 21:553–555
67. Puchta S (1996) On the role of mathematics and mathematical knowledge in the invention of Vannevar Bush’s early analog computers. *IEEE Ann Hist Comput* 18(4):49–59
68. Reiner JM (1968) *The Organism as an Adaptive Control System*. Prentice-Hall, Englewood Cliffs
69. Rimon E, Koditschek DE (1989) The construction of analytic diffeomorphisms for exact robot navigation on star worlds. In: *Proceedings of the 1989 IEEE International Conference on Robotics and Automation*, Scottsdale AZ. IEEE Press, New York, pp 21–26
70. Rogers AE, Connolly TW (1960) *Analog Computation in Engineering Design*. McGraw-Hill, New York
71. Rubel LA (1985) The brain as an analog computer. *J Theor Neurobiol* 4:73–81
72. Rubel LA (1988) Some mathematical limitations of the general-purpose analog computer. *Adv Appl Math* 9:22–34
73. Rubel LA (1993) The extended analog computer. *Adv Appl Math* 14:39–50
74. Rumelhart DE, McClelland JL, the PDP Research Group (1986) *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol 1: Foundations. MIT Press, Cambridge
75. Sanger TD (1996) Probability density estimation for the interpretation of neural population codes. *J Neurophysiol* 76:2790–2793
76. Shannon CE (1941) Mathematical theory of the differential analyzer. *J Math Phys Mass Inst Technol* 20:337–354
77. Shannon CE (1993) Mathematical theory of the differential analyzer. In: Sloane NJA, Wyner AD (eds) *Claude Elwood Shannon: Collected Papers*. IEEE Press, New York, pp 496–513
78. Siegelmann HT (1999) *Neural Networks and Analog Computation: Beyond the Turing Limit*. Birkhäuser, Boston
79. Siegelmann HT, Ben-Hur A, Fishman S (1999) Computational complexity for continuous time dynamics. *Phys Rev Lett* 83(7):1463–6

80. Siegelmann HT, Sontag ED (1994) Analog computation via neural networks. *Theor Comput Sci* 131:331–360
81. Small JS (1993) General-purpose electronic analog computing. *IEEE Ann Hist Comput* 15(2):8–18
82. Small JS (2001) *The Analogue Alternative: The electronic analogue computer in Britain and the USA, 1930–1975*. Routledge, London & New York
83. Stannett M (1990) X-machines and the halting problem: Building a super-Turing machine. *Form Asp Comput* 2:331–341
84. Thomson W (Lord Kelvin) (1876) Mechanical integration of the general linear differential equation of any order with variable coefficients. *Proc Royal Soc* 24:271–275
85. Thomson W (Lord Kelvin) (1878) Harmonic analyzer. *Proc Royal Soc* 27:371–373
86. Thomson W (Lord Kelvin) (1938). The tides. In: *The Harvard Classics*, vol 30: Scientific Papers. Collier, New York, pp 274–307
87. Truitt TD, Rogers AE (1960) *Basics of Analog Computers*. John F. Rider, New York
88. van Gelder T (1997) Dynamics and cognition. In: Haugeland J (ed) *Mind Design II: Philosophy, Psychology and Artificial Intelligence*. MIT Press, Cambridge MA, revised & enlarged edition, Chap 16, pp 421–450
89. Weyrick RC (1969) *Fundamentals of Analog Computers*. Prentice-Hall, Englewood Cliffs
90. Wolpert DH (1991) A computationally universal field computer which is purely linear. Technical Report LA-UR-91-2937. Los Alamos National Laboratory, Los Alamos
91. Wolpert DH, MacLennan BJ (1993) A computationally universal field computer that is purely linear. Technical Report CS-93-206. Dept. of Computer Science, University of Tennessee, Knoxville

Books and Reviews

- Fifer S (1961) *Analog computation: Theory, techniques and applications*, 4 vols. McGraw-Hill, New York
- Bissell CC (2004) A great disappearing act: The electronic analogue computer. In: IEEE conference on the history of electronics, 28–30 Bletchley, June 2004
- Lipka J (1918) *Graphical and mechanical computation*. Wiley, New York
- Mead C (1989) *Analog VLSI and neural systems*. Addison-Wesley, Reading
- Siegelmann HT (1999) *Neural networks and analog computation: Beyond the Turing limit*. Birkhäuser, Boston
- Small JS (2001) *The analogue alternative: The electronic analogue computer in Britain and the USA, 1930–1975*. Routledge, London & New York
- Small JS (1993) General-purpose electronic analog computing: 1945–1965. *IEEE Ann Hist Comput* 15(2):8–18

Anisotropic Networks, Elastomers and Gels

EUGENE M. TERENTJEV
 Cavendish Laboratory, University of Cambridge,
 Cambridge, UK

Article Outline

Glossary
 Definition of the Subject
 Introduction
 Nematic Elastomers: Reversible Shape Change
 Nematic Elastomers: Soft Elasticity and Dynamics
 Swelling and Anisotropic Gels
 Cholesteric Elastomers: Photonics
 New Frontiers
 Bibliography

Glossary

Elastomer Nominally equivalent to “rubber”, this defines a weakly crosslinked network of polymer chains which retain high thermal mobility of the strands between crosslinks. In this case the entropic effects dominate the response of such networks. Entropic rubber elasticity arises when such a network percolates the whole macroscopic system and the crosslinks constrain it such that it remembers its equilibrium shape, and respond elastically to deformations. In this sense, an *elastomer* is contrasted to a polymer *glass*: The latter can refer to a network so densely crosslinked that the individual chain segments have no significant mobility – or a polymer system below its structural glass transition. The elastic modulus of elastomers is usually in the range 10–1000 kPa, the estimate arising from the thermal energy kT per network strand, while the glass modulus is usually 0.1–10 GPa, as in most solids.

Gel This word is used in many different contexts, in all cases referring to a soft object that is capable of retaining its shape in ambient conditions against, e.g., gravity. The latter condition distinguishes a gel from a “sol”, or a nominal liquid. The softness is a relative concept; usually it refers to the modulus at or below the “human” scale, around 1–100 kPa, but there are many examples of even softer gels. In this article, a *gel* is contrasted to an *elastomer*, referring to a crosslinked network of polymer chains which is swollen by a good solvent. In this case the effective modulus becomes very low, nevertheless the system remains elastic as long as the integrity of percolating network remains intact.

Quenched constraints The term *quenched* (as opposed to *annealed*) refers to objects or systems that are prevented from exploring their full phase or conformational space during thermal motion. The classical example of quenched system is a structural glass, where the mobility of elements is very limited. In elastomers and gels, network crosslinks are the quenched objects, constraining the polymer chains connecting to them.

The effect of randomly quenched constraints is profound in many physical systems, as the local thermodynamic equilibrium has to establish among the mobile (annealed) elements while observing the constraints, the random local realization of which is determined by external factors, such as the preparation history.

Liquid crystal This refers to a group of anisotropic phases with incomplete translational order (which would represent a crystalline lattice). Classical examples are: The *nematic* phase – a fluid with no translational order at all, but with an orientational order of anisotropic molecules, and *smectic* or *lamellar* phases, which in addition to orientational order also have a 1-dimensional density modulation (a stack of parallel layers). Liquid crystallinity is a state of spontaneous, equilibrium anisotropy (breaking of orientational symmetry), in contrast to anisotropy induced by external factors such as electric/magnetic field or mechanical deformation.

Shape memory Strictly, any elastic material “remembers” its equilibrium shape and returns to it after deformation. The term *shape-memory* was introduced to distinguish materials that could be made to preserve their deformed state, until a trigger (e. g. heating above a transition temperature) induces the return to the original equilibrium shape. In shape-memory alloys, this is due to the martensitic transition, in shape-memory polymers the deformed state can be fixed by a glass transition or partial crystallization. Liquid crystal elastomers and gels have a reversible, or *equilibrium* shape-memory in that the shape of the body is determined by the current state of order at any given temperature.

Definition of the Subject

Anisotropic (liquid crystalline) elastomers and gels bring together, as nowhere else, three important ideas: *Orientalional order* in amorphous soft materials, *responsive molecular shape* and *quenched topological constraints*. Acting together, they create many new physical phenomena that are briefly reviewed in this article. Classical liquid crystals are typically fluids of relatively stiff rod molecules with long range orientational order. Long polymer chains, with incorporated rigid anisotropic units can also form orientationally ordered liquid crystalline phases. By contrast with free rigid rods, these flexible chains change their average molecular shape, from isotropic spherical to ellipsoidal, when their component rods align. Linking the polymer chains together into a network fixes their relative topol-

ogy, and the polymer melt (or solution) becomes an elastic solid – an elastomer (or gel). Radically new properties arise from the ability to change average molecular shape of anisotropic polymer chains in the solid state.

In ordinary solids the deformations are created by relative movement of the same atoms (or molecules) that form the bonded lattice. Hence, when the deformation is small, the lattice symmetry is preserved and one obtains an ordinary elastic response (often anisotropic). There is a classical elastic response found in glasses as well (either isotropic or anisotropic), where in place of the crystalline lattice recording the preferred position of atoms, they are confined by constraints of local cages. Either way, the elements of the body “know” their positions and the system responds with an increase of elastic energy when these elements are displaced. Large deformations destroy the lattice (or cage) integrity and simply break the material.

In contrast, in elastomers and gels the macroscopic elastic response arises from the entropy change of chains on relative movement of their crosslinked end-points, which are relatively far apart. Monomers remain highly mobile and thus liquid-like. What happens to chain segments (monomer moieties) on a smaller length scale is a relatively independent matter and such weakly crosslinked network behaves as a liquid on length scales smaller than the end-point separation of strands. In particular, the liquid crystalline order can be established within these segments. The magnitude of this order can be altered and its director can rotate, in principle, independently of deformation of the crosslinking points. Such an internal degree of freedom within, and coupled to the elastic body provides additional local forces and torques, intricately connected in the overall macroscopic response of the body.

In the simplest and most straightforward case of nematic elastomers and gels, the uniaxial ellipsoidal average anisotropy of chain constituting the network leads to two key physical properties that make these materials so unique. If one can change the level of nematic (orientational) order, which is not difficult to achieve by e. g. changing temperature, changes at the molecular level induce a corresponding mechanical strains: A block of rubber can contract or elongate by a large amount on heating or cooling, respectively. This process is perfectly reversible. This leads to a group of possible applications in artificial muscles and actuators, which can be driven by any stimulus that affects the local nematic order: temperature, solvent intake, irradiation, etc.

It is also possible to rotate the nematic director axis and the rubber matrix independently, although in contrast to ordinary liquid crystals it costs energy to uni-

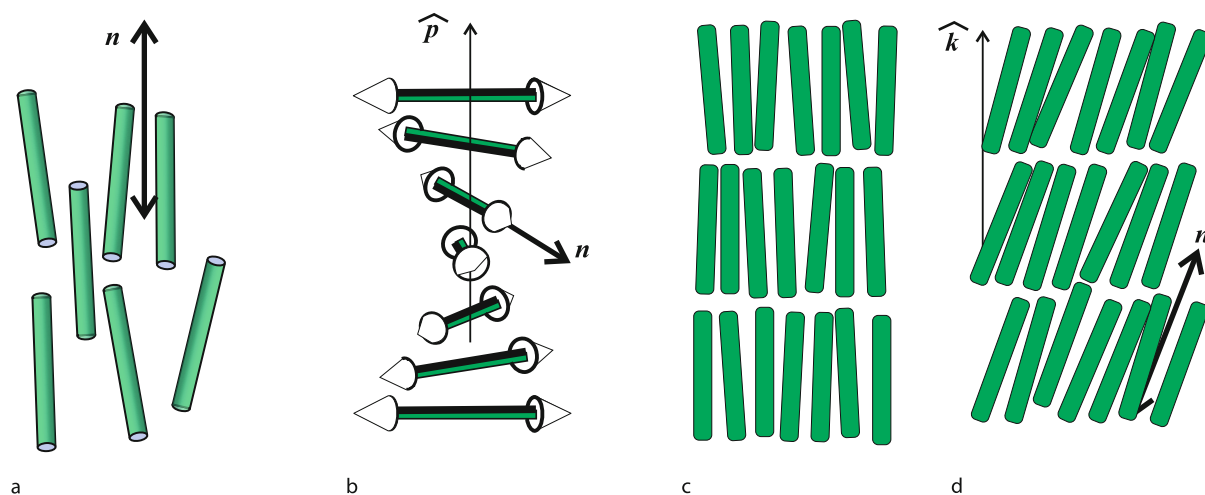
formly rotate the director within the matrix. This penalty leads to suppression of orientational fluctuations and high optical clarity of birefringent elastomers and gels. Local rotations also yield a subtle and spectacular elastic phenomenon which is called ‘soft elasticity’. Contrary to intuition, there is an infinite number of non-trivial mechanical deformations that can accommodate the rotation of anisotropic distribution of chains without its distortion. As a result, the entropy of the chains does not change, in spite of macroscopic deformations, and the material can be distorted without any significant energy cost! A special combination of shears and elongations/compressions is required, but it turns out not very difficult to achieve. Elastic softness, or attempts by the material to achieve it, pervade much of the elasticity of nematic elastomers and gels. For instance, another unique and spectacular property of these systems is anomalously high damping, where the mechanical energy is dissipated on these soft rotational deformation modes.

Cholesteric liquid crystals have a helical director texture. When crosslinked into elastomers or gels, this periodic helical anisotropy is made to couple to the mechanical state of the whole body. Their optical and mechanical responses to imposed stress are exceedingly rich as a result. Cholesteric elastomers are brightly colored due to selective reflection, and change color as they are stretched – their photonic band structure can be powerfully manipulated by applied deformations. Such periodic (distributed feedback) photonic systems can emit laser radiation, the color of which can be continuously shifted by mechanical means

across the whole visible spectrum. Further, the effect of topological imprinting can select and extract molecules of specific handedness from a mixed solvent. Cholesteric gels can act as a mechanical separator of chirality.

Smectic or lamellar elastomers and gels have plane-like, layered modulation of density in one direction (SmA), or additionally a tilt of the director away from the layer normal (SmC). Many other more complex smectic phases exist and could also be made into elastomers. Here the layers are often constrained by crosslinking density modulation not to move relative to the rubber matrix. Deformations along the layer normal can be resisted by a modulus up to 100 times greater than the normal rubber (shear) modulus of the matrix. Thus smectic elastomers are rubbery in the two dimensions of their layer planes, but respond as hard conventional solids in their third dimension. Such extreme mechanical anisotropy promises interesting applications.

The director tilt associated with the transition from SmA to SmC induces distortion in the polymer chain shape distribution. Since chain shape is coupled to mechanical shape for an elastomer with quite low symmetry, one expects a large variety of independent soft-elasticity modes. The tilted SmC systems also exist in chiral forms which must on symmetry grounds be ferroelectric, with spontaneous polarization pointing along the vector product $[\hat{k} \times \mathbf{n}]$, see Fig. 1d. Ferroelectric elastomer is very special: Mechanically it is soft, about 10^4 times lower in modulus than conventional solid ferro- and piezoelectrics. Distortions give polarization changes comparable to those



Anisotropic Networks, Elastomers and Gels, Figure 1

Arrangement of anisotropic (rod-like) molecular moieties in common liquid crystal phases: **a** nematic, **b** cholesteric, **c** smectic-A and **d** smectic-C. The average orientation of long axes is the director \mathbf{n} . The director, as well as the cholesteric helix axis \hat{p} and the smectic layer normal \hat{k} , are all directionless “quadrupolar” vectors, e.g. $\mathbf{n} = -\mathbf{n}$

in ordinary ferroelectrics, but the response to an applied stress must necessarily be much larger than in conventional materials due to the very low mechanical impedance.

It is debatable, whether an anisotropic glass could be called a liquid crystal, although nominally a glass can be made to possess orientational but not a translational order. The lack of molecular mobility makes the question of thermodynamic equilibrium very ambiguous.

Introduction

Within the simplest affine deformation approach one regards the change in each chain end-to-end vector \mathbf{R} as $\mathbf{R}' = \mathbf{F} \cdot \mathbf{R}$, when a deformation of the overall polymer network is characterized by a deformation gradient tensor F_{ij} . In continuum elasticity this tensor is often called $\mathbf{F} = \nabla \mathbf{f}$, a gradient of the geometrical mapping \mathbf{f} from the reference to the current state of deformation. Assuming the chain connecting the two crosslinks is long enough, the Gaussian approximation for the number of its conformations $Z(\mathbf{R})$ gives for the free energy (per chain): $W_{\text{ch}} = -kT \ln Z(\mathbf{R}') \simeq (kT/a^2N)[\mathbf{F}^T \cdot \mathbf{F}]_{ij} R_i R_j$, where a is the step length of the chain random walk and N the number of such steps. In order to find the total elastic work function of all chains affinely deforming in the network, one needs to add the contributions $W_{\text{ch}}(\mathbf{R})$ with the statistical weight to find a chain with a given initial end-to-end distance \mathbf{R} in the system. This procedure, called the quenched averaging, produces the average $\langle R_i R_j \rangle \simeq \frac{1}{3} a^2 N \delta_{ij}$ in W_{ch} . The resulting rubber-elastic free energy (per unit volume) is $W_{\text{el}} = \frac{1}{2} n_c kT (\mathbf{F}^T : \mathbf{F})$, with n_c a number of chains per unit volume of the network. This is a remarkably robust expression, with many seemingly relevant effects, such as the fluctuation of crosslinking points, only contributing a small quantitative change in the prefactor. It is, however, incomplete since the simplified Gaussian statistics of chains does not take into account their physical volume and thus does not address the compressibility issue. The proper way of dealing with it is by adding an additional independent penalty for the volume change: $\frac{1}{2} K (\det \mathbf{F} - 1)^2$ in elastomers, with K the typical bulk modulus of the order 10 GPa in organic materials. This should be modified into $\frac{1}{2} K (\det \mathbf{F} - \Phi)^2$ for gels swollen by a volume fraction Φ of solvent.

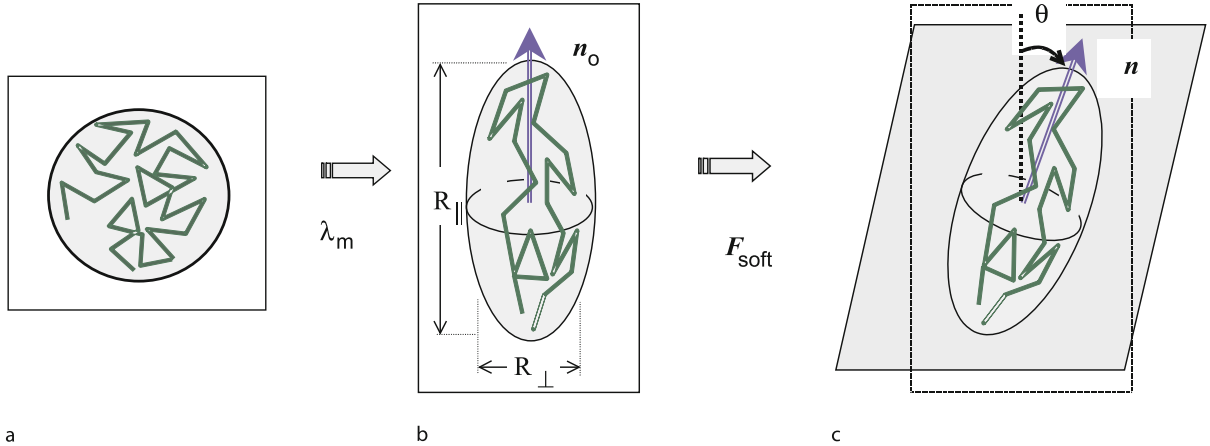
If one assumes that the rubber elastic modulus is low, then a very reasonable approximation is to simply impose the volume-conservation constraint on deformation tensor: $\det \mathbf{F} = 1$. The value of the rubber modulus is found on expanding the deformation gradient tensor in small strains, say for a small extension, $F_{zz} = 1 + \varepsilon$, and obtaining $W_{\text{el}} \simeq \frac{3}{2} n_c kT \varepsilon^2$. This means the exten-

sion (Young) modulus $E = 3n_c kT$; the analogous construction for a small simple shear will give the shear modulus $G = n_c kT$, exactly a third of the Young modulus as required in an incompressible medium. This shear modulus G , having its origin in the entropic effect of reduction of conformational freedom on polymer chain deformation, is usually so much smaller than the bulk modulus (determined by the enthalpy of compressing the dense polymer liquid) that the assumption of rubber or gel deforming at constant volume is justified. This constraint leads to the familiar rubber-elastic expression $W_{\text{el}} = \frac{1}{2} n_c kT (\lambda^2 + 2/\lambda)$ where one has assumed that the imposed extension $F_{zz} = \lambda$ is accompanied by the symmetric contraction in both transverse directions, $F_{xx} = F_{yy} = 1/\sqrt{\lambda}$ due to the incompressibility.

When the chains forming the rubbery network are liquid crystalline, their end-to-end distance distribution becomes anisotropic. The case of smectic/lamellar ordering is much more complicated. For a simple uniaxial nematic one obtains $\langle R_{\parallel} R_{\parallel} \rangle = \frac{1}{3} \ell_{\parallel} L$ and $\langle R_{\perp} R_{\perp} \rangle = \frac{1}{3} \ell_{\perp} L$, with $L = aN$ the chain contour length and $\ell_{\parallel}/\ell_{\perp}$ the ratio of average chain step lengths along and perpendicular to the nematic director. In the isotropic phase one recovers $\ell_{\parallel} = \ell_{\perp} = a$. The uniaxial anisotropy of polymer chains has a principal axis along the nematic director \mathbf{n} , with a prolate ($\ell_{\parallel}/\ell_{\perp} > 1$) or oblate ($\ell_{\parallel}/\ell_{\perp} < 1$) ellipsoidal conformation of polymer backbone. The ability of this principal axis to rotate independently under the influence of network strains makes the rubber elastic response non-symmetric, so that the elastic work function is [6]

$$W_{\text{el}} = \frac{1}{2} G \text{Tr} \left(\mathbf{F}^T \cdot \boldsymbol{\ell}^{-1} \cdot \mathbf{F} \cdot \boldsymbol{\ell}_0 \right) + \frac{1}{2} K (\det \mathbf{F} - 1)^2, \quad (1)$$

with $\boldsymbol{\ell}$ the uniaxial matrices of chain step-lengths before ($\boldsymbol{\ell}_0$) and after the deformation to the current state: $\ell_{ij} = \ell_{\perp} \delta_{ij} + [\ell_{\parallel} - \ell_{\perp}] n_i n_j$. Note that the deformation gradient tensor \mathbf{F} does no longer enter the elastic energy in the symmetric combination $\mathbf{F}^T \cdot \mathbf{F}$, but is “sandwiched” between the matrices $\boldsymbol{\ell}$ with different principal axes. This means that antisymmetric components of strain will now have a non-trivial physical effect, in contrast to isotropic rubbers and, more crucially, to elastic solids with uniaxial anisotropy. There, the anisotropy axis is immobile and the response is anisotropic but symmetric in stress and strain. The uniqueness of nematic rubbers stems from the competing microscopic interactions and the difference in characteristic length scales: The uniaxial anisotropy is established on a small (monomer) scale of nematic coherence length, while the strains are defined (and the elastic response is arising) on a much greater length scale of polymer chain end-to-end distance, see Fig. 2, [67,68].



Anisotropic Networks, Elastomers and Gels, Figure 2

Relation between the equilibrium chain shape and deformations in nematic elastomers. When the network of initially isotropic chains, each on average forming a spherical shape of gyration **a**, is brought into the nematic phase, a corresponding spontaneous deformation of the sample occurs in proportion to the backbone anisotropy, $\lambda_m = (\ell_{\parallel} / \ell_{\perp})^{-1/3}$, **b**. An example of soft deformation is given in **c**, when rotating anisotropic chains can affinely accommodate all strains (a combination of compression along the initial director, extension across it and a shear in the plane of director rotation), not causing any entropic rubber-elastic response

Nematic Elastomers: Reversible Shape Change

The “Trace formula” (1) has proven very successful in describing many physical properties of nematic and smectic elastomers and gels. One of the most important consequences of coupling the nematic order to the shape of an elastic body (and perhaps the most relevant for applications) is the effect of spontaneous uniaxial extension/contraction. It is a very simple phenomenon, pictorially illustrated in transformation between states (a) and (b) of Fig. 2. Mathematically, it is straightforward to obtain from (1) for the fixed director orientation $n_z = 1$ and $\det[\mathbf{F}] = 1$, that

$$W_{el} = \frac{1}{2} G \left(\lambda^2 \frac{\ell_{\parallel}^{(0)}}{\ell_{\parallel}} + \frac{2}{\lambda} \frac{\ell_{\perp}^{(0)}}{\ell_{\perp}} \right), \quad (2)$$

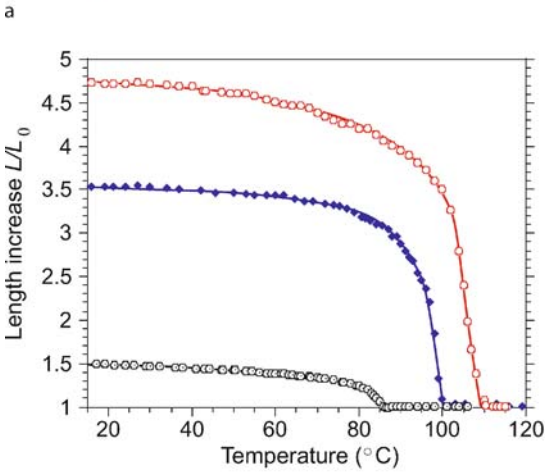
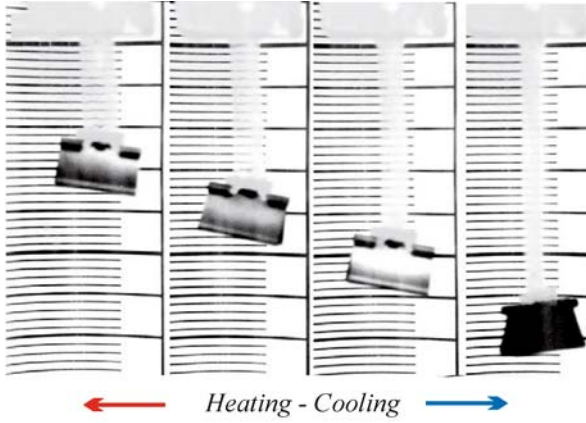
minimizing which one obtains the equilibrium uniaxial extension λ along the nematic director. In the case when the reference state ℓ_0 is isotropic, $\ell_{\parallel}^{(0)} = \ell_{\perp}^{(0)} = a$, this spontaneous extension along the director takes the especially simple form [1,69]:

$$\lambda_m = (\ell_{\parallel} / \ell_{\perp})^{1/3}. \quad (3)$$

The particular way ℓ_{\parallel} and ℓ_{\perp} depend on the value of nematic order parameter Q depends on the molecular structure of the system. For weak chain anisotropy it is cer-

tainly linear: $\ell_{\parallel} / \ell_{\perp} = 1 + \alpha Q$, however, there are situations (e. g. in highly anisotropic main-chain nematic polymers) when the ratio $\ell_{\parallel} / \ell_{\perp} \propto \exp[1/(1 - Q)]$ takes very high values. The key relation (3) lies at the heart of nematic elastomers performing as thermal actuators or artificial muscles, Fig. 3, [34,54,60]. The analogy is further enhanced by the fact that the typical stress exerted on such actuation is in the range of 10–100 kPa, the same as in human muscle [54]. A large amount of work has been recently reporting on photo-actuation of nematic elastomers containing e. g. azobenzene moieties, Fig. 4, or on the effect of adding/removing solvents [22,26,37] – in all cases the effect is due to the changing of underlying nematic order parameter Q , affecting the magnitude of the ratio ($\ell_{\parallel} / \ell_{\perp}$) and thus the length of elastomer sample or the exerted force if the length is constrained [19]. Finally, creating an inhomogeneous internal deformation results in the bending actuation response, Fig. 5, [9,53,71].

However useful, the concise expression (1) does arise from a highly idealized molecular model based on the simple Gaussian statistics of polymer chains between crosslinks and, as such, may seem oversimplified. It turns out that the symmetry features, correctly captured in the Trace formula, are more important than various corrections and generalizations of the model. In particular, a much more complex theory of nematic rubber elasticity taking into account chain entanglements (which have to play a very important role in crosslinked networks, where the constraint release by diffusion is prohibited), devel-



b

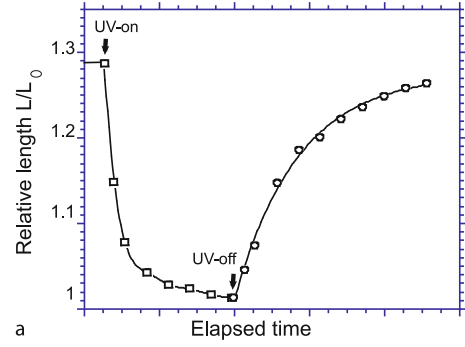
Anisotropic Networks, Elastomers and Gels, Figure 3

Thermal actuation of nematic elastomers. Images a show a strip of rubber with its director aligned along its length, reversibly changing its length on heating and cooling. The same underlying mechanism of changing the orientational order parameter $Q(T)$ translates itself into the mechanical effect differently, depending on the molecular structure of the polymer. Plot b shows the relative change in sample length (the strain λ_m) for different materials, illustrating the range of motion and the different point of transition into the isotropic phase, where $Q = 0$ and $L = L_0$

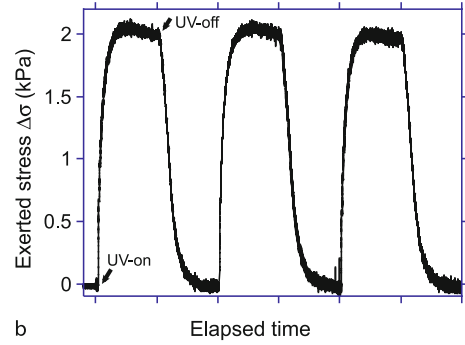
oped in [33], shows the same key symmetries:

$$W_{el} = \frac{2}{3}G \frac{2M+1}{3M+1} \text{Tr}(\mathbf{F}^T \cdot \boldsymbol{\ell}^{-1} \cdot \mathbf{F} \cdot \boldsymbol{\ell}_0) + \frac{3}{2}G(M-1) \frac{2M+1}{3M+1} \left(\overline{|\boldsymbol{\ell}^{-1/2} \cdot \mathbf{F} \cdot \boldsymbol{\ell}_0^{1/2}|} \right)^2 + G(M-1) \ln \left| \boldsymbol{\ell}^{-1/2} \cdot \mathbf{F} \cdot \boldsymbol{\ell}_0^{1/2} \right|, \quad (4)$$

where M is a number of entanglements on a typical chain of length N ($M = 1$ in the ideal Trace formula), and the



a



b

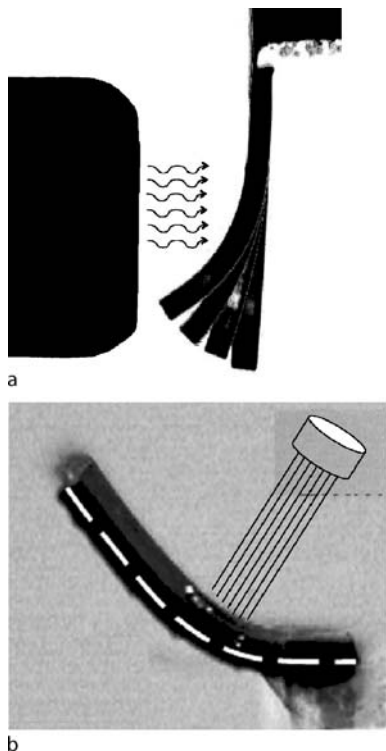
Anisotropic Networks, Elastomers and Gels, Figure 4

Photo-induced actuation of nematic elastomers. Plot a shows the contraction of a strip of azobenzene-containing nematic rubber on UV-irradiation, returning back to its equilibrium value when the light is turned off. Plot b shows the exerted stress under illumination, when the sample is mechanically held. The time-scales of photo-actuation can vary greatly depending on the molecular structure of chromophores

notation $\overline{[\dots]}$ denotes an orientational average of the matrix \dots applied to an arbitrary tube segment. The important feature of the complicated expression (4) is the “sandwiching” of the deformation gradient tensor \mathbf{F} between the two orientational tensors defined in the current and the reference states, respectively. The continuum model of fully non-linear elasticity is also quite complicated in its applications, but the fundamental requirements that any elastic free energy expression must be (i) invariant with respect to body rotations of the reference state and the current state, separately, and (ii) reduce to the classical expression $\text{Tr}(\mathbf{F}^T \cdot \mathbf{F})$ when the underlying chains become isotropic, requires it to have the form

$$W_{el} \propto \text{Tr}(\mathbf{F}^T \cdot [\boldsymbol{\ell}]^{-m} \cdot \mathbf{F} \cdot [\boldsymbol{\ell}_0]^m), \quad (5)$$

where the power is only allowed to be $m = 1, 2, 3$. Once again, one recognizes the key symmetry of separating the two powers of deformation gradient tensor by the orien-



Anisotropic Networks, Elastomers and Gels, Figure 5
 Bending of nematic elastomer cantilevers. Image collage **a** shows the effect of heat (IR radiation) applied from one side of the rubber strip, resulting in the near side contracting more than the far side. Image **b** shows the similar effect on photo-elastomer illuminated from one side

tation tensors defined in the current and reference frames, respectively.

Nematic Elastomers: Soft Elasticity and Dynamics

Two essential consequences of the coupling between the elastic modes of polymer network, described by the general strain tensor F , and the rotational modes of the nematic director, which are the principal axes of chain anisotropy tensor ℓ , are the reduction of the effective elastic response and the penalty on the director fluctuations. The first effect has received the name “soft elasticity” and is the result of the special symmetry of the coupling between the orientational modes and deformation gradients. It is easy to see [44] that there is a whole continuous class of deformations described by the form

$$F_{\text{soft}} = \ell_{\theta}^{1/2} \cdot V \cdot \ell_0^{-1/2}, \quad (6)$$

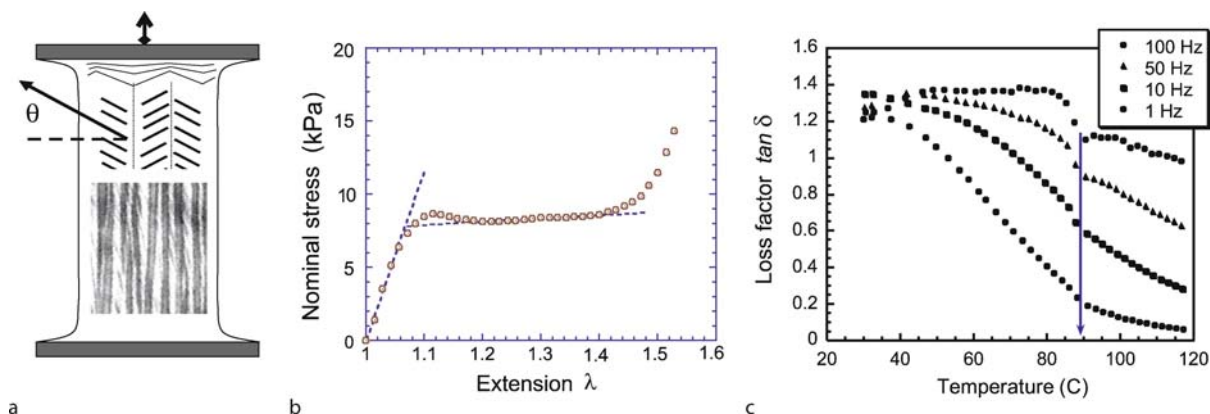
with V an arbitrary unitary matrix, which leave the elas-

tic free energy W_{el} at its constant minimum value (for an incompressible system). Remarkably, this remains true whether one examines the Trace formula (1), or any other expression above, (4) or (5), as long as they respect the correct symmetries of the two independent degrees of freedom [24,50]. Figure 2c illustrates one example of such soft deformation.

This phenomenon is unique to anisotropic elastomers and gels, which have an internal microstructure (liquid crystalline order) that is capable of continuously changing its tensor structure. It can also be seen in smectic elastomers, although the layer constraints make it more complicated mathematically. Possibly the closest elastic system is the famous shape-memory alloy [5], where the martensite transition creates a crystalline unit cell with several (usually just two) equal-energy states. The ability of the lattice to switch between these states in response to certain deformation modes gives the “soft deformation” modes to these alloys, albeit only a few discrete ones. In elastomers and gels the “soft deformation” reflects the ability of anisotropic polymer chains to rotate their long axis to accommodate some imposed elastic deformations without changing their shape.

If one instead chooses to focus on the director modes in a nematic elastomer with a fixed (constrained) shape, the coupling between the internal mobile order parameter and the network elasticity provide a large energy penalty for uniform director rotations $\delta \mathbf{n}$ (with respect to the elastically constrained network). This penalty, which appears as a large mass term in the expression for mean-square director fluctuation, results in the suppression of such fluctuations and the related scattering of light from a nematic elastomer [48]. In contrast to optically turbid ordinary liquid nematics, where light is scattered by long-wavelength director fluctuations, the aligned monodomain anisotropic elastomer or gel is totally transparent. However, when the elastic deformations in the network are not constrained and are free to relax, there are certain combinations of polarization and wave vectors of director fluctuations (corresponding to the soft deformation modes), for which the “effective mass” vanishes and the fluctuation spectrum should appear as in ordinary liquid nematics [44].

It is natural to expect that if a sample of monodomain, uniformly aligned nematic elastomer (which usually implies that it has been crosslinked in the aligned nematic phase [34,35]) is stretched along the axis perpendicular to the nematic director \mathbf{n}_0 , the director will switch and point along the axis of uniaxial extension. The early theory (ignoring the effect of soft elasticity) [6] has predicted, and the experiment on polyacrylate LCE [43] confirmed



Anisotropic Networks, Elastomers and Gels, Figure 6

a The scheme of stripe domains in nematic elastomers (*inset* is their image from a polarizing microscope). **b** The stress-strain curve during the stripe domain regime: After a threshold strain, the soft plateau extends over the whole region where the internal director rotation takes place ($\theta \in 0 \div 90^\circ$). **c** The effect of anomalous vibration damping, demonstrated by the consistently high loss factor across the whole range of nematic phase; the arrow points at the onset of the isotropic phase where the high damping disappears

that this switching may occur in an abrupt discontinuous fashion when the natural long dimension of anisotropic polymer chains can fit into the new shape of the sample, much extended in the perpendicular direction. However the same experiment performed on a polysiloxane LCE [31] has shown an unexpected stripe domain pattern. Further investigation has proven that the nematic director rotates continuously from \mathbf{n}_0 towards the new perpendicular axis, over a substantial range of deformations, but the direction of this rotation alternates in semi-regular stripes of several microns width oriented along the stretching direction, Fig. 6a. Later the same textures have been observed by other groups and on different materials, including polyacrylates [32,73], although there also have been reports confirming the single director switching mode [46].

Theoretical description of stripe domains [66] has been very successful and has led to several long-reaching consequences. First of all, this phenomenon has become a conclusive proof of the soft elasticity principles. As such, the nematic elastomers have been recognized as part of a greater family of elastic systems with microstructure (shape memory alloys being the most famous member of this family to date), all exhibiting similar phenomena. Finally, the need to understand the threshold value of extension λ_c has led to deeper understanding of internal microscopic constraints in LCE and resulted in introduction of the concept of semi-softness: A class of deformations that has the “soft” symmetry, as in Eq. (6), penalized by a small but physically distinct elastic energy due to such random constraints.

The main result of theoretical model [66] gives the director angle variation with strain,

$$\theta(\lambda) = \pm \arcsin \left[\frac{\ell_{\parallel}}{\ell_{\parallel} - \ell_{\perp}} \left(1 - \frac{\lambda_c^2}{\lambda^2} \right) \right]^{1/2} \quad (7)$$

with only one free parameter, the threshold strain λ_c . In fact, if one uses the pure Trace formula (1), the threshold is not present at all, $\lambda_c = 1$. The backbone chain anisotropy $\ell_{\parallel}/\ell_{\perp}$, which enters the theory, is an independent experimentally accessible quantity related, e.g. to the spontaneous shape change of LCE on heating it into the isotropic phase, $\lambda_m \approx (\ell_{\parallel}/\ell_{\perp})^{1/3}$ in Fig. 2. This allowed the data for director angle to be collapsed onto the same master curve, spanning the whole range of non-linear deformations.

The physical reason for the stretched LCE to break into stripe domains with the opposite director rotation $\pm\theta(\lambda)$ becomes clear when one recalls the idea of soft elasticity [44,67]. The polymer chains forming the network are anisotropic, in most cases having the average shape of uniaxial prolate ellipsoid, see Fig. 2. If the assumption of affine deformation is made, the strain applied to the whole sample is locally applied to all network strands. The origin of (entropic) rubber elasticity is the corresponding change of shape of the chains, away from their equilibrium shape frozen at network formation, which results in the reduction in entropy and rise in the elastic free energy. However, the nematic (e.g. prolate anisotropic) chains may find another way of accommodating the deformation: If the sample is stretched perpendicular to the director \mathbf{n} (the

long axis of chain gyration volume), the chains may rotate their *undeformed* ellipsoidal shapes – thus providing an extension, but necessarily in combination with simple shear – and keep their entropy constant and elastic free energy zero! This, of course, is unique to nematic elastomers: Isotropic chains (with spherical shape) have to deform to accommodate any deformation. It is also important that with deformations costing no elastic energy, there is no need for the material to change its preferred nematic order parameter Q , so the ratio $\ell_{\parallel}/\ell_{\perp}$ remains constant for the duration of soft deformations (this is not the case when, e. g., one stretched the elastomer along the director axis). The physical explanation of stripe domains is now clear: the stretched nematic network attempts to follow the soft deformation route to reduce its elastic energy, but this requires a global shear deformation which is prohibited by rigid clamps on two opposite ends of the sample, Fig. 6a. The resolution of this macroscopic constraint is to break into small parallel stripes, each with a completely soft deformation (and a corresponding shear) but with the sign of director rotation (and thus – the shear) alternating between the stripes. Then there is no global shear and the system can lower its elastic energy in the bulk, although it now has to pay the penalty for domain walls and for the non-uniform region of deformation near the clamps. The balance of gains and losses determines the domain size d .

The argument above seems to provide a reason for the threshold strain λ_c , which is necessary to overcome the barrier for creating domain walls between the “soft” stripes. However, it turns out that the numbers do not match. The energy of domain walls must be determined by the gradient Frank elasticity of the nematic (with the Frank elastic constant independently measured, $K \sim 10^{-11}$ N) and thus should be very small, since the characteristic length scale of nematic rubbers is $\xi = \sqrt{K/G} \sim 10^{-9}$ m. Hence the threshold provided by domain walls alone would be vanishingly small whereas most experiments have reported $\lambda_c \sim 1.1$ or more. This mystery has led to a whole new concept of what is now called semi-softness of LCE. The idea is that, due to several different microscopic mechanisms [65], a small addition to the classical nematic rubber-elastic free energy is breaking the symmetry required for the soft deformations:

$$W_{\text{el}} \approx \frac{1}{2} G \left[\text{Tr}(\mathbf{F}^T \cdot \boldsymbol{\ell}_{\theta}^{-1} \cdot \mathbf{F} \cdot \boldsymbol{\ell}_0) + \alpha (\delta \mathbf{n} + \mathbf{n} \cdot \mathbf{F} \times \delta \mathbf{n})^2 \right] \quad (8)$$

(usually $\alpha \ll 1$). The soft-elastic pathways are still representing the low-energy deformations, but the small penalty $\sim \alpha G$ provides the threshold for stripe domain formation

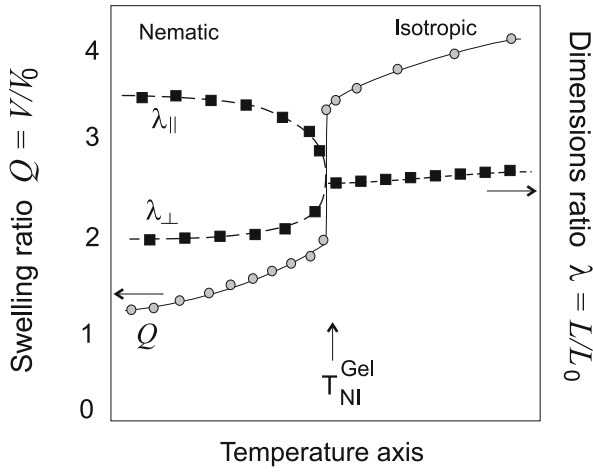
($\lambda_c \sim 1 + \alpha$) and also makes the slope of the stress-strain soft-elastic plateau small but nonzero, Fig. 6b. Compare this with the ordinary extension modulus of rubber before and after the region of stripe domains. Regardless of small complications of semi-soft corrections, the main advance in identifying the whole class of special low-energy soft deformations in LCE and proving their existence by direct experiment is worth noting.

The presence of the underlying nematic order, with its director capable of independently moving with respect to the matrix [16], leads to another remarkable feature of anisotropic elastomers and gels: the anomalous dissipation of mechanical energy, see Fig. 6c. This effect is represented in very high values of the so-called loss factor $\tan \delta = G''/G'$, representing the ratio of the imaginary and real parts of the linear modulus of response to the oscillating shear. $\tan \delta > 1$ in a fully elastic material is a very unusual situation, no doubt to be explored in applications in vibration and impact damping [15]. Another aspect of high internal dissipation due to the rotating director modes is the very high amplitude and very long times of stress relaxation of stress in all liquid crystalline elastomers and gels [14].

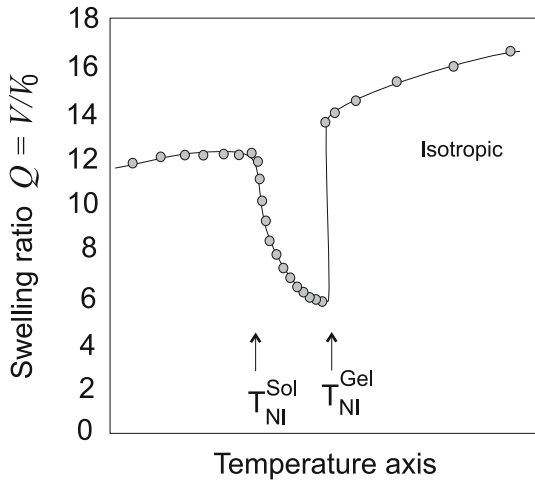
Swelling and Anisotropic Gels

Polymer gels are usually isotropic and swell and shrink equally in all directions. Volume phase transition phenomena of polymer gels have been extensively investigated and it is known that an infinitesimal change in an environmental intensive variable such as solvent composition, pH and temperature yields a discontinuous volume change for some polymer gels [4,27,55]. These volume transitions are driven by the balance between the repulsive and attractive forces acting on the network chains such as van der Waals, hydrophobic, ionic, hydrogen bonding [49].

If gels of anisotropic molecular structure were prepared, anisotropic mechanical behavior will be expected. Additional volume transition can be triggered by nematic ordering of anisotropic molecules inside the gel [62]. This effect adds a new driving molecular force for volume transition phenomena. Temperature-sensitive gels exhibiting volume transition at a certain temperature have attracted much attention of scientist and technologists because of their applications to drug delivery systems and sensors, originally concentrating on N-isopropylacrylamide-based isotropic systems. Spontaneously anisotropic gels undergoing a sharp and large volume change accompanied by nematic-isotropic transition is a new type of such temperature-sensitive gels, which will extend the potential applications.



a



b

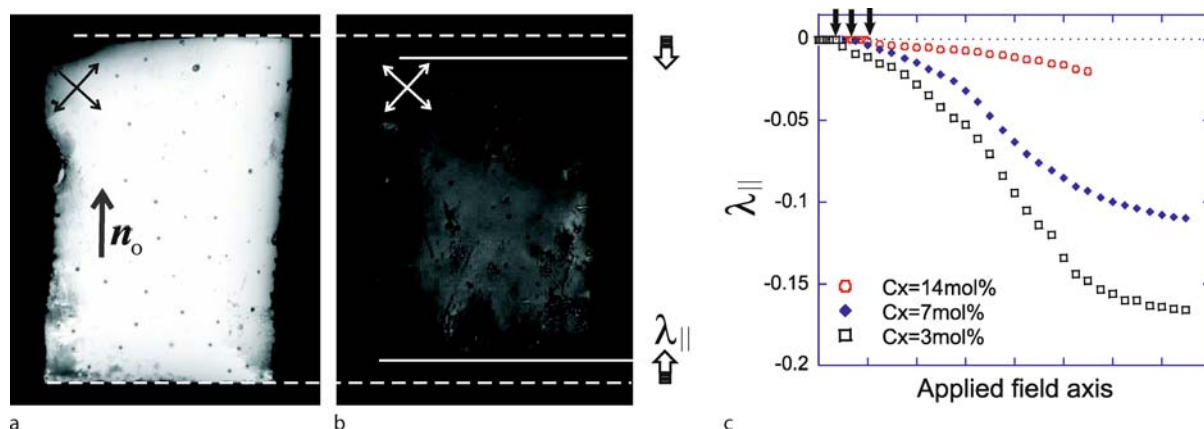
Anisotropic Networks, Elastomers and Gels, Figure 7

Volume transitions and anisotropic swelling of nematic gels. Plot a shows the generic effect of equilibrium swelling by isotropic solvent, at different temperatures (with T_{NI}^{Gel} marking the point of nematic-isotropic transition). The anisotropy of swelling is indicated by the growing difference between the expanding length along the nematic director (a factor $\lambda_{||}$) and perpendicular to the director (λ_{\perp}), with $Q = \lambda_{||}\lambda_{\perp}^2$. Plot b indicates the transitions when the network is swollen by the liquid crystalline solvent, which has its own transition temperature T_{NI}^{Sol} .

Monodomain nematic networks in the dry state show a large spontaneous elongation parallel to the director axis upon nematic ordering, as discussed in earlier sections.

This shape change in dry anisotropic elastomers occurs without appreciable volume change. Large anisotropic volume changes occur in such networks since the addition of isotropic solvent also has the effect of reducing the nematic order, but also remove the incompressibility constraint [29,61,72]. Upon cooling, the swollen isotropic gel continuously changes into the shrunken monodomain nematic gel around the transition point T_{NI}^{Gel} , see Fig. 7a. Importantly, the volume decrease upon nematic ordering is accompanied by the shape change of a gel with a considerable degree of anisotropy, as indicated on the generic plot. The driving force behind the abrupt and pronounced shrinking of the nematic gel is the thermodynamic force (effective chemical potential) that tries to expel the “wrong” solvent and allow the network to lower its free energy of the nematic phase. Liquid crystalline networks swollen in a liquid crystalline solvent is an even richer system, expected to act as a soft actuator driven under electric field utilizing the strong dielectric anisotropy of mesogenic units controllable by external field [30]. Figure 7b shows a generic plot of the swelling ratio $Q = V/V_0$. This plot indicates that the large shrinking occurs in the window of temperatures when the solvent is still isotropic. However, the solvent is taken into the gel once again when it becomes nematic itself, since this now promotes the order in the network and lowers the joint free energy [41,62].

In stark contrast with ordinary liquid crystals, dry anisotropic elastomers do not show any significant response to an applied electric or magnetic field. The reason is that the local elastic resistance (measured by the elastic energy density, essentially the modulus $G \sim 10^4\text{--}6\text{ Pa}$) is much greater than the energy density associated with local dielectric or diamagnetic anisotropy of the material. Essentially, the fields are too weak to rotate the director pinned to the elastic matrix. However, in highly swollen gels the elastic modulus is much lower and one expects (and indeed finds) an electro-optical and electro-mechanical response [10,59,63]. An additional interesting possibility is when a gel sample is mechanically unconstrained and therefore capable of finding a soft-deformation route in response to a dielectric torque applied to its local director. In this case the only resistance to dielectric torque would come from the weak semi-softness. Figure 8 shows what happens when a monodomain nematic gel, freely floating in a solvent, is subjected to an electric field perpendicular to its director [63]. The director rotates, and the shape of the gel changes accordingly, see Fig. 2c, contracting in the parallel direction but remaining constant in the perpendicular direction ($\lambda_{\perp} = 1$) as required by the soft-deformation geometry.



Anisotropic Networks, Elastomers and Gels, Figure 8

Response of an unconstrained nematic gel to an electric field. Image a shows the gel with the director n_0 aligned in the plane, viewed from above between crossed polars. Electric field is applied perpendicular to the image plane and the director rotates out of the plane, so that image b shows extinction between crossed polars. Note that the sample width remains constant. The associated contraction of the gel along the original director, $\lambda_{||}$ is plotted against applied field for several crosslinking density values. The arrows on the plot indicate the semi-soft threshold for the director rotation onset, shifting with crosslinking density. (The data by K. Urayama)

Cholesteric Elastomers: Photonics

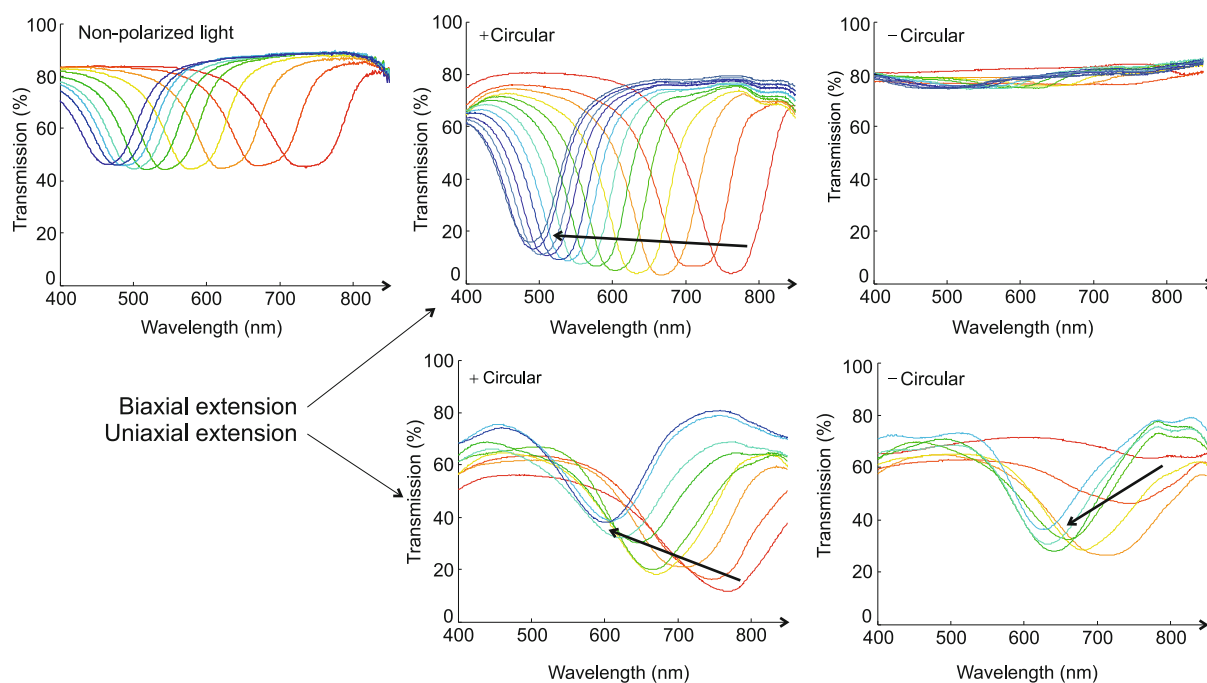
When chiral molecular moieties (or dopant) are added to nematic polymers, they form cholesteric phases. Crosslinked networks made from such polymers accordingly have a spontaneously twisted director distribution. Their director distribution being periodic with characteristic length scale in the optical range, they display their inhomogeneous modulation in brilliant colors of selective reflection. Being macroscopically non-centrosymmetric, such elastic materials possess correspondingly rare physical properties, such as the piezoelectricity.

As with nematic LCE, in order to access many unique properties of cholesteric elastomers one needs to form well-aligned monodomain samples, which would remain ordered due to crosslinking. As with nematic systems, the only way to achieve this is to introduce the network crosslinks when the desired phase is established by either external fields or boundary conditions – otherwise a polydomain elastomer would result. As with nematics, there are certain practical uses of cholesteric polydomain state – for instance, the applications of stereo-selective swelling (see below) may work even better in a system with micron-size cholesteric domains [17].

There are two established ways to create monodomain cholesteric elastomer films with the director in the film plane and the helical axis uniformly aligned perpendicular to the film. Each has its advantages, but also unavoidable drawbacks. Traditionally, a technique of photopolymerization (and network crosslinking when di-functional

molecules are used, such as di-acrylates) was used in cells where the desired cholesteric alignment was achieved and maintained by proper boundary conditions. Many elaborate and exciting applications have been developed using this approach, e. g. [8,25]. However, it is well-known that the influence of surface anchoring only extends a maximum of few microns into the bulk of a liquid crystal (unless a main-chain mesogenic polymer, with dramatically increased planar anchoring strength is employed [57]). Therefore, only very thin cholesteric LCE films can be prepared by this surface-alignment and photopolymerization technique.

A different approach to preparing monodomain cholesteric elastomers has been developed by Kim and Finkelmann [28], utilizing the two-step crosslinking principle originally used in nematic LCE. One prepares a sample of cholesteric polymer gel, crosslinked only partially, and the deposits it on a substrate to de-swell. Since the area in contact with the substrate remains constrained, the sample cannot change its two lateral dimensions in the plane to accommodate the loss of volume; only the thickness can change. These constraints in effect apply a large uniaxial compression, equivalent to a biaxial extension in the flat plane, which serves to align the director in the plane and the helical axis perpendicular to it. After this is achieved, the full crosslinking of the network is initiated and the well-aligned free-standing cholesteric elastomer results. This is a delicate technique, which requires a careful balance of crosslinking/deswelling rates, solvent content and temperature of the phases. In princi-

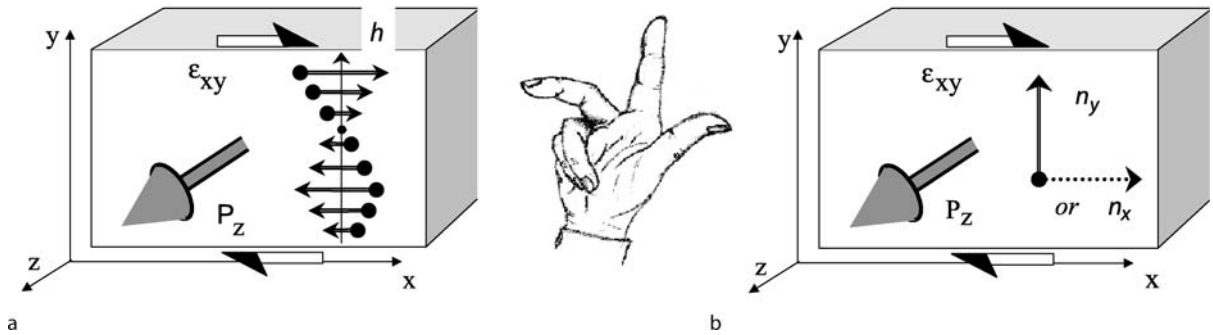


Anisotropic Networks, Elastomers and Gels, Figure 9

Selective reflection of light by cholesteric elastomers shifts under deformation. The *top row of plots* shows the results for a biaxial extension in the plane, leading to uniaxial compression along the helix from 1 to 0.65, separating the transmission of right- and left-handed circular polarized light. The *bottom row of plots* shows the analogous results for a uniaxial extension in the plane, from 1 to 1.6. The key difference to note is the emergence of the matching selective reflection for the opposite chirality of light in the uniaxial extension case [13]

ple it has no significant restrictions on the resulting elastomer area or thickness. However, there is a limitation of a different nature: one cannot achieve too long cholesteric pitch, e.g. in the visible red or infra-red range. The reason is the required high strength of chiral twisting at the stage of biaxial extension on deswelling: With no or weak twisting strength the material is close to the ordinary nematic and although the symmetric biaxial extension does confine the director to its plane, it does nothing to counter the quenched disorder of the establishing crosslinks and the resulting propensity to form a 2-dimensional (planar) polydomain state. Only at sufficiently high chiral twisting power the cholesteric propensity overcomes this planar disorder – and the resulting cholesteric pitch will end up relatively short. Another unavoidable drawback of this 2-step crosslinking technique is the disordering effect of the portion of crosslinks established at the first stage of gel formation (one has to have them to sustain elastic stress and prevent flow on deswelling), so the optical quality of samples prepared in this way is always worse than that in thin films photo-polymerized in a perfect surface-aligned cholesteric state.

Cholesteric elastomers and gels respond to imposed mechanical deformations in different ways. Bending of flexible free-standing cholesteric LCE has been used to great effect [40]. The more mechanically straightforward idea is to compress the elastomer along the helical axis, which is easiest to achieve by imposing a symmetric biaxial extension in the director plane [21]. This leads to the affine contraction of the helical pitch and the corresponding dramatic shift of the selective reflection band, see Fig. 9. If a more traditional uniaxial stretching of elastomer films is applied, with the initial helical pitch perpendicular to the direction of stretching, the modified director distribution is usually no longer simply helical although the characteristic length scales also affinely scale with sample dimensions. On deformation the director texture remains periodic along the former helical pitch, but becomes a non-chiral step-wise modulation [12,70], leading to new photonic bandgaps and selective reflection in both right- and left-handed circular polarized light, Fig. 9. Laser emission has been predicted and observed in cholesteric systems when light was excited in their bulk, e.g. by a photo-dye stimulated by a pumping laser emit-



Anisotropic Networks, Elastomers and Gels, Figure 10

Piezoelectric effects in chiral elastomers. **a** Polarization induced by the shear applied to helically twisted textures (flexoelectric effect); **b** Polarization induced by shearing a uniformly aligned chiral nematic, with the director either along x- or y-axes (the true piezoelectric effect $\mathbf{P} = \gamma[\mathbf{n} \times (\boldsymbol{\epsilon} \cdot \mathbf{n})]$). The sketch illustrates the chiral geometry that produces a polarization vector along z due to the shear and anisotropy axis lying in the x-y plane

ting near the bandgap edge. The ability to continuously change the position of bandgap by mechanically deforming cholesteric elastomers has led to an attractive application opportunity of tunable lasers [21,40,47].

Another possibility is topological imprinting of helical director distribution. Being anchored to the rubber matrix due to crosslinking, the director texture can remain macroscopically chiral even when all chiral dopants are removed from the material [18,39]. Internally stored mechanical twist can cause cholesteric elastomers to interact selectively with solvents according to its imprinted handedness. Separation of racemic mixtures by stereo-selective swelling is an interesting application possibility.

The lack of centro-symmetric invariance in chiral elastic materials leads to interesting and rare physical properties [56], in particular, piezoelectric effect and non-linear optical properties. Such characteristics of elastomers with a chiral smectic C^* order (the ferroelectric liquid crystalline elastomers, FLCE) have been studied with some intensity for several years now. After the permanently monodomain (fixed by crosslinking) free-standing samples of FLCE were prepared [20,23], several useful experiments targeting various electro-mechanical and electro-optical properties, in particular – the piezoelectricity and the non-linear optical response, have been reported in recent years [7,36]. Clearly, the prospect of important applications will continue driving this work. In this short review we shall concentrate on the analogous effect in chiral nematic LCE which do not possess a spontaneous polarization.

With much higher symmetry of nematic and cholesteric elastomers (the point group D_∞ in a chiral material, in contrast to a very low symmetry: C_2 plus the translational effect of layers in ferroelectric smectic C^*), there is

a real possibility to identify microscopic origins of piezoelectricity in amorphous polymers or indeed elastomers, if one aims to have an equilibrium effect in a stress-resistant material. The piezoelectric effect in a uniform chiral nematic LCE has been described phenomenologically [56] and by a fully non-linear microscopic theory similar in its spirit to the basic Trace formula for nematic rubber elasticity [58]. All experimental research so far has concentrated on the helically twisted cholesteric elastomers [11,42,64]. However, the cholesteric texture under the required shear deformation [45], Fig. 10a, will produce highly non-uniform distortions giving rise to the well-understood flexoelectric effect and masking the possible chiral piezoelectricity.

The uniform linear piezoelectricity, i. e. the polarization induced by a uniform strain, Fig. 10b, (with the small deformation $\epsilon = \lambda - 1$), is unknown in completely amorphous polymers and rubbers. Even the famous PVDF polymer-based piezoelectric has its response due to inhomogeneity of crystalline regions affected by macroscopic deformation. The molecular theory [58] has examined the effect of chirality in molecular structure of chain monomers and the bias in their short-axis alignment when the chains are stretched at an angle to the average nematic director \mathbf{n} . If the monomers possess a transverse dipole moment, this bias leads to macroscopic polarization:

$$\mathbf{P} \simeq -\frac{1}{2}(n_c \Delta) \underline{\underline{\epsilon}} : (\boldsymbol{\lambda}^T \cdot \boldsymbol{\ell}_\theta^{-1} \cdot \boldsymbol{\lambda} \cdot \boldsymbol{\ell}_0). \quad (9)$$

Here, as in the underlying rubber elasticity, n_c is the number of network strands per unit volume (the crosslinking density) and the parameter Δ is the measure of monomer chirality with the transverse dipole moment d_t , see [58] for detail. Compare this with the original Trace formula

of Eq. (1) to note the characteristic “sandwiching” of strain between the two different orientation tensors. This expression involves the full deformation gradient tensor λ and, therefore can describe large deformations of a chiral nematic rubber. When shear deformations are small, the linear approximation of (9) gives, for a symmetric shear,

$$\mathbf{P} \simeq \gamma [\mathbf{n} \times (\boldsymbol{\varepsilon} \cdot \mathbf{n})],$$

with the linear coefficient $\gamma = \partial P / \partial \varepsilon \approx -\frac{1}{2} n_c \Delta(\ell_{\parallel}^2 - \ell_{\perp}^2) / \ell_{\parallel} \ell_{\perp}$ clearly proportional to the degree of local nematic order through the chain anisotropy $\ell_{\parallel} - \ell_{\perp}$. Piezoelectricity in amorphous rubbers is not only interesting from the point of view of fundamentals of chiral polymer random walks and symmetry breaking. On the practical side, due to the rubber modulus much lower than in ordinary solid piezoelectrics (typically $G \sim 10^5$ Pa), the relevant coefficient $d = \partial P / \partial \sigma = \gamma / G$ is much higher than the corresponding response to stress in, for instance, quartz or PVDF. The corresponding low mechanical impedance should make the piezoelectric rubber attractive for many energy transducing applications.

New Frontiers

When positional ordering, in the form of layering, is added to the orientational order of nematics, we have the smectic phase, Fig. 1c,d. When made into networks, they remain locally liquid-like, can suffer large extensions and have mobile internal degrees of freedom – much like with translationally homogeneous nematic elastomers and gels. This freedom, which gave spontaneous distortions and soft elasticity in nematics, is restricted in smectics by the layers to which the matrix is often strongly coupled. The mechanics of smectic elastomers [38,51] is decidedly more complex than that of nematic elastomers. In many cases the effect of layers means that the material has the non-linear mechanics of a 2D rubber in the layer plane, while perform as a 1-dimensional crystalline solid in the third dimension [2] when the layer translation with respect to the rubbery matrix is penalized by a periodic potential provided by the lamellar order. Alternatively, concentrating on the layer structure, one finds the behavior radically altered from the liquid crystal case. The layers interact via the effective background provided by the elastomer network and, as a result, no longer display the celebrated Landau–Peierls loss of long range order in one- or two- dimensions. The layer buckling instabilities occur when deformations along the layer normal are applied. An even more rich system is the smectic-C elastomer and its variants. This phase has the director tilted in the layer planes and a much lower symmetry, allowing for a great variety

of soft deformation modes [3,52]. The chiral variants of smectic-C networks are ferroelectric solids: Soft, greatly extensible and non-crystalline and thus of great technological significance.

This article examines some of the recent and relevant findings about the new class of materials – liquid crystalline elastomers and gels. Nematic rubbers have already proved themselves an unusual and exciting system, with a number of unique optical and mechanical properties, indeed a real example of the Cosserat media with couple-stress elasticity. Polymer networks with cholesteric order are an equally provocative system promising new physical properties.

The basic molecular theory of nematic rubber appears to be exceptionally simple in its foundation and does not involve any model parameters apart from the backbone polymer chain anisotropy $\ell_{\parallel} / \ell_{\perp}$, which can be independently measured. This is a great advantage over many other soft condensed matter systems requiring complicated, sometimes ambiguous theoretical approaches. Of course, in many real situations and materials one finds a need to look deeper into the microscopic properties; an example of this is the “semi-softness” of nematic networks. Most of these systems are characterized by non-uniform deformations: Even in the simplest experimental set-up a large portion of the sample near the clamps is subjected to non-uniform strains and, therefore, responds with a non-uniform director field.

Looking into the future, many challenging and fundamental problems in this field are still outstanding. The field of dynamics and relaxation in rubbery networks, although not young by any means, is still not offering an unambiguous physical picture of stress relaxation. Adding the liquid crystalline order, we find an additional (director) field undergoing its own relaxation process and coupled to that of an elastic network. In the particular case of polydomain (i. e. randomly disordered in equilibrium) elastomers, we can identify a hierarchical sequence of physical processes in the underlying network (above its T_g) and the superimposed glass-like nematic order. This leads to a particularly slow relaxation, but much remains to be done to understand the physics of such complex systems.

The general problem of dynamic mechanical properties, rheology and relaxation in Cosserat-like incompressible solids, also characterized by the effect of soft elasticity, brings to mind a number of possible applications. An example would be the selective attenuation of certain acoustic waves, with polarization and propagation direction satisfying the condition for softness, essentially leading to an acoustic filtering system. Another example of application of soft elasticity, also related to the problem of relaxation,

is the damping of shear vibrations in an engineering component when its surface is covered by a layer of nematic or smectic rubber, particularly aligned to allow the director rotation and softness.

Other very important area of applications is based on polarizational properties of materials with chirality. Most non-linear optical applications (which have a great technological potential) deal with elastomers in the ferroelectric smectic C^* phase. The low symmetry (in particular, chirality) and large spontaneous polarization of C^* smectics have a strong effect on the underlying elastic network, and vice versa. This also returns back to the general problem of mechanical properties of smectic rubbers and gels. In conclusion, after the initial period of exploration and material synthesis, liquid crystalline elastomers, in all their variety, now offer themselves as an exciting ground for both fundamental research and for technology. Smectic and lamellar liquid-crystalline elastomers and gels are a much more recent area of study, with much more theoretical and experimental work required to underpin their dramatically anisotropic and non-linear mechanical properties combining a two-dimensional rubber-elastic response and a solid-like properties in the third direction.

Bibliography

1. Abramchuk SS, Khokhlov AR (1987) Molecular theory of high elasticity of the polymer networks with orientational ordering of links. *Doklady Akad Nauk SSSR* 297:385
2. Adams JM, Warner M (2005) Elasticity of smectic-A elastomers. *Phys Rev E* 71:021708
3. Adams JM, Warner M (2005) Soft elasticity in smectic elastomers. *Phys Rev E* 72:011703
4. Annaka M, Tanaka T (1992) Multiple phases of polymer gels. *Nature* 355:430
5. Bhattacharya K (2003) *Microstructure of martensite*. Oxford University Press, Oxford
6. Bladon P, Terentjev EM, Warner M (1994) Deformation-induced orientational transitions in liquid crystal elastomers. *J Phys II* 4:75
7. Brehmer M, Zentel R, Giesselmann F, Germer R, Zugemaier P (1996) Coupling of liquid crystalline and polymer network properties in LC-elastomers. *Liq Cryst* 21:589
8. Broer DJ, Lub J, Mol GN (1995) Wide-band reflective polarizers from cholesteric polymer networks with a pitch gradient. *Nature* 378:467
9. Camacho-Lopez M, Finkelmann H, Palffy-Muhoray P, Shelley M (2004) Fast liquid-crystal elastomer swims into the dark. *Nat Mater* 3:307
10. Chang C-C, Chien L-C, Meyer RB (1997) Electro-optical study of nematic elastomer gels. *Phys Rev E* 56:595
11. Chang C-C, Chien L-C, Meyer RB (1997) Piezoelectric effects in cholesteric elastomer gels. *Phys Rev E* 55:534
12. Cicuta P, Tajbakhsh AR, Terentjev EM (2002) Evolution of photonic structure on deformation of cholesteric elastomers. *Phys Rev E* 65:051704
13. Cicuta P, Tajbakhsh AR, Terentjev EM (2004) Photonic bandgaps and optical activity in cholesteric elastomers. *Phys Rev E* 70:011703
14. Clarke SM, Terentjev EM (1998) Slow stress relaxation in randomly disordered nematic elastomers and gels. *Phys Rev Lett* 81:4436
15. Clarke SM, Tajbakhsh AR, Terentjev EM, Remillat C, Tomlinson GR, House JR (2001) Soft elasticity and mechanical damping in liquid crystalline elastomers. *J Appl Phys* 89:6530
16. Clarke SM, Tajbakhsh AR, Terentjev EM, Warner M (2001) Anomalous viscoelastic response of nematic elastomers. *Phys Rev Lett* 86:4044
17. Courty S, Tajbakhsh AR, Terentjev EM (2003) Phase chirality and stereo-selective swelling of cholesteric elastomers. *Eur Phys J E* 12:617
18. Courty S, Tajbakhsh AR, Terentjev EM (2003) Stereo-selective swelling of imprinted cholesterics networks. *Phys Rev Lett* 91:085503
19. Cviklinski J, Tajbakhsh AR, Terentjev EM (2002) UV-isomerisation in nematic elastomers as a route to photo-mechanical transducer. *Eur Phys J E* 9:427
20. Finkelmann H, Benne I, Semmler K (1995) Smectic liquid single-crystal elastomers. *Macromol Symp* 96:169
21. Finkelmann H, Kim ST, Munoz A, Palffy-Muhoray P, Taheri B (2001) Tunable mirrorless lasing in cholesteric liquid crystalline elastomers. *Adv Mater* 13:1069
22. Finkelmann H, Nishikawa E, Pereira GG, Warner M (2001) A new opto-mechanical effect in solids. *Phys Rev Lett* 87:015501
23. Gebhard E, Zentel R (1998) Freestanding ferroelectric elastomer films. *Macromol Rapid Comm* 19:341
24. Golubović L and Lubensky TC (1989) Nonlinear elasticity of amorphous solids. *Phys Rev Lett* 63:1082
25. Hikmet RAM, Kemperman H (1998) Electrically switchable mirrors and optical components made from liquid-crystal gels. *Nature* 392:476
26. Hogan PM, Tajbakhsh AR, Terentjev EM (2002) UV-manipulation of order and macroscopic shape in nematic elastomers. *Phys Rev E* 65:041720
27. Ilmain F, Tanaka T, Kokufuta E (1991) Volume transition in a gel driven by hydrogen-bonding. *Nature* 349:400
28. Kim ST, Finkelmann H (2001) Cholesteric liquid single-crystal elastomers (LSCe) obtained by the anisotropic deswelling method. *Macromol Rapid Comm* 22:429
29. Kishi R, Shishido M, Tazuke S (1990) Liquid-crystalline polymer gels: Anisotropic swelling of poly(γ -benzyl L-glutamate) gel crosslinked under a magnetic field. *Macromolecules* 23:3868
30. Kishi R, Suzuki Y, Ichijo H, Hiras H (1997) Electrical deformation of thermotropic liquid-crystalline polymer gels. *Mol Cryst Liq Cryst* 294:411
31. Kundler I, Finkelmann H (1995) Strain-induced director reorientation in nematic liquid single-crystal elastomers. *Macromol Rapid Comm* 16:679
32. Kundler I, Finkelmann H (1998) Director reorientation via stripe-domains in nematic elastomers. *Macromol Chem Phys* 199:677
33. Kutter S, Terentjev EM (2001) Tube model for the elasticity of entangled nematic rubbers. *Eur Phys J E* 6:221
34. Küpfer J, Finkelmann H (1991) Nematic liquid single-crystal elastomers. *Macromol Rapid Comm* 12:717

35. Legge CH, Davis FJ, Mitchell GR (1991) Memory effects in liquid-crystal elastomers. *J Phys II* 1:1253
36. Lehmann W, Gattinger P, Keck M, Kremer F, Stein P, Eckert T, Finkelmann H (1998) The inverse electromechanical effect in mechanically oriented SmC*-elastomers. *Ferroelectrics* 208:373
37. Li MH, Keller P, Li B, Wang XG, Brunet M (2003) Light-driven side-on nematic elastomer actuators. *Adv Mater* 15:569
38. Lubensky TC, Terentjev EM, Warner M (1994) Layer-network coupling in smectic elastomers. *J Phys II* 4:1457
39. Mao Y, Warner M (2000) Theory of chiral imprinting. *Phys Rev Lett* 84:5335
40. Matsui T, Ozaki R, Funamoto K, Ozaki M, Yoshino K (2002) Flexible mirrorless laser based on a free-standing film of photopolymerized cholesteric liquid crystal. *Appl Phys Lett* 81:3741
41. Matsuyama A, Kato T (2002) Nematic ordering-induced volume phase transitions of liquid crystalline gels. *J Chem Phys* 116:8175
42. Meier W, Finkelmann H (1990) Piezoelectricity of cholesteric elastomers. *Macromol Chem Rapid Comm* 11:599
43. Mitchell GR, Davis FJ, Guo W (1993) Strain-induced transitions in liquid-crystal elastomers. *Phys Rev Lett* 71:2947
44. Olmsted PD (1994) Rotational invariance and goldstone modes in nematic elastomers and gels. *J Phys II* 4:2215
45. Pelcovits RA, Meyer RB (1995) Piezoelectricity of cholesteric elastomers. *J Phys II* 5:877
46. Roberts PMS, Mitchell GR, Davis FJ (1997) A single director switching mode for monodomain liquid crystal elastomers. *J Phys II* 7:1337
47. Schmidtke J, Stille W, Finkelmann H (2003) Defect mode emission of a dye doped cholesteric polymer network. *Phys Rev Lett* 90:083902
48. Schönstein M, Stille W, Strobl G (2001) Effect of the network on the director fluctuations in a nematic side-group elastomer analysed by static and dynamic light scattering. *Eur Phys J E* 5:511
49. Shibayama M, Tanaka T (1993) Volume phase-transition and related phenomena of polymer gels. *Adv Polym Sci* 109:1
50. Stenull O, Lubensky TC (2004) Anomalous elasticity of nematic and critically soft elastomers. *Phys Rev E* 69:021807
51. Stenull O, Lubensky TC (2005) Phase transitions and soft elasticity of smectic elastomers. *Phys Rev Lett* 94:018304
52. Stenull O, Lubensky TC (2006) Soft elasticity in biaxial smectic and smectic-C elastomers. *Phys Rev E* 74:051709
53. Tabiryan N, Serak S, Dai X-M, Bunning T (2005) Polymer film with optically controlled form and actuation. *Opt Express* 13:7442
54. Tajbakhsh AR, Terentjev EM (2001) Spontaneous thermal expansion of nematic elastomers. *Eur Phys J E* 6:181
55. Tanaka T (1978) Collapse of gels and critical endpoint. *Phys Rev Lett* 40:820
56. Terentjev EM (1993) Phenomenological theory of non-uniform nematic elastomers: Free energy of deformations and electric field effects. *Europhys Lett* 23:27
57. Terentjev EM (1995) Density functional model of anchoring energy at a liquid crystalline polymer-solid interface. *J Phys II* 5:159
58. Terentjev EM, Warner M (1999) Piezoelectricity of chiral nematic elastomers. *Eur Phys J B* 8:595
59. Terentjev EM, Warner M, Bladon P (1994) Orientation of liquid crystal elastomers and gels by an electric field. *J Phys II* 4:667
60. Thomsen DL, Keller P, Naciri J, Pink R, Jeon H, Shenoy D, Ratna BR (2001) Liquid crystal elastomers with mechanical properties of a muscle. *Macromolecules* 34:5868
61. Urayama K, Arai YO, Takigawa T (2005) Volume phase transition of monodomain nematic polymer networks in isotropic solvents. *Macromolecules* 38:3469
62. Urayama K, Okuno Y, Nakao T, Kohjiya S (2003) Volume transition of nematic gels in nematogenic solvents. *J Chem Phys* 118:2903
63. Urayama K, Kondo H, Arai YO, Takigawa T (2005) Electrically driven deformations of nematic gels. *Phys Rev E* 71:051713
64. Vallerien SU, Kremer F, Fischer EW, Kapitza H, Zentel R, Poths H (1990) Experimental proof of piezoelectricity in cholesteric and chiral smectic C* phases of LC-elastomers. *Macromol Chem Rapid Comm* 11:593
65. Verwey GC, Warner M (1997) Compositional fluctuations and semisoftness in nematic elastomers. *Macromolecules* 30:4189
66. Verwey GC, Warner M, Terentjev EM (1996) Elastic instability and stripe domains in liquid crystalline elastomers. *J Phys II* 6:1273
67. Warner M, Bladon P, Terentjev EM (1994) 'Soft Elasticity' – Deformations without resistance in liquid crystal elastomers. *J Phys II* 4:93
68. Warner M, Terentjev EM (2007) *Liquid crystal elastomers*, 2nd edn. Oxford University Press, Oxford
69. Warner M, Gelling KP, Vilgis TA (1988) Theory of nematic networks. *J Chem Phys* 88:4008
70. Warner M, Terentjev EM, Meyer RB, Mao Y (2000) Untwisting of a cholesteric elastomer by a mechanical field. *Phys Rev Lett* 85:2320
71. Yu Y, Nakano M, Ikeda T (2003) Directed bending of a polymer film by light – miniaturizing a simple photomechanical system. *Nature* 425:145
72. Yusuf Y, Ono Y, Sumisaki Y, Cladis PE, Brand HR, Finkelmann H, Kai S (2004) Swelling dynamics of liquid crystal elastomers swollen with low molecular weight liquid crystals. *Phys Rev E* 69:021710
73. Zubarev ER, Talroze RV, Yuranova TI, Vasilets VN, Plate NA (1996) Influence of crosslinking conditions on the phase behavior of a polyacrylate-based liquid-crystalline elastomer. *Macromol Rapid Comm* 17:43

Anomalous Diffusion on Fractal Networks

IGOR M. SOKOLOV

Institute of Physics, Humboldt-Universität zu Berlin, Berlin, Germany

Article Outline

[Glossary](#)

[Definition of the Subject](#)

[Introduction](#)

[Random Walks and Normal Diffusion](#)

[Anomalous Diffusion](#)

[Anomalous Diffusion on Fractal Structures](#)

[Percolation Clusters](#)

Scaling of PDF and Diffusion Equations
on Fractal Lattices
Further Directions
Bibliography

Glossary

Anomalous diffusion An essentially diffusive process in which the mean squared displacement grows, however, not as $\langle R^2 \rangle \propto t$ like in normal diffusion, but as $\langle R^2 \rangle \propto t^\alpha$ with $\alpha \neq 0$, either asymptotically faster than in normal diffusion ($\alpha > 1$, superdiffusion) or asymptotically slower ($\alpha < 1$, subdiffusion).

Comb model A planar network consisting of a backbone (spine) and teeth (dangling ends). A popular simple model showing anomalous diffusion.

Fractional diffusion equation A diffusion equation for a non-Markovian diffusion process, typically with a memory kernel corresponding to a fractional derivative. A mathematical instrument which adequately describes many processes of anomalous diffusion, for example the continuous-time random walks (CTRW).

Walk dimension The fractal dimension of the trajectory of a random walker on a network. The walk dimension is defined through the mean time T or the mean number of steps n which a walker needs to leave for the first time the ball of radius R around the starting point of the walk: $T \propto R^{d_w}$.

Spectral dimension A property of a fractal structure which substitutes the Euclidean dimension in the expression for the probability to be at the origin at time t , $P(0, t) \propto t^{-d_s/2}$. Defines the behavior of the network Laplace operator.

Alexander–Orbach conjecture A conjecture that the spectral dimension of the incipient infinite cluster in percolation is equal to $4/3$ independently on the Euclidean dimension. The invariance of spectral dimension is only approximate and holds within 2% accuracy even in $d = 2$. The value of $3/4$ is achieved in $d > 6$ and on trees.

Compact visitation A property of a random walk to visit practically all sites within the domain of the size of the mean squared displacement. The visitation is compact if the walk dimension d_w exceeds the fractal dimension of the substrate d_f .

Definition of the Subject

Many situations in physics, chemistry, biology or in computer science can be described within models related to random walks, in which a “particle” (walker) jumps from one node of a network to another following the network’s

links. In many cases the network is embedded into Euclidean space which allows for discussing the displacement of the walker from its initial site. The displacement’s behavior is important e. g. for description of charge transport in disordered semiconductors, or for understanding the contaminants’ behavior in underground water. In other cases the discussion can be reduced to the properties which can be defined for a whatever network, like return probabilities to the starting site or the number of different nodes visited, the ones of importance for chemical kinetics or for search processes on the corresponding networks. In the present contribution we only discuss networks embedded in Euclidean space.

One of the basic properties of normal diffusion is the fact that the mean squared displacement of a walker grows proportionally to time, $R^2 \propto t$, and asymptotic deviations from this law are termed anomalous diffusion. Microscopic models leading to normal diffusion rely on the spatial homogeneity of the network, either exact or statistical. If the system is disordered and its local properties vary from one its part to another, the overall behavior at very large scales, larger than a whatever inhomogeneity, can still be diffusive, a property called homogenization.

The ordered systems or systems with absent long-range order but highly homogeneous on larger scales do not exhaust the whole variety of relevant physical systems. In many cases the system can not be considered as homogeneous on a whatever scale of interest. Moreover, in some of these cases the system shows some extent of scale invariance (dilatation symmetry) typical for fractals. Examples are polymer molecules and networks, networks of pores in some porous media, networks of cracks in rocks, and so on. Physically all these systems correspond to a case of very strong disorder, do not homogenize even at largest relevant scales, and show fractal properties. The diffusion in such systems is anomalous, exhibiting large deviations from the linear dependence of the mean squared displacement on time. Many other properties, like return probabilities, mean number of different visited nodes etc. also behave in a way different from the one they behave in normal diffusion. Understanding these properties is of primary importance for theoretical description of transport in such systems, chemical reactions in them, and in devising efficient search algorithms or in understanding ones used in natural search, as taking place in gene expression or incorporated into the animals’ foraging strategies.

Introduction

The theory of diffusion processes was initiated by A. Fick who was interested in the nutrients’ transport in a living

body and put down the phenomenological diffusion equation (1855). The microscopic, probabilistic discussion of diffusion started with the works of L. Bachelier on the theory of financial speculation (1900), of A. Einstein on the motion of colloidal particles in a quiescent fluid (the Brownian motion) (1905), and with a short letter of K. Pearson to the readers of Nature in the same year motivated by his work on animal motion. These works put forward the description of diffusion processes within the framework of random walk models, the ones describing the motion as a sequence of independent bounded steps in space requiring some bounded time to be completed. These models lead to larger scales and longer times to a typical diffusion behavior as described by Fick's laws. The most prominent property of this so-called normal diffusion is the fact that the mean squared displacement of the diffusing particle from its initial position at times much longer than the typical time of one step grows as $\langle x^2(t) \rangle \propto t$. Another prominent property of normal diffusion is the fact that the distribution of the particle's displacements tends to a Gaussian (of width $\langle x^2(t) \rangle^{1/2}$), as a consequence of the independence of different steps.

The situations described by normal diffusion are abundant, but looking closer into many other processes (which still can be described within a random walk picture, however, with only more or less independent steps, or with steps showing a broad distribution of their lengths or times) revealed that these processes, still resembling or related to normal diffusion, show a vastly different behavior of the mean squared displacement, e. g. $\langle x^2(t) \rangle \propto t^\alpha$ with $\alpha \neq 1$. In this case one speaks about anomalous diffusion [11]. The cases with $\alpha < 1$ correspond to *subdiffusion*, the ones with $\alpha > 1$ correspond to *superdiffusion*. The subdiffusion is typical for the motion of charge carriers in disordered semiconductors, for contaminant transport by underground water or for proteins' motion in cell membranes. The models of subdiffusion are often connected either with the continuous-time random walks or with diffusion in fractal networks (the topic of the present article) or with combinations thereof. The superdiffusive behavior, associated with divergent mean square displacement per complete step, is encountered e. g. in transport in some Hamiltonian models and in maps, in animal motion, in transport of particles by flows or in diffusion in porous media in Knudsen regime.

The diffusion on fractal networks (typically subdiffusion) was a topic of extensive investigation in 1980s–1990s, so that the most results discussed here can be considered as well-established. Therefore most of the references in this contribution are given to the review articles, and only few of them to original works (mostly to the pioneering publi-

cations or to newer works which didn't find their place in reviews).

Random Walks and Normal Diffusion

Let us first turn to normal diffusion and concentrate on the mean squared displacement of the random walker. Our model will correspond to a random walk on a regular lattice or in free space, where the steps of the random walker will be considered independent, identically distributed random variables (step lengths \mathbf{s}_i) following a distribution with a given symmetric probability density function (PDF) $p(\mathbf{s})$, so that $\langle \mathbf{s} \rangle = 0$. Let a be the root mean square displacement in one step, $\langle \mathbf{s}_i^2 \rangle = a^2$. The squared displacement of the walker after n steps is thus

$$\langle r_n^2 \rangle = \left\langle \left(\sum_{i=1}^n \mathbf{s}_i \right)^2 \right\rangle = \sum_{i=1}^n \langle \mathbf{s}_i^2 \rangle + 2 \sum_{i=1}^n \sum_{j=i+1}^n \langle \mathbf{s}_i \mathbf{s}_j \rangle.$$

The first sum is simply na^2 , and the second one vanishes if different steps are uncorrelated and have zero mean. Therefore $\langle r_n^2 \rangle = a^2 n$: The mean squared displacement in a walk is proportional to the number of steps.

Provided the mean time τ necessary to complete a step exists, this expression can be translated into the temporal dependence of the displacement $\langle r^2(t) \rangle = (a^2/\tau)t$, since the mean number of steps done during the time t is $n = t/\tau$. This is the time-dependence of the mean squared displacement of a random walker characteristic for normal diffusion. The prefactor a^2/τ in the $\langle r^2(t) \rangle \propto t$ dependence is connected with the usual diffusion coefficient K , defined through $\langle r^2(t) \rangle = 2dKt$, where d is the dimension of the Euclidean space. Here we note that in the theory of random walks one often distinguishes between the situations termed as genuine “walks” (or velocity models) where the particle moves, say, with a constant velocity and changes its direction at the end of the step, and “random flight” models, when steps are considered to be instantaneous (“jumps”) and the time cost of a step is attributed to the waiting time at a site before the jump is made. The situations considered below correspond to the flight picture.

There are three important postulates guaranteeing that random walks correspond to a normal diffusion process:

- The existence of the mean squared displacement per step, $\langle \mathbf{s}^2 \rangle < \infty$
- The existence of the mean time per step (interpreted as a mean waiting time on a site) $\tau < \infty$ and
- The uncorrelated nature of the walk $\langle \mathbf{s}_i \mathbf{s}_j \rangle = \langle \mathbf{s}^2 \rangle \delta_{ij}$.

These are exactly the three postulates made by Einstein in his work on Brownian motion. The last postulate can be

weakened: The persistent (i. e. correlated) random walks still lead to normal diffusion provided $\sum_{j=i}^{\infty} \langle \mathbf{s}_i \mathbf{s}_j \rangle < \infty$.

In the case when the steps are independent, the central limit theorem immediately states that the limiting distribution of the displacements will be Gaussian, of the form

$$P(\mathbf{r}, t) = \frac{1}{(4\pi Kt)^{d/2}} \exp\left(-\frac{d\mathbf{r}^2}{4Kt}\right).$$

This distribution scales as a whole: rescaling the distance by a factor of λ , $x \rightarrow \lambda x$ and of the time by the factor of λ^2 , $t \rightarrow \lambda^2 t$, does not change the form of this distribution. The moments of the displacement scale according to $\langle \mathbf{r}^k \rangle = \int P(\mathbf{r}, t) \mathbf{r}^k d^d \mathbf{r}^k = \text{const}(k) t^{k/2}$. Another important property of the the PDF is the fact that the return probability, i. e. the probability to be at time t at the origin of the motion, $P(\mathbf{0}, t)$, scales as

$$P(\mathbf{0}, t) \propto t^{-d/2}, \quad (1)$$

i. e. shows the behavior depending on the substrate's dimension.

The PDF $P(\mathbf{r}, t)$ is the solution of the Fick's diffusion equation

$$\frac{\partial}{\partial t} P(\mathbf{r}, t) = k \Delta P(\mathbf{r}, t), \quad (2)$$

where Δ is a Laplace operator. Equation 2 is a parabolic partial differential equation, so that its form is invariant under the scale transformation $t \rightarrow \lambda^2 t$, $r \rightarrow \lambda r$ discussed above. This invariance of the equation has very strong implications. It follows, for example, that one can invert the scaling relation for the mean squared displacement

$$R^2 \propto t$$

and interpret this inverse one

$$T \propto r^2$$

as e. g. a scaling rule governing the dependence of a characteristic passage time from some initial site 0 to the sites at a distance r from this one. Such scaling holds e. g. for the mean time spent by a walk in a prescribed region of space (i. e. for the mean first passage time to its boundary). For an isotropic system Eq. (2) can be rewritten as an equation in the radial coordinate only:

$$\frac{\partial}{\partial t} P(r, t) = K \frac{1}{r^{d-1}} \frac{\partial}{\partial r} \left[r^{d-1} \frac{\partial}{\partial r} P(r, t) \right]. \quad (3)$$

Looking at the trajectory of a random walk at scales much larger than the step's length one infers that it is

self-similar in statistical sense, i. e. that its whatever portion (of the size considerably larger than the step's length) looks like the whole trajectory, i. e. it is a statistical, random fractal. The fractal (mass) dimension of the trajectory of a random walk can be easily found. Let us interpret the number of steps as the "mass" M of the object and $\langle \mathbf{r}_n^2 \rangle^{1/2}$ as its "size" L . The mass dimension D is then defined as $M \propto L^D$, it corresponds to the scaling of the mass of a solid D -dimensional object with its typical size. The scaling of the mean squared displacement of a simple random walk with its number of steps suggests that the corresponding dimension $D = d_w$ for a walk is $d_w = 2$. The large-scale properties of the random walks in Euclidean space are universal and do not depend on exact form of the PDF of waiting times and of jumps' lengths (provided the first possesses at least one and the second one at least two moments).

This fact can be used for an interpretation of several known properties of simple random walks. Thus, random walks on a one-dimensional lattice ($d = 1$) are recurrent, i. e. each site earlier visited by the walk is revisited repeatedly afterwards, while for $d \geq 3$ they are non-recurrent. This property can be rationalized through arguing that an intrinsically two-dimensional object cannot be "squeezed" into a one-dimensional space without self-intersections, and that it cannot fill fully a space of three or more dimensions. The same considerations can be used to rationalize the behavior of the mean number of different sites visited by a walker during n steps $\langle S_n \rangle$. $\langle S_n \rangle$ typically goes as a power law in n . Thus, on a one-dimensional lattice $\langle S_n \rangle \propto n^{1/2}$, in lattices in three and more dimensions $\langle S_n \rangle \propto n$, and in a marginal two-dimensional situation $\langle S_n \rangle \propto n / \log n$. To understand the behavior in $d = 1$ and in $d = 3$, it is enough to note that in a recurrent one-dimensional walk all sites within the span of the walk are visited by a walker at least once, so that the mean number of different sites visited by a one-dimensional walk ($d = 1$) grows as a span of the walk, which itself is proportional to a mean squared displacement, $\langle S_n \rangle \propto L = \langle \mathbf{r}_n^2 \rangle^{1/2}$, i. e. $\langle S_n \rangle \propto n^{1/2} = n^{1/d_w}$. In three dimensions ($d = 3$), on the contrary, the mean number of different visited sites grows as the number of steps, $\langle S_n \rangle \propto n$, since different sites are on the average visited only finite number of times. The property of random walks on one-dimensional and two-dimensional lattices to visit practically all the sites within the domain of the size of their mean square displacement is called the *compact visitation* property.

The importance of the number of different sites visited (and its moments like $\langle S_n \rangle$ or $\langle S(t) \rangle$) gets clear when one considers chemical reactions or search problems on the corresponding networks. The situation when each site of

a network can with the probability p be a trap for a walker, so that the walker is removed whenever it visits such a marked site (reaction scheme $A + B \rightarrow B$, where the symbol A denotes the walker and symbol B the “immortal” and immobile trap) corresponds to a trapping problem of reaction kinetics. The opposite situation, corresponding to the evaluation of the survival probability of an immobile A -particle at one of the network’s nodes in presence of mobile B -walkers (the same $A + B \rightarrow B$ reaction scheme, however now with immobile A and mobile B) corresponds to the target problem, or scavenging. In the first case the survival probability $\Phi(N)$ for the A -particle after N steps goes for p small as $\Phi(t) \propto \langle \exp(-pS_N) \rangle$, while for the second case it behaves as $\Phi(t) \propto \exp(-p\langle S_N \rangle)$. The simple $A + B \rightarrow B$ reaction problems can be also interpreted as the ones of the distribution of the times necessary to find one of multiple hidden “traps” by one searching agent, or as the one of finding one special marked site by a swarm of independent searching agents.

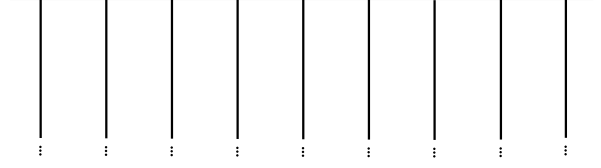
Anomalous Diffusion

The Einstein’s postulates leading to normal diffusion are to no extent the laws of Nature, and do not have to hold for a whatever random walk process. Before discussing general properties of such anomalous diffusion, we start from a few simple examples [10].

Simple Geometric Models Leading to Subdiffusion

Let us first discuss two simple models of geometrical structures showing anomalous diffusion. The discussion of these models allows one to gain some intuition about the emergence and properties of anomalous diffusion in complex geometries, as opposed to Euclidean lattices, and discuss mathematical notions appearing in description of such systems. In all cases the structure on which the walk takes place will be considered as a lattice with the unit spacing between the sites, and the walk corresponds to jumps from a site to one of its neighbors. Moreover, we consider the random walk to evolve in discrete time; the time t and the number of steps n are simply proportional to each other.

Diffusion on a Comb Structure The simplest example of how the subdiffusive motion can be created in a more or less complex (however yet non-fractal) geometrical structure is delivered by the diffusion on a comb. We consider the diffusion of a walker along the backbone of a comb with infinite “teeth” (dangling ends), see Fig. 1. The walker’s displacement in x -direction is only possible when it is on the backbone; entering the side branch switches



Anomalous Diffusion on Fractal Networks, Figure 1

A comb structure: Trapping in the dangling ends leads to the anomalous diffusion along the comb’s backbone. This kind of anomalous diffusion is modeled by CTRW with power-law waiting time distribution and described by a fractional diffusion equation

off the diffusion along the backbone. Therefore the motion on the backbone of the comb is interrupted by waiting times. The distribution of these waiting times is given by the power law $\psi(t) \propto t^{-3/2}$ corresponding to the first return probability of a one-dimensional random walk in a side branch to its origin. This situation – simple random walk with broad distribution of waiting times on sites – corresponds to a continuous-time random walk model. In this model, for the case of the power-law waiting time distributions of the form $\psi(t) \propto t^{-1-\alpha}$ with $\alpha < 1$, the mean number of steps $\langle n(t) \rangle$ (in our case: the mean number of steps over the backbone) as a function of time goes as $\langle n(t) \rangle \propto t^\alpha$. Since the mean squared displacement along the backbone is proportional to this number of steps, it is also governed by $\langle x^2(t) \rangle \propto t^\alpha$, i. e. in our case we have to do with strongly subdiffusive behavior $\langle x^2(t) \rangle \propto t^{1/2}$.

The PDF of the particle’s displacements in CTRW with the waiting time distribution $\psi(t)$ is governed by a non-Markovian generalization of the diffusion equation

$$\frac{\partial}{\partial t} P(x, t) = a^2 \frac{\partial}{\partial t} \Delta \int_0^\infty M(t - t') P(x, t') dt',$$

with the memory kernel $M(t)$ which has a physical meaning of the time-dependent density of steps and is given by the inverse Laplace transform of $\tilde{M}(u) = \tilde{\psi}(u)/[1 - \tilde{\psi}(u)]$ (with $\tilde{\psi}(u)$ being the Laplace-transform of $\psi(t)$). For the case of a power-law waiting-time probability density of the type $\psi(t) \propto t^{-1-\alpha}$ the structure of the right-hand side of the equation corresponds to the *fractional Riemann–Liouville derivative* ${}_0D_t^\beta$, an integro-differential operator defined as

$${}_0D_t^\beta f(t) = \frac{1}{\Gamma(1 - \beta)} \frac{d}{dt} \int_0^t dt' \frac{f(t')}{(t - t')^\beta}, \quad (4)$$

(here only for $0 < \beta < 1$), so that in the case of the diffusion on a backbone of the comb we have

$$\frac{\partial}{\partial t} P(x, t) = K_0 {}_0D_t^{1-\alpha} \Delta P(x, t), \quad (5)$$

(here with $\alpha = 1/2$). An exhaustive discussion of such equations is provided in [13,14]. Similar fractional diffusion equation was used in [12] for the description of the diffusion in a percolation system (as measured by NMR). However one has to be cautious when using CTRW and corresponding fractional equations for the description of diffusion on fractal networks, since they may capture some properties of such diffusion and fully disregard other ones. This question will be discussed to some extent in Sect. “Anomalous Diffusion on Fractal Structures”.

Equation (5) can be rewritten in a different but equivalent form

$${}_0^*D_t^\alpha \frac{\partial}{\partial t} P(x, t) = K \Delta P(x, t),$$

with the operator ${}_0^*D_t^\alpha$, a Caputo derivative, conjugated to the Riemann–Liouville one, Eq. (4), and different from Eq. (4) by interchanged order of differentiation and integration (temporal derivative inside the integral). The derivation of both forms of the fractional equations and their interpretation is discussed in [21]. For the sake of transparency, the corresponding integrodifferential operators ${}_0D_t^\beta$ and ${}_0^*D_t^\beta$ are simply denoted as a fractional partial derivative $\partial^\beta / \partial t^\beta$ and interpreted as a Riemann–Liouville or as a Caputo derivative depending whether they stand on the left or on the right-hand side of a generalized diffusion equation.

Random Walk on a Random Walk and on a Self-Avoiding Walk Let us now turn to another source of anomalous diffusion, namely the nondecaying correlations between the subsequent steps, introduced by the structure of the possible ways on a substrate lattice.

The case where the anomalies of diffusion are due to the tortuosity of the path between the two sites of a fractal “network” is illustrated by the examples of random walks on polymer chains, being rather simple, topologically one-dimensional objects. Considering lattice models for the chains we encounter two typical situations, the simpler one, corresponding to the random walk (RW) chains and a more complex one, corresponding to self-avoiding ones. A conformation of a RW-chain of N monomers corresponds to a random walk on a lattice, self-intersections are allowed. In reality, RW chains are a reasonable model for polymer chains in melts or in θ -solutions, where the repulsion between the monomers of the same chain is compensated by the repulsion from the monomers of other chains or from the solvent molecules, so that when not real intersections, then at least very close contacts between the monomers are possible. The other case corresponds to

chains in good solvents, where the effects of steric repulsion dominate. The end-to-end distance in a random walk chain grows as $R \propto l^{1/2}$ (with l being its contour length). In a self-avoiding-walk (SAW) chain $R \propto l^\nu$, with ν being a so-called Flory exponent (e.g. $\nu \approx 3/5$ in $3d$). The corresponding chains can be considered as topologically one-dimensional fractal objects with fractal dimensions $d_f = 2$ and $d_f = 1/\nu \approx 5/3$ respectively.

We now concentrate on a walker (excitation, enzyme, transcription factor) performing its diffusive motion on a chain, serving as a “rail track” for diffusion. This chain itself is embedded into the Euclidean space. The contour length of the chain corresponds to the *chemical coordinate* along the path, in each step of the walk at a chain the chemical coordinate changes by ± 1 . Let us first consider the situation when a particle can only travel along this chemical path, i.e. cannot change its track at a point of a self-intersection of the walk or at a close contact between the two different parts of the chain. This walk starts with its first step at $\mathbf{r} = 0$. Let K be the diffusion coefficient of the walker along the chain. The typical displacement of a walker along the chain after time t will then be $l \propto t^{1/2} = t^{1/d_w}$. For a RW-chain this displacement along the chain is translated into the displacement in Euclidean space according to $R = \langle \mathbf{R}^2 \rangle^{1/2} \propto l^{1/2}$, so that the typical displacement in Euclidean space after time t goes as $R \propto t^{1/4}$, a strongly subdiffusive behavior known as one of the regimes predicted by a reptation theory of polymer dynamics. The dimension of the corresponding random walk is, accordingly, $d_w = 4$. The same discussion for the SAW chains leads to $d_w = 1/2\nu$.

After time t the displacement in the chemical space is given by a Gaussian distribution

$$p_c(l, t) = \frac{1}{\sqrt{4\pi Kt}} \exp\left(-\frac{l^2}{4Kt}\right).$$

The change in the contour length l can be interpreted as the number of steps along the random-walk “rail”, so that the Euclidean displacement for the given l is distributed according to

$$p_E(\mathbf{r}, l) = \left(\frac{d}{2\pi a^2 |l|}\right) \exp\left(-\frac{dr^2}{2a^2 |l|}\right). \quad (6)$$

We can, thus, obtain the PDF of the Euclidean displacement:

$$P(\mathbf{r}, l) = \int_0^\infty p_E(\mathbf{r}, l) p_c(l, t) dl. \quad (7)$$

The same discussion can be pursued for a self-avoiding

walk for which

$$p_E(\mathbf{r}, l) \simeq \left(\frac{r}{l^\nu}\right)^{\frac{\gamma-1}{\nu}} \exp\left[-\text{const}\left(\frac{r}{l^\nu}\right)^{1/(1-\nu)}\right], \quad (8)$$

with γ being the exponent describing the dependence of the number of realizations of the corresponding chains on their lengths. Equation (6) is a particular case of Eq. (8) with $\gamma = 1$ and $\nu = 1/2$. Evaluating the integral using the Laplace method, we get

$$P(\mathbf{r}, t) \simeq \exp\left[-\text{const}\left(\frac{r}{t^{1/d_w}}\right)^{(1-1/d_w)^{-1}}\right]$$

up to the preexponential. We note that in this case the reason for the anomalous diffusion is the tortuosity of the chemical path. The topologically one-dimensional structure of the lattice allowed us for discussing the problem in considerable detail.

The situation changes when we introduce the possibility for a walker to jump not only to the nearest-neighbor in chemical space by also to the sites which are far away along the chemical sequence but close to each other in Euclidean space (say one lattice site of the underlying lattice far). In this case the structure of the chain in the Euclidean space leads to the possibility to jump very long in chemical sequence of the chain, with the jump length distribution going as $p(l) \propto l^{-3/2}$ for an RW and $p(l) \propto l^{-2.2}$ for a SAW in $d = 3$ (the probability of a close return of a SAW to its original path). The corresponding distributions lack the second moment (and even the first moment for RW chains), and therefore one might assume that also the diffusion in a chemical space of a chain will be anomalous. It indeed shows considerable peculiarities. If the chain's structure is frozen (i. e. the conformational dynamics of a chain is slow compared to the diffusion on the chain), the situation in both cases corresponds to "paradoxical diffusion" [22]: Although the PDF of displacements in the chemical space lacks the second (and higher) moments, the width of the PDF (described e. g. as its interquartile distance) grows, according to $W \propto t^{1/2}$. In the Euclidean space we have here to do with the diffusion on a static fractal structure, i. e. with a network of fractal dimension d_f coinciding with the fractal dimension of the chain. Allowing the chain conformation changing in time (i. e. considering the opposite limiting case when the conformation changes are fast enough compared to the diffusion along the chain) leads to another, superdiffusive, behavior, namely to Lévy flights along the chemical sequence. The difference between the static, fixed networks and the transient ones has to be borne in mind when interpreting physical results.

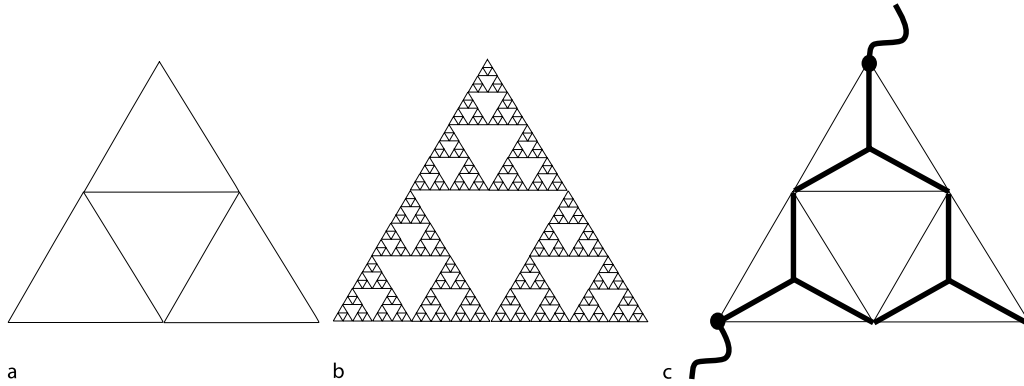
Anomalous Diffusion on Fractal Structures

Simple Fractal Lattices: the Walk's Dimension

As already mentioned, the diffusion on fractal structures is often anomalous, $\langle x^2(t) \rangle \propto t^\alpha$ with $\alpha \neq 1$. Parallel to the situation with random walks in Euclidean space, the trajectory of a walk is typically self similar (on the scales exceeding the typical step length), so that the exponent α can be connected with the fractal dimension of the walk's trajectory d_w . Assuming the mean time cost per step to be finite, i. e. $t \propto n$, one readily infers that $\alpha = 2/d_w$. The fractal properties of the trajectory can be rather easily checked, and the value of d_w can be rather easily obtained for simple regular fractals, the prominent example being a Sierpinski gasket, discussed below. Analytically, the value of d_w is often obtained via scaling of the first passage time to a boundary of the region (taken to be a sphere of radius r), i. e. using not the relation $R \propto t^{1/d_w}$ but the inverse one, $T \propto r^{d_w}$. The fact that both values of d_w coincide witnesses strongly in favor of the self-similar nature of diffusion on fractal structures.

Diffusion on a Sierpinski Gasket Sierpinski gasket (essentially a Sierpinski lattice), one of the simplest regular fractal network is a structure obtained by iterating a generator shown in Fig. 2. Its simple iterative structure and the ease of its numerical implementation made it a popular toy model exemplifying the properties of fractal networks. The properties of diffusion on this network were investigated in detail, both numerically and analytically. Since the structure does not possess dangling ends, the cause of the diffusion anomalies is connected with tortuosity of the typical paths available for diffusion, however, differently from the previous examples, these paths are numerous and non-equivalent.

The numerical method allowing for obtaining exact results on the properties of diffusion is based on exact enumeration techniques, see e. g. [10]. The idea here is to calculate the displacement probability distribution based on the exact number of ways $W_{i,n}$ the particle, starting at a given site 0 can arrive to a given site i at step n . For a given network the number $W_{i,n}$ is simply a sum of the numbers of ways $W_{j,n-1}$ leading from site 0 to the nearest neighbors j of the site i , $W_{j,n-1}$. Starting from $W_{0,0} = 1$ and $W_{i,0} = 0$ for $i \neq 0$ one simply updates the array of $W_{i,n}$ according to the rule $W_{i,n} = \sum_{j=nn(i)} W_{j,n-1}$, where $nn(i)$ denotes the nearest neighbors of the node i in the network. The probability $P_{i,n}$ is proportional to the number of these equally probably ways, and is obtained from $W_{i,n}$ by normalization. This is essentially a very old method used to solve numerically a diffusion equation

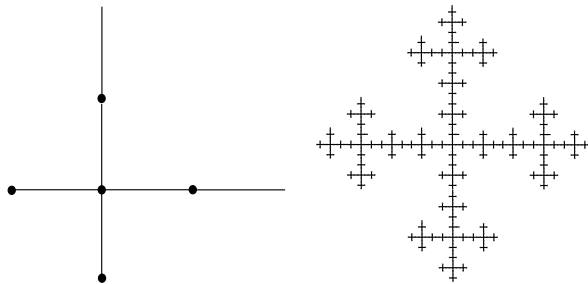


Anomalous Diffusion on Fractal Networks, Figure 2

A Sierpinski gasket: a its generator b a structure after 4th iteration. The lattice is rescaled by the factor of 4 to fit into the picture. c A constriction obtained after triangle-star transformation used in the calculation of the walk's dimension, see Subsect. "The Spectral Dimension"

in the times preceding the computer era. For regular lattices this gives us the exact values of the probabilities. For statistical fractals (like percolation clusters) an additional averaging over the realizations of the structure is necessary, so that other methods of calculation of probabilities and exponents might be superior. The value of d_w can then be obtained by plotting $\ln n$ vs. $\ln(\mathbf{r}_n^2)^{1/2}$: the corresponding points fall on a straight line with slope d_w . The dimension of the walks on a Sierpinski gasket obtained by direct enumeration coincides with its theoretical value $d_w = \ln 5 / \ln 2 = 2.322 \dots$, see Subsect. "The Spectral Dimension".

Loopless Fractals (Trees) Another situation, the one similar to the case of a comb, is encountered in loopless fractals (fractal trees), as exemplified by a Vicsek's construction shown in Fig. 3, or by diffusion-limited aggregates (DLA). For the structure depicted in Fig. 3, the walk's



Anomalous Diffusion on Fractal Networks, Figure 3

A Vicsek fractal: the generator (left) and its 4th iteration. This is a loopless structure; the diffusion anomalies here are mainly caused by trapping in dangling ends

dimension is $d_w = \log 15 / \log 3 = 2.465 \dots$ [3]. Parallel to the case of the comb, the main mechanism leading to subdiffusion is trapping in dangling ends, which, different from the case of the comb, now themselves have a fractal structure. Parallel to the case of the comb, the time of travel from one site of the structure to another one is strongly affected by trapping. However, trapping inside the dangling end does not lead to a halt of the particle, but only to confining its motion to some scale, so that the overall equations governing the PDF might be different in the form from the fractional diffusion ones.

Diffusion and Conductivity

The random walks of particles on a network can be described by a master equation (which is perfectly valid for the exponential waiting time distribution on a site and asymptotically valid in case of all other waiting time distributions with finite mean waiting time τ): Let $\mathbf{p}(t)$ be the vector with elements $p_i(t)$ being the probabilities to find a particle at node i at time t . The master equation

$$\frac{d}{dt} \mathbf{p} = \mathbf{W} \mathbf{p} \quad (9)$$

gives then the temporal changes in this probability. A similar equation with the temporal derivative changed to d/dn describes the n -dependence for the probabilities $p_{i,n}$ in a random walk as a function of the number of steps, provided n is large enough to be considered as a continuous variable. The matrix \mathbf{W} describes the transition probabilities between the nodes of the lattice or network. The non-diagonal elements of the corresponding matrix are w_{ij} , the transition probabilities from site i to site j per unit time

or in one step. The diagonal elements are the sums of all non-diagonal elements in the corresponding column taken with the opposite sign: $w_{ii} = -\sum_j w_{ji}$, which represents the probability conservation law. The situation of unbiased random walks corresponds to a symmetric matrix \mathbf{W} : $w_{ij} = w_{ji}$. Considering homogeneous networks and putting all nonzero w_{ij} to unity, one sees that the difference operator represented by each line of the matrix is a symmetric difference approximation to a Laplacian.

The diffusion problem is intimately connected with the problem of conductivity of a network, as described by Kirchhoff's laws. Indeed, let us consider a stationary situation in which the particles enter the network at some particular site A at constant rate, say, one per unit time, and leave it at some site B (or a given set of sites B). Let us assume that after some time a stationary distribution of the particles over the lattice establishes, and the particles' concentration on the sites will be described by the vector proportional to the vector of stationary probabilities satisfying

$$\mathbf{W}\mathbf{p} = 0. \quad (10)$$

Calculating the probabilities \mathbf{p} corresponds then formally to calculating the voltages on the nodes of the resistor network of the same geometry under given overall current using the Kirchhoff's laws. The conductivities of resistors connecting nodes i and j have to be taken proportional to the corresponding transition probabilities w_{ij} . The condition given by Eq. (10) corresponds then to the second Kirchhoff's law representing the particle conservation (the fact that the sum of all currents to/from the node i is zero), and the fact that the corresponding probability current is proportional to the probability difference is replaced by the Ohm's law. The first Kirchhoff's law follows from the uniqueness of the solution. Therefore calculation of the dimension of a walk can be done by discussing the scaling of conductivity with the size of the fractal object. This typically follows a power law.

As an example let us consider a Sierpinski lattice and calculate the resistance between the terminals A and B (depicted by thick wires outside of the triangle in Fig. 2c) of a fractal of the next generation, assuming that the conductivity between the corresponding nodes of the lattice of the previous generation is $R = 1$. Using the triangle-star transformation known in the theory of electric circuits, i.e. passing to the structure shown in Fig. 2c by thick lines inside the triangle, with the conductivity of each bond $r = 1/2$ (corresponding to the same value of the conductivity between the terminals), we get the resistance of a renormalized structure $R' = 5/3$. Thus, the dependence of R on the spacial scale L of the object is $R \propto L^\zeta$ with

$\zeta = \log(5/3)/\log 2$. The scaling of the conductivity G is correspondingly $G \propto L^{-\zeta}$.

Using the flow-over-population approach, known in calculation of the mean first passage time in a system, we get $I \propto N/\langle t \rangle$, where I is the probability current through the system (the number of particles entering A per unit time), N is the overall stationary number of particles within the system and $\langle t \rangle$ is the mean time a particle spends inside the system, i.e. the mean first passage time from A to B. The mean number of particles inside the system is proportional to a typical concentration (say to the probability p_A to find a particle at site A for the given current I) and to the number of sites. The first one, for a given current, scales as the system's resistivity, $p_A \propto R \propto L^\zeta$, and the second one, clearly, as L^{d_f} where L is the system's size. On the other hand, the mean first passage time scales according to $\langle t \rangle \propto L^{d_w}$ (this time corresponds to the typical number of steps during which the walk transverses an Euclidean distance L), so that

$$d_w = d_f + \zeta. \quad (11)$$

Considering the d -dimensional generalizations of Sierpinski gaskets and using analogous considerations we get $\zeta = \log[(d+3)/(d+1)]/\log 2$. Combining this with the fractal dimension $d_f = \log(d+1)/\log 2$ of a gasket gives us for the dimension of the walks $d_w = \log(d+3)/\log 2$.

The relation between the scaling exponent of the conductivity and the dimension of a random walk on a fractal system can be used in the opposite way, since d_w can easily be obtained numerically for a whatever structure. On the other hand, the solutions of the Kirchhoff's equations on complex structures (i.e. the solution of a large system of algebraic equations), which is typically achieved using relaxation algorithms, is numerically much more involved.

We note that the expression for ζ can be rewritten as $\zeta = d_f(2/d_s - 1)$, where d_s is the spectral dimension of the network, which will be introduced in Subsect. "The Spectral Dimension". The relation between the walk dimension and this new quality therefore reads

$$d_w = \frac{2d_f}{d_s}. \quad (12)$$

A different discussion of the same problem based on the Einstein's relation between diffusion coefficient and conductivity, and on crossover arguments can be found in [10,20].

The Spectral Dimension

In mathematical literature spectral dimension is defined through the return probability of the random walk, i.e. the

probability to be at the origin after n steps or at time t . We consider a node on a network and take it to be an origin of a simple random walk. We moreover consider the probability $P(\mathbf{0}, t)$ to be at this node at time t (a return probability). The spectral dimension defines then the asymptotic behavior of this probability for n large:

$$P(\mathbf{0}, t) \propto t^{-d_s/2}, \quad \text{for } t \rightarrow \infty. \quad (13)$$

The spectral dimension d_s is therefore exactly a quantity substituting the Euclidean dimension d is Eq. (1). In the statistical case one should consider an average of $p(t)$ over the origin of the random walk and over the ensemble of the corresponding graphs.

For fixed graphs the spectral dimension and the fractal (Hausdorff) dimension are related by

$$\frac{2d_f}{1+d_f} \leq d_s \leq d_f, \quad (14)$$

provided both exist [5]. The same relation is also shown true for some random geometries, see e. g. [6,7] and references therein. The bounds are optimal, i. e. both equalities can be realized in some structures.

The connection between the walk dimension d_w and the spectral dimension d_s of a network can be easily rationalized by the following consideration. Let the random walk have a dimension d_w , so that after time t the position of the walker can be considered as more or less homogeneously spread within the spatial region of the linear size $R \propto n^{1/d_w}$ or $R \propto t^{1/d_w}$, with the overall number of nodes $N \propto R^{d_f} \propto t^{d_f/d_w}$. The probability to be at one particular node, namely at $\mathbf{0}$, goes then as $1/N$, i. e. as t^{-d_f/d_w} . Comparing this with the definition of the spectral dimension, Eq. (13) we get exactly Eq. (12). The lower bound in the inequality Eq. (14) follows from Eq. (13) and from the observation that the value of ζ in Eq. (11) is always smaller or equal to one (the value of $\zeta = 1$ would correspond to a one-dimensional wire without shunts). The upper bound can be obtained using Eq. (12) and noting that the walks' dimension never gets less than 2, its value for the regular network: the random walk on a whatever structure is more compact than one on a line.

We note that the relation Eq. (12) relies on the assumption that the spread of the walker within the r -domain is homogeneous, and needs reinterpretation for strongly anisotropic structures like combs, where the spread in teeth and along the spine are very different (see e. g. [2]). For the planar comb with infinite teeth $d_f = 2$, and d_s as calculated through the return time is $d_s = 3/2$, while the walk's dimension (mostly determined by the motion in teeth) is $d_w = 2$, like for random walks in Euclidean space. The inequality Eq. (14) is, on the contrary, universal.

Let us discuss another meaning of the spectral dimension, the one due to which it got its name. This one has to do with the description of random walks within the master equation scheme. From the spectral (Laplace) representation of the solution of the master equation, Eq. (9), we can easily find the probability $P(\mathbf{0}, t)$ that the walker starting at site $\mathbf{0}$ at $t = 0$ is found at the same site at time t . This one reads

$$P(\mathbf{0}, t) = \sum_{i=1}^{\infty} a_i \exp(-\epsilon_i t),$$

where ϵ_i is the i th eigenvalue of the matrix \mathbf{W} and a_i is the amplitude of its i th eigenvector at site $\mathbf{0}$. Considering the lattice as infinite, we can pass from discrete eigenvalue decomposition to a continuum

$$P(\mathbf{0}, t) = \int_0^{\infty} \mathcal{N}(\epsilon) a(\epsilon) \exp(-\epsilon t) d\epsilon.$$

For long times, the behavior of $P(\mathbf{0}, t)$ is dominated by the behavior of $\mathcal{N}(\epsilon)$ for small values of ϵ . Here $\mathcal{N}(\epsilon)$ is the density of states of a system described by the matrix \mathbf{W} . The exact forms of such densities are well known for many Euclidean lattices, since the problem is equivalent to the calculating of spectrum in tight-binding approximation used in the solid state physics. For all Euclidean lattices $\mathcal{N}(\epsilon) \propto \epsilon^{d/2-1}$ for $\epsilon \rightarrow 0$, which gives us the forms of famous van Hove singularities of the spectrum. Assuming $a(\epsilon)$ to be nonsingular at $\epsilon \rightarrow 0$, we get $P(\mathbf{0}, t) \propto t^{-d/2}$. For a fractal structure, the value of d changed for the one of the spectral dimension d_s . This corresponds to the density of states $\mathcal{N}(\epsilon) \propto \epsilon^{d_s/2-1}$ which describes the properties of spectrum of a fractal analog of a Laplace operator. The dimension d_s is also often called fracton dimension of the structure, since the corresponding eigenvectors of the matrix (corresponding to eigenstates in tight-binding model) are called fractons, see e. g. [16].

In the examples of random walks on quenched polymer chains the dimension d_w was twice the fractal dimension of the underlying structures, so that the spectral dimension of the corresponding structures (without inter-sections) was exactly 1 just like their topological dimension.

The spectral dimension of the network governs also the behavior of the mean number of different sites visited by the random walk. A random walk of n steps having a property of compact visitation typically visits all sites within the radius of the order of its typical displacement $R_n \propto n^{1/d_w}$. The number of these sites is $S_n \propto R_n^{d_f}$ where d_f is the fractal dimension of the network, so that $S_n \propto n^{d_s/2}$ (provided $d_s \leq 2$, i. e. provided the random

walk is recurrent and shows compact visitation). The spectral dimension of a structure plays important role in many other applications [16]. This and many other properties of complex networks related to the spectrum of \mathbf{W} can often be easier obtained by considering random walk on the lattice than by obtaining the spectrum through direct diagonalization.

Percolation Clusters

Percolation clusters close to criticality are one of the most important examples of fractal networks. The properties of these clusters and the corresponding fractal and spectral dimensions are intimately connected to the critical indices in percolation theory.

Thus, simple crossover arguments show that the fractal dimension of an incipient infinite percolation cluster in $d \leq 6$ dimensions is $d_f = d - \beta/\nu$, where β is the critical exponent of the density of the infinite cluster, $P_\infty \propto (p - p_c)^\beta$ and ν is the critical exponent of the correlation length, $\xi \propto |p - p_c|^\nu$, and stagnates at $d_f = 4$ for $d > 6$, see e. g. [23].

On the other hand, the critical exponent t , describing the behavior of the resistance of a percolation system close to percolation threshold, $\rho \propto (p - p_c)^{-t}$, is connected with the exponent ζ via the following crossover argument. The resistance of the system is the one of the infinite cluster. For $p > p_c$ this cluster can be considered as fractal at scales $L < \xi$ and as homogeneous at larger scales. Our value for the fractal dimension of the cluster follows exactly from this argument by matching the density of the cluster $P(L) \propto L^{(d_f-d)}$ at $L = \xi \propto (p - p_c)^{-\nu}$ to its density $P_\infty \propto (p - p_c)^\beta$ at larger scales. The infinite cluster for $p > p_c$ is then a dense regular assembly of subunits of size ξ which on their turn are fractal. If the typical resistivity of each fractal subunit is R_ξ then the resistivity of the overall assembly goes as $R \propto R_\xi(L/\xi)^{2-d}$. This is a well-known relation showing that the resistivity of a wire grows proportional to its length, the resistivities of similar flat figures are the same etc. This relation holds in the homogeneous regime. On the other hand, in the fractal regime $R_L \propto L^\zeta$, so that overall $R \propto \xi^{\zeta+d-2} \propto (p - p_c)^{-\nu(\zeta+d-2)}$ giving the value of the critical exponent $t = \nu(\zeta + d - 2)$. The value of ζ can in its turn be expressed through the values of spectral and fractal dimensions of a percolation cluster.

The spectral dimension of the percolation cluster is very close to $4/3$ in any dimension larger than one. This surprising finding lead to the conjecture by Alexander and Orbach that this value of $d_s = 4/3$ might be exact [1] (it is exact for percolation systems in $d \geq 6$ and for trees). Much

effort was put into proving or disproving the conjecture, see the discussion in [10]. The latest, very accurate simulations of two-dimensional percolation by Grassberger show that the conjecture does not hold in $d = 2$, and the prediction $d_s = 4/3$ is off by around 2% [9]. In any case it can be considered as a very useful mnemonic rule.

Anomalous diffusion on percolation clusters corresponds theoretically to the most involved case, since it combines all mechanisms generating diffusion anomalies. The infinite cluster of the percolation system consists of a backbone, its main current-carrying structure, and the dangling ends (smaller clusters on all scales attached to the backbone through only one point). The anomalous diffusion on the infinite cluster is thus partly caused by trapping in this dangling ends. If one considers a shortest (chemical) way between the two nodes on the cluster, this way is tortuous and has fractal dimension larger than one.

The Role of Finite Clusters

The exponent t of the conductivity of a percolation system is essentially the characteristic of an infinite cluster only. Depending on the physical problem at hand, one can consider the situations when only the infinite cluster plays the role (like in the experiments of [12], where only the infinite cluster is filled by fluid pressed through the boundary of the system), and situations when the excitations can be found in infinite as well as in the finite clusters, as it is the case for optical excitation in mixed molecular crystals, a situation discussed in [24].

Let us concentrate on the case $p = p_c$. The structure of large but finite clusters at lower scales is indistinguishable from those of the infinite cluster. Thus, the mean squared displacement of a walker on a cluster of size L (one with $N \propto L^{d_f}$ sites) grows as $R^2(t) \propto t^{2/d_w}$ until it stagnates at the value of R of the order of the cluster's size, $R^2 \propto N^{2/d_f}$. At a given time t we can subdivide all clusters into two classes: those whose size is small compared to t^{1/d_w} (i. e. the ones with $N < t^{d_s/2}$), on which the mean square displacement stagnates, and those of the larger size, on which it still grows:

$$R^2(t) \propto \begin{cases} t^{2/d_w}, & t^{1/d_w} < N^{1/d_f} \\ N^{2/d_f}, & \text{otherwise.} \end{cases}$$

The characteristic size $N_{\text{cross}}(t)$ corresponds to the crossover between these two regimes for a given time t . The probability that a particle starts at a cluster of size N is proportional to the number of its sites and goes as $P(N) \propto Np(N) \propto N^{1-\tau}$ with $\tau = (2d\nu - \beta)/(d\nu - \beta)$, see e. g. [23]. Here $p(N)$ is the probability to find a cluster

of N sites among all clusters. The overall mean squared displacement is then given by averaging over the corresponding cluster sizes:

$$\langle r^2(t) \rangle \propto \sum_{N=1}^{N_{\text{cross}}(t)} N^{2/d_f} N^{1-\tau} + \sum_{N_{\text{cross}}(t)}^{\infty} t^{2/d_w} N^{1-\tau}.$$

Introducing the expression for $N_{\text{cross}}(t)$, we get

$$\langle r^2(t) \rangle \propto t^{2/d_w + (2-\tau)d_f/d_w} = t^{(1/d_w)(2d_f - \beta/\nu)}.$$

A similar result holds for the number of distinct sites visited, where one has to perform the analogous averaging with

$$S(t) \propto \begin{cases} t^{d_s/2}, & t^{1/d_w} < N^{1/d_f} \\ N, & \text{otherwise} \end{cases}$$

giving $S(t) \propto t^{(d_s/2)(1-\beta\nu/d_f)}$. This gives us the effective spectral dimension of the system $\tilde{d}_s = (d_s/2)(1 - \beta\nu/d_f)$.

Scaling of PDF and Diffusion Equations on Fractal Lattices

If the probability density of the displacement of a particle on a fractal scales, the overall scaling form of this displacement can be obtained rather easily: Indeed, the typical displacement R during the time t goes as $r \propto t^{1/d_w}$, so that the corresponding PDF has to scale as

$$P(r, t) = \frac{r^{d_f-d}}{t^{d_s/2}} f\left(\frac{r}{t^{1/d_w}}\right). \quad (15)$$

The prefactor takes into account the normalization of the probability density on a fractal structure (denominator) and also the scaling of the density of the fractal structure in the Euclidean space (enumerator). The simple scaling assumes that all the moments of the distance scale in the same way, so that $\langle r^k \rangle = \int P(r, t) r^k d^d r^k = \text{const}(k) t^{k/d_w}$. The overall scaling behavior of the PDF was confirmed numerically for many fractal lattices like Sierpinski gaskets and percolation clusters, and is corroborated in many other cases by the coincidence of the values of d_w obtained via calculation of the mean squared displacement and of the mean first passage time (or resistivity of the network). The existence of the scaling form leads to important consequences e.g. for quantification of anomalous diffusion in biological tissues by characterizing the diffusion-time dependence of the magnetic resonance signal [17], which, for example, allows for differentiation between the healthy and the tumor tissues. However, even in the cases when the scaling form Eq. (15) is believed to

hold, the exact form of the scaling function $f(x)$ is hard to get; the question is not yet resolved even for simple regular fractals.

For applications it is often interesting to have a kind of a phenomenological equation roughly describing the behavior of the PDF. Such an approach will be analogous to putting down Richardson's equation for the turbulent diffusion. Essentially, the problem here is to find a correct way of averaging or coarse-graining the microscopic master equation, Eq. (9), to get its valid continuous limit on larger scales. The regular procedure to do this is unknown; therefore, several phenomenological approaches to the problem were formulated.

We are looking for an equation for the PDF of the walker's displacement from its initial position and assume the system to be isotropic on the average. We look for an analog of the classical Fick's equation, $\frac{\partial}{\partial t} P(r, t) = \nabla K(r) \nabla P(r, t)$ which in spherical coordinates and for $K = K(r)$ takes the form of Eq. (3):

$$\frac{\partial}{\partial t} P(r, t) = \frac{1}{r^{d-1}} \frac{\partial}{\partial r} \left[r^{d-1} K(r) \frac{\partial}{\partial r} P(r, t) \right].$$

On fractal lattices one changes from the Euclidean dimension of space to the fractal dimension of the lattice d_f , and takes into account that the effective diffusion coefficient $K(r)$ decays with distance as $K(r) \simeq K r^{-\theta}$ with $\theta = d_w - 2$ to capture the slowing down of anomalous diffusion on a fractal compared to the Euclidean lattice situation. The corresponding equation put forward by O'Shaughnessy and Procaccia [18] reads:

$$\frac{\partial}{\partial t} P(r, t) = K \frac{1}{r^{d_f-1}} \frac{\partial}{\partial r} \left[r^{d_f-d_w+1} \frac{\partial}{\partial r} P(r, t) \right]. \quad (16)$$

This equation was widely used in description of anomalous diffusion on fractal lattices and percolation clusters, but can be considered only as a rough phenomenological approximation. Its solution

$$P(r, t) = A \frac{r^{d_f-d}}{t^{d_s/2}} \exp\left(-B \frac{r^{d_w}}{t}\right),$$

(with $B = 1/Kd_w^2$ and A being the corresponding normalization constant) corresponds exactly to the type of scaling given by Eq. (15), with the scaling function $f(x) = \exp(x^{d_w})$. This behavior of the scaling function disagrees e.g. with the results for random walks on polymer chains, for which case we had $f(x) = \exp(x^\mu)$ with $\mu = (1 - 1/d_w)^{-1}$ for large enough x .

In literature, several other proposals (based on plausible assumptions but to no extent following uniquely from

the models considered) were made, taking into account possible non-Markovian nature of the motion. These are the fractional equations of the type

$$\frac{\partial^{1/d_w}}{\partial t^{1/d_w}} P(r, t) = -K_1 \frac{1}{r^{(d_s-1)/2}} \frac{\partial}{\partial r} \left[r^{(d_s-1)/2} P(r, t) \right], \quad (17)$$

resembling “half” of the diffusion equation and containing a fractional derivative [8], as well as the “full” fractional diffusion equation

$$\frac{\partial^{2/d_w}}{\partial t^{2/d_w}} P(r, t) = K_2 \frac{1}{r^{d_s-1}} \frac{\partial}{\partial r} \left[r^{d_s-1} \frac{\partial}{\partial r} P(r, t) \right], \quad (18)$$

as proposed in [15]. All these equations are invariant under the scale transformation

$$t \rightarrow \lambda t, \quad r \rightarrow \lambda^{1/d_w} r,$$

and lead to PDFs showing correct overall scaling properties, see [19]. None of them reproduces correctly the PDF of displacements on a fractal (i. e. the scaling function $f(x)$) in the whole range of distances. Ref. [19] comparing the corresponding solutions with the results of simulation of anomalous diffusion on a Sierpinski gasket shows that the O’Shaughnessy–Procaccia equation, Eq. (16), performs the best for the central part of the distribution (small displacements), where Eq. (18) overestimates the PDF and Eq. (17) shows an unphysical divergence. On the other hand, the results of Eqs. (17) and (18) reproduce equally well the PDF’s tail for large displacements, while Eq. (16) leads to a PDF decaying considerably faster than the numerical one. This fact witnesses favor of strong non-Markovian effects in the fractal diffusion, however, the physical nature of this non-Markovian behavior observed here in a fractal network without dangling ends, is not as clear as it is in a comb model and its more complex analogs. Therefore, the question of the correct equation describing the diffusion in fractal system (or different correct equations for the corresponding different classes of fractal systems) is still open. We note also that in some cases a simple fractional diffusion equation (with the corresponding power of the derivative leading to the correct scaling exponent d_w) gives a reasonable approximation to experimental data, as the ones of the model experiment of [12].

Further Directions

The physical understanding of anomalous diffusion due to random walks on fractal substrates may be considered as rather deep and full although it does not always lead to simple pictures. For example, although the spectral dimen-

sion for a whatever graph can be rather easily calculated, it is not quite clear what properties of the graph are responsible for its particular value. Moreover, there are large differences with respect to the degree of rigor with which different statements are proved. Mathematicians have recognized the problem, and the field of diffusion on fractal networks has become a fruitful field of research in probability theory. One of the examples of recent mathematical development is the deep understanding of spectral properties of fractal lattices and a proof of inequalities like Eq. (14).

A question which is still fully open is the one of the detailed description of the diffusion within a kind of (generalized) diffusion equation. It seems clear that there is more than one type of such equations depending on the concrete fractal network described. However even the classification of possible types of such equations is still missing.

All our discussion (except for the discussion of the role of finite clusters in percolation) was pertinent to infinite structures. The recent work [4] has showed that finite fractal networks are interesting on their own and opens a new possible direction of investigations.

Bibliography

Primary Literature

1. Alexander S, Orbach RD (1982) Density of states on fractals—fractons. *J Phys Lett* 43:L625–L631
2. Bertacci D (2006) Asymptotic behavior of the simple random walk on the 2-dimensional comb. *Electron J Probab* 45:1184–1203
3. Christou A, Stinchcombe RB (1986) Anomalous diffusion on regular and random models for diffusion-limited aggregation. *J Phys A Math Gen* 19:2625–2636
4. Condamin S, Bénichou O, Tejedor V, Voituriez R, Klafter J (2007) First-passage times in complex scale-invariant media. *Nature* 450:77–80
5. Coulhon T (2000) Random Walks and Geometry on Infinite Graphs. In: Ambrosio L, Cassano FS (eds) *Lecture Notes on Analysis on Metric Spaces*. Trento, CIMR, (1999) Scuola Normale Superiore di Pisa
6. Durhuus B, Jonsson T, Weather JF (2006) Random walks on combs. *J Phys A Math Gen* 39:1009–1037
7. Durhuus B, Jonsson T, Weather JF (2007) The spectral dimension of generic trees. *J Stat Phys* 128:1237–1260
8. Giona M, Roman HE (1992) Fractional diffusion equation on fractals – one-dimensional case and asymptotic-behavior. *J Phys A Math Gen* 25:2093–2105; Roman HE, Giona M, Fractional diffusion equation on fractals – 3-dimensional case and scattering function, *ibid.*, 2107–2117
9. Grassberger P (1999) Conductivity exponent and backbone dimension in 2- d percolation. *Physica A* 262:251–263
10. Havlin S, Ben-Avraham D (2002) Diffusion in disordered media. *Adv Phys* 51:187–292
11. Klafter J, Sokolov IM (2005) Anomalous diffusion spreads its wings. *Phys World* 18:29–32

12. Klemm A, Metzler R, Kimmich R (2002) Diffusion on random-site percolation clusters: Theory and NMR microscopy experiments with model objects. *Phys Rev E* 65:021112
13. Metzler R, Klafter J (2000) The random walk's guide to anomalous diffusion: a fractional dynamics approach. *Phys Rep* 339:1–77
14. Metzler R, Klafter J (2004) The restaurant at the end of the random walk: recent developments in the description of anomalous transport by fractional dynamics. *J Phys A Math Gen* 37:R161–R208
15. Metzler R, Glöckle WG, Nonnenmacher TF (1994) Fractional model equation for anomalous diffusion. *Physica A* 211:13–24
16. Nakayama T, Yakubo K, Orbach RL (1994) Dynamical properties of fractal networks: Scaling, numerical simulations, and physical realizations. *Rev Mod Phys* 66:381–443
17. Özarslan E, Bassier PJ, Shepherd TM, Thelwall PE, Vemuri BC, Blackband SJ (2006) Observation of anomalous diffusion in excised tissue by characterizing the diffusion-time dependence of the MR signal. *J Magn Res* 183:315–323
18. O'Shaughnessy B, Procaccia I (1985) Analytical solutions for diffusion on fractal objects. *Phys Rev Lett* 54:455–458
19. Schulzky C, Essex C, Davidson M, Franz A, Hoffmann KH (2000) The similarity group and anomalous diffusion equations. *J Phys A Math Gen* 33:5501–5511
20. Sokolov IM (1986) Dimensions and other geometrical critical exponents in the percolation theory. *Usp Fizicheskikh Nauk* 150:221–255 (1986) translated in: *Sov. Phys. Usp.* 29:924
21. Sokolov IM, Klafter J (2005) From diffusion to anomalous diffusion: A century after Einstein's Brownian motion. *Chaos* 15:026103
22. Sokolov IM, Mai J, Blumen A (1997) Paradoxical diffusion in chemical space for nearest-neighbor walks over polymer chains. *Phys Rev Lett* 79:857–860
23. Stauffer D (1979) Scaling theory of percolation clusters. *Phys Rep* 54:1–74
24. Webman I (1984) Diffusion and trapping of excitations on fractals. *Phys Rev Lett* 52:220–223
- Haus JW, Kehr K (1987) Diffusion in regular and disordered lattices. *Phys Rep* 150:263–406
- Bouchaud JP, Georges A (1990) Anomalous diffusion in disordered media – statistical mechanisms, models and physical applications. *Phys Rep* 195:127–293
- Stauffer D, Aharony A (2003) *Introduction to Percolation Theory*. Taylor & Francis, London
- Isichenko MB (1992) *Percolation, statistical topography, and transport in random-media*. *Rev Mod Phys* 64:961–1043
- Hughes BD (1995) *Random Walks and random Environments*. Oxford University Press, New York

Applications of Physics and Mathematics to Social Science, Introduction to

ANDRZEJ NOWAK¹, URSZULA STRAWIŃSKA²

¹ Institute for Social Studies, University of Warsaw, Warsaw, Poland

² Warsaw School for Social Psychology, Warsaw, Poland

The human mind is often regarded as the most complex structure in the universe. If anything could be argued to be of higher complexity it is a collection of interacting minds: social groups and societies. Kluver argues that social cognitive complexity (see ► [Social Cognitive Complexity](#)) stems from the fact that such systems consist of multiple (at least two – social and cognitive) levels that are constantly at flux due to the multiple influences both between and within levels.

The complexity of the subject matter of the social sciences has made adequate description using the traditional models of mathematics and physics difficult. For a long time it has been argued that tools of mathematics and physics are not adequate for social processes and therefore science had to proceed along two independent fronts that seemed irreconcilable (Snow, 1993). One approach, epitomized by such disciplines as sociology and anthropology, was characterized by deep insights into the nature of psychological social phenomena. These insights reflected socially constructed metrics and meanings that emphasized the contextual aspects of human experience. In the other scientific culture, epitomized in the physical sciences by precise measurement, qualitative laws and formal models defined the paradigm within which the invariant and general features of phenomena are understood.

The bridge between these two approaches consisted mainly of statistical models of psychological and social phenomena and models based on the assumption that rational choice underlies human behavior. The discoveries in the late 70s and 80s of the physical and mathematical sciences have produced a profound change in the social

Additional Reading

The present article gave a brief overview of what is known about the diffusion on fractal networks, however this overview is far from covering all the facets of the problem. Thus, we only discussed unbiased diffusion (the effects of bias may be drastic due to e.g. stronger trapping in the dangling ends), and considered only the situations in which the waiting time at all nodes was the same (we did not discuss e.g. the continuous-time random walks on fractal networks), as well as left out of attention many particular systems and applications. Several review articles can be recommended as a further reading, some of them already mentioned in the text. One of the best-known sources is [10] being a reprint of the text from the “sturm und drang” period of investigation of fractal geometries. A lot of useful information on random walks models in general and on walks on fractals is contained in the review by Haus and Kehr from approximately the same time. General discussion of the anomalous diffusion is contained in the work by Bouchaud and Georges. The classical review of the percolation theory is given in the book of Stauffer and Aharony. Some additional information on anomalous diffusion in percolation systems can be found in the review by Isichenko. A classical source on random walks in disordered systems is the book by Hughes.

sciences by providing concepts and tools that integrate the deep understanding inherent in social sciences with the precision of mathematics and physics. This perspective has provided solutions for many apparent paradoxes of social sciences, such as how altruism can emerge in self-centered individuals. It has led to surprising insights, such as the realization that the avoidance by individuals of being in the minority leads to full segregation at the level of social groups.

The core of the complexity approach into social domain is the application of mathematics and physics. These applications, described and explained by Solomon and Stauffer (see ► [Applications of Physics and Mathematics to Social Science, Introduction to](#)), are often based on the Ising model, where a binary state of an element corresponds to individual opinions (e.g. yes-no opinions), choices (e.g. cooperate or compete), or behaviors. The binary representations of individuals are arranged into structures such as a grid structure of cellular automata or networks to represent the structure of interactions of social groups and societies. The characteristics of every individual, such as opinions, depend on “neighbors”: adjacent cells in a lattice, or linked nodes in a network. The tools of statistical physics, especially the mean field approach, may be used to analyze dynamics of such systems. The dynamics of such a system is often characterized by dramatic changes, phase transitions in response to small and continuous changes in the factors that influence the system (such as percolation). Physical models of social processes, described by Slanina (see ► [Social Processes, Physical Models of](#)) concentrate on the link between the micro (individual) and the macro (societies, social groups, organizations) levels of description of social reality. This approach can explain how cooperation emerges among self-centered individuals, how social classes are formed, and explains properties of behaviors stemming from bounded confidence models.

In models of opinion dynamics explained by Stauffer (see ► [Opinion Dynamics and Sociophysics](#)), individuals either move in the social space, usually seeking others with similar opinions (Schelling’s model), or change their opinions in a way that depends on the opinions of the interacting partners. In a Volter model, for example, everyone takes an opinion of a random neighbor, in the majority model one adopts the opinion of the majority.

The primary tool for analyzing dynamics of complexity models of social processes are computer simulations. Simulation models described by Troitzsch (see ► [Social Processes, Simulation Models of](#)) allow one to discover dynamical properties of models that, due to their complexity and nonlinearity of interactions, may be unsolvable using

analytical tools. Most computer simulations utilize agent-based models. In such models individuals are represented as agents who interact with other agents following the rules specified by the model. Models taken from physics assume extremely simple agents who are often represented by a single binary variable such as an opinion in a position in a social space. In such models the rules assumed on the level of a single agent are usually trivial; however interaction between agents leads to the emergence of complex spatial and temporal patterns on the social level. The rules specifying the behavior of a single agent may sometimes become complex. As Conte points out, an approach originating from computer science also assumes much richer representation of individuals (see ► [Rational, Goal-Oriented Agents](#)).

Complexity models of social processes are very different from traditional mathematical models. Traditional mathematical and physical models describe the temporal evolution of the model in terms of equations involving the global characteristics of systems. Yaari, Stauffer and Solomon discuss intermittency and localization, central properties of the dynamics of complexity-based models (see ► [Intermittency and Localization](#)). Most classical equations describing social and demographic processes, such as the logistic equation of growth, acquire unexpected and fascinating properties when they define local rather than global dynamics. These models produce a variety of behaviors that are observed in many real-life systems, for example in economics, sociology, biology, and ecology. According to Kelso and Oullier coordination is one of the pivotal aspects of emergent social processes (see ► [Social Coordination, from the Perspective of Coordination Dynamics](#)). Individuals not only act alone but they coordinate in the achievement of their goals. Understanding how this coordination occurs and what forms it may take allows us to gain insight into how individuals are combined into dyads and higher level social structures.

White suggests that human behavior may be analyzed at different levels (see ► [Human Behavior, Dynamics of](#)). Other people provide the critical context for actions of individuals. Moreover, the actions of individuals change the structure of society, for example by creating connections between cooperating individuals. Cohesion provides an important characteristic for group processes. Networks organize individuals into social groups and societies where the cohesion is an important characteristic of the resultant aggregates. Whereas the individual level of description may involve rules, the dynamics of aggregates may be described by equations.

Processes occurring in organizations, social groups and societies to high degree depend on structure of in-

teractions among individuals. Reed–Tsochas explains that the structure of human interactions and human relations may be described as a complex network where individuals correspond to nodes and links to social context or social relations. Most networks existing in nature as well as in societies have several common properties such as scale free distribution: there is a small number of nodes with a very high number of connections (degree) and a large number of nodes with a small number of connections. Networks tend to have a small world structure. In societies it is possible to go from one node to most of the others by traversing no more than six links. Autoaffiliate character means that similar nodes tend to be connected. There is a high chance of being connected for nodes that are similar than for nodes that are dissimilar. The network formalism offers the very tool to analyze structure of complex relations among individuals, groups and organizations in a society. These network properties have important consequences for the dynamics of processes occurring in social groups and organizations.

Complexity models are applied to different levels and within levels to different domains of social processes. Complexity models based in the neurophysiology of the human brain have provided the first precise model of the functioning of consciousness. As described by Edelman and Seth, momentary progressive coordination of activity of selective neuronal groups underlies the phenomena of consciousness (see ► [Consciousness and Complexity](#)). This precise complexity-based model suggests that consciousness is not a property of a specific brain region but an emergent, collective phenomenon. This model explains a variety of experimental results from psychology. Although affective responses are produced by specific brain regions, they may be linked to the dynamical properties of patterns of activity of the cognitive system. In particular, emotions, as described by Winkielman and Huber in ► [Dynamics and Evaluation: The Warm Glow of Processing Fluency](#), depend on the dynamics of perception. Positive affect is elicited by the fluent processing of stimuli resulting from high coherence in the processing mechanisms, whereas negative emotions result from low fluency.

As Port points out, the dynamics of language (see ► [Dynamics of Language](#)) can be discussed either on the level of an individual or a society. At the individual level language can be understood as the skills of individuals processing speech in real time, at the social level as a common pattern of the speech of speakers in a community.

Rational rules governing the interaction of individual agents can produce moral dynamics in which the norms of a society evolve among individuals who are

linked by social interdependence. In his article, Hegselman explains how rational choices may underlie the emergence of norms and the evolution of solidarity and trust (see ► [Moral Dynamics](#)).

Applications of complexity to social psychology involve dynamical models of various individual and group-level processes described by Vallacher (see ► [Social Psychology, Applications of Complexity to](#)). The attractor is the key concept for finding stability in the constant dynamics of all psychological processes. Processes oscillate around stable points which function as attractors for their dynamics. Dynamical minimalism describes the core of the approach of complexity – how to build very simple models of very complex psychological and social processes. At the abstract level many aspects of individual interactions may be described in terms of social influence.

The intimate link between complexity and the dynamics of psychological processes is most clearly seen in development. The concept of development may be captured in terms of the increasing complexity of the human system. Van Geert explains one of the main contributions of the complexity approach to development, namely the realization that mental and behavioral phenomena result, not from individual factors, but from the joint impact of many factors acting at different time scales (see ► [Development, Complex Dynamic Systems of](#)). The question of time scales goes beyond individuals to generations when one considers similarities between development and evolution. Evolutionary principles as applied to human systems, described by Bjorklund and Gardiner, define the subject matter of memetics (see ► [Development, Evolution, and the Emergence of Novel Behavior](#)). Human opinions, norms and beliefs decide on the fitness of a given individual to the social environment. Successful individuals, because of the higher fitness, have a higher chance of transmitting this property to others. Through such processes human modeling occurs. From the evolutionary point of view, the characteristics of individuals (such as opinions, norms, values) may be treated as genes and the process of transmission of these properties can be captured in terms of reproduction, which depends on fitness. As explained by Heylighen and Chielens, evolution of culture can simply be modeled by genetic algorithms resembling evolutionary processes and the approach taken by memetics (see ► [Evolution of Culture, Memetics](#)).

The complexity approach provides principles to describe the evolution of cities in time, and explain patterns of urban dynamics (see ► [Cities as Complex Systems: Scaling, Interaction, Networks, Dynamics and Urban Morphologies](#) by Batty). Comprehensive models of cities show the interrelation between various processes in

the city. Bazzani, Giorgini, and Rambaldi present traffic and crowd dynamics that provided focus for studies concerning the dynamics of urban processes (see ► [Traffic and Crowd Dynamics: The Physics of the City](#)).

Interactions within the organization of multiple people, factors, and ideas may be understood in terms of knowledge economy. As described by Marion in ► [Social Organizations with Complexity Theory: A Dramatically Different Lens for the Knowledge Economy](#), creativity, adaptability, and learning may be described in terms of knowledge transfers and transformations within an organization.

The spread of product adoption may be described by the diffusion model. Barriers in communication between early adopters and followers may result in a dual market phenomenon, which may be analyzed using the tools of complexity modeling. Goldenberg presents models in which cellular automata describe the geographical spread of a product (see ► [Marketing: Complexity Modeling, Theory and Applications in](#)). The interdisciplinary approach allows one to develop a universal framework for modeling market penetration dynamics.

The most precise models of complexity are being developed to model economic processes. Gallegati defines agent-based models as the core of complexity modeling of economic processes (see ► [Agent Based Models in Economics and Complexity](#)). These models are usually characterized by heterogeneity, explicit space, and local interactions. Agents' behaviors are usually governed by the principles of bounded rationality in that they have limited information and therefore limited processing capacity.

Minority game, as described by Zheng (see ► [Minority Games](#)), provides a formal model of how the financial market functions. In this model, individuals in a group must make a binary choice. Those who are in the minority and choose the less popular option win. The interesting feature of the game is that any winning strategy becomes a losing strategy as soon as it is adopted by the majority of the group. It forces constant evolution of the rules governing the market, where each rule is invalidated as soon as the majority of the agents discovers it.

Agent-based modeling challenges many assumptions of neoclassical economics. Moss suggests in his critical analysis (see ► [Agent Based Modeling and Neoclassical Economics: A Critical Perspective](#)) that the neoclassical economy describes an idealized society and does not apply to any real social system in which economic decisions are made by heterogeneous interacting agents.

Andersen describes models of investment decision making in finance that are constructed by assuming different decision rules of agents trying to maximize their profit

(see ► [Investment Decision Making in Finance, Models of](#)). Researchers try to understand patterns of change in financial markets in relation to simple rules of agents trying to maximize their outcomes. These models allow one to study typical patterns of market dynamics as the link between decision rules of agents and the resultant dynamics of the market indexes.

The dynamics of social and political processes are highly nonlinear and often abrupt. The state of societies is shaped to a substantial degree by rare extreme events rather than by the mass of average processes. Keilis-Borok, Soloviev, and Lichtman claim that predictability of extreme events in socio-economics and politics (see ► [Extreme Events in Socio-economic and Political Complex Systems, Predictability of](#)) is thus an important consideration for the prediction of any social, political, and economic process. New developments concerning the predictability of extreme events improve the possibility of predicting the course of dynamics of various social processes such as unemployment, recession, surges in homicides, etc.

In contrast to traditional mathematical and physical models, the approach of complexity allows us to combine deep understanding of psychological and social phenomena with the precision of the formal sciences. It provides a platform for the integration of natural and social sciences. It describes phenomena occurring at higher levels of reality as emergent effects of the interaction of lower level elements. It allows, therefore, for the integration of theories describing various levels of psychological and social reality, from the level of consciousness to high social levels such as economies, cities, and societies. In addition to its heuristic value this approach has a strong integrative potential also between different disciplines of behavioral and social sciences.

Everyone agrees that psychological and social processes display the greatest complexity of all the phenomena research can encounter. The surprising discovery of the complexity approach is that extremely complex processes may be produced by very simple rules and interactions. From this perspective simplicity is a coin flip of complexity rather than its opposition. The approach of complexity can therefore provide for simple, precise explanations of extremely complex psychological and social processes. The title of this section could be recast as Finding simplicity in complex psychological and social processes.

Acknowledgment

Preparation of this article was supported by a CO3 grant from the European NEST PATHFINDER initiative.

Artificial Chemistry

PETER DITTRICH

Department of Mathematics and Computer Science,
Friedrich Schiller University Jena, Jena, Germany

Article Outline

Glossary

Definition of the Subject

Introduction

Basic Building Blocks of an Artificial Chemistry

Structure-to-Function Mapping

Space

Theory

Evolution

Information Processing

Future Directions

Bibliography

Glossary

Molecular species A molecular species is an abstract class denoting an ensemble of identical molecules. Equivalently the terms “species”, “compound”, or just “molecule” are used; in some specific context also the terms “substrate” or “metabolite”.

Molecule A molecule is a concrete instance of a molecular species. Molecules are those entities of an artificial chemistry that react. Note that sometimes the term “molecule” is used equivalently to molecular species.

Reaction network A set of molecular species together with a set of reaction rules. Formally, a reaction network is equivalent to a Petri network. A reaction network describes the stoichiometric structure of a reaction system.

Order of a reaction The order of a reaction is the sum of the exponents of concentrations in the kinetic rate law (if given). Note that an order can be fractional. If only the stoichiometric coefficients of the reaction rule are given (Eq. (1)), the order is the sum of the coefficients of the left-hand side species. When assuming mass-action kinetics, both definitions are equivalent.

Autocatalytic set A (self-maintaining) set where each molecule is produced catalytically by molecules from that set. Note that an autocatalytic may produce molecules not present in that set.

Closure A set of molecules A is closed, if no combination of molecules from A can react to form a molecule outside A . Note that the term “closure” has been also

used to denote the catalytical closure of an autocatalytic set.

Self-maintaining A set of molecules is called self-maintaining, if it is able to maintain all its constituents. In a purely autocatalytic system under flow condition, this means that every molecule can be catalytically produced by at least one reaction among molecule from the set.

(Chemical) Organization A closed and self-maintaining set of molecules.

Multiset A multiset is like a set but where elements can appear more than once; that is, each element has a multiplicity, e. g., $\{a, a, b, c, c, c\}$.

Definition of the Subject

Artificial chemistries are chemical-like systems or abstract models of chemical processes. They are studied in order to illuminate and understand fundamental principles of chemical systems as well as to exploit the chemical metaphor as a design principle for information processing systems in fields like chemical computing or nonlinear optimization.

An artificial chemistry (AC) is usually a formal (and, more seldom, a physical) system that consists of objects called molecules, which interact according to rules called reactions. Compared to conventional chemical models, artificial chemistries are more abstract in the sense that there is usually not a one-to-one mapping between the molecules (and reactions) of the artificial chemistry to real molecules (and reactions). An artificial chemistry aims at capturing the logic of chemistry rather than trying to explain a particular chemical system. More formally, an artificial chemistry can be defined by a set of molecular species \mathcal{M} , a set of reaction rules \mathcal{R} , and specifications of the dynamics, e. g., kinetic laws, update algorithm, and geometry of the reaction vessel.

The scientific motivation for AC research is to build abstract models in order to understand chemical-like systems in all kind of domains ranging from physics, chemistry, biology, computer science, and sociology. Abstract chemical models to study fundamental principles, such as spatial pattern formation and self-replication, can be traced back to the beginning of modern computer science in the 40s [91,97]. Since then a growing number of approaches have tackled questions concerning the origin of complex forms, the origin of life itself [19,81], or cellular diversity [48]. In the same way a variety of approaches for constructing ACs have appeared, ranging from simple ordinary differential equations [20] to complex individual-based algebraic transformation systems [26,36,51].

The engineering motivation for AC research aims at employing the chemical metaphor as a programming and computing paradigm. Approaches can be distinguished according to whether chemistry serves as a paradigm to program or construct conventional *in-silico* information processing systems [4]. Or whether real molecules are used for information processing like in molecular computing [13] or DNA computing [2] ► [Cellular Computing](#), ► [DNA Computing](#).

Introduction

The term artificial chemistry appeared around 1990 in the field of artificial life [72]. However, models that fell under the definition of artificial chemistry appeared also decades before, such as von Neumann's universal self-replicating automata [97]. In the 50s, a famous abstract chemical system was introduced by Turing [91] in order to show how spatial diffusion can destabilize a chemical system leading to spatial pattern formation. Turing's artificial chemistries consist of only a handful of chemical species interacting according to a couple of reaction rules, each carefully designed. The dynamics is simulated by a partial differential equation.

Turing's model possesses the structure of a conventional reaction kinetics chemical model (Subsect. "The Chemical Differential Equation"). However it does not aim at modeling a particular reaction process, but aims at exploring how, in principle, spatial patterns can be generated by simple chemical processes. Turing's artificial chemistries allow one to study whether a particular mechanism (e. g., destabilization through diffusion) can explain a particular phenomena (e. g., symmetry breaking and pattern formation).

The example illustrates that studying artificial chemistries is more a synthetic approach. That is, understanding should be gained through the synthesis of an artificial system and its observation under various conditions [45,55]. The underlying assumption is that understanding a phenomena and how this phenomena can be (re-)created (without copying it) is closely related [96].

Today's artificial chemistries are much more complex than Turing's model. They consist of a huge, sometimes infinite, amount of different molecular species and their reaction rules are defined not manually one by one, but implicitly (Sect. "Structure-to-Function Mapping") and can evolve (Sect. "Evolution"). Questions that are tackled are the structure and organization of large chemical networks, the origin of life, prebiotic chemical evolution, and the emergence of chemical information and its processing.

Introductory Example

It is relatively easy to create a complex, infinitely sized reaction network. Assume that the set of possible molecules \mathcal{M} are strings over an alphabet. Next, we have to define a reaction function describing what happens when two molecules $a_1, a_2 \in \mathcal{M}$ collide. A general approach is to map a_1 to a machine, operator, or function $f = \text{fold}(a_1)$ with $f : \mathcal{M} \rightarrow \mathcal{M} \cup \{\text{elastic}\}$. The mapping fold can be defined in various ways, such as by interpreting a_1 as a computer program, a Turing machine, or a lambda expression (Sect. "Structure-to-Function Mapping"). Next, we apply the machine f derived from a_1 to the second molecule. If $f(a_2) = \text{elastic}$ the molecules collided elastically and nothing happens. This could be the case if the machine f does not halt within a predefined amount of time. Otherwise the molecules react and a new molecule $a_3 = f(a_2)$ is produced. How this new molecule changes the reaction vessel depends on the algorithm simulating the dynamics of the vessel. Algorithm 1 is a simple, widely applied algorithm that simulates second-order catalytic reactions under flow conditions: Because the population size is constant and thus limited, there is competition and only those molecules that are somehow created will sustain.

First, the algorithm chooses two molecules randomly, which simulates a **collision**. Note that a realistic measure of time takes the amount of collision and not the amount of reaction events into account. Then, we determine stochastically whether the selected molecules react. If the molecules react, a randomly selected molecule from the population is replaced by the new product, which assures that the population size stays constant. The replaced molecules constitute the outflow. The inflow is implicitly assumed and is generated by the catalytic production of molecules. From a chemical point of view, the algorithm assumes that a product is built from an energy-rich substrate, which does not appear in the algorithm and which is assumed to be available at constant concentration.

The following section (Sect. "Basic Building Blocks of an Artificial Chemistry") describes the basic building blocks of an artificial chemistry in more detail. In Sect. "Structure-to-Function Mapping" we will explore various techniques to define reaction rules. The role of space is briefly discussed in Sect. "Space". Then the important theoretical concepts of an autocatalytic set and a chemical organization are explained (Sect. "Theory"). Sect. "Evolution" shows how artificial chemistries can be used to study (chemical) evolution. This article does not discuss information processing in detail, because there are a series of other specialized articles on that topic, which are summarized in Sect. "Information Processing".

```

INPUT: P : population (array of k molecules)

randomPosition() := randomInt(0, k-1);
reaction(r1, r2) := (fold(r1))(r2);

while not terminate do
  reactant1 := P[randomPosition()];
  reactant2 := P[randomPosition()];
  product := reaction(reactant1, reactant2);
  if product == not elastic
    P[randomPosition()] := product;
  fi
  t := t + 1/k ;; increment simulated time
od

```

Artificial Chemistry, Algorithm 1

Basic Building Blocks of an Artificial Chemistry

Molecules

The first step in defining an AC requires that we define the set of all molecular species that can appear in the AC. The easiest way to specify this set is to enumerate explicitly all molecules as symbols. For example: $\mathcal{M} = \{H_2, O_2, H_2O, H_2O_2\}$ or, equivalently, $\mathcal{M} = \{a, b, c, d\}$. A symbol is just a name without any structure. Most conventional (bio-)chemical reaction system models are defined this way. Also many artificial chemistries where networks are designed “by hand” [20], are created randomly [85], or are evolved by changing the reaction rules [41,42,43] use only symbolic molecules without any structure.

Alternatively, molecules can possess a structure. In that case, the set of molecules is implicitly defined. For example: $\mathcal{M} = \{\text{all polymers that can be made from the two monomers } a \text{ and } b\}$. In this case, the set of all possible molecules can even become infinite, or at least quite large. A vast variety of such rule-based molecule definitions can be found in different approaches. For example, structured molecules may be abstract character sequences [3,23,48,64], sequences of instructions [1,39], lambda-expressions [26], combinators [83], binary strings [5,16,90], numbers [8], machine code [72], graphs [78], swarms [79], or proofs [28].

We can call a molecule’s representation its **structure**, in contrast to its **function**, which is given by the reaction rules \mathcal{R} . The description of the valid molecules and their structure is usually the first step when an AC is defined. This is analogous to a part of chemistry that describes what kind of atom configurations form stable molecules and how these molecules appear.

Reaction Rules

The set of reaction rules \mathcal{R} describes how molecules from $\mathcal{M} = \{a_1, a_2, \dots\}$ interact. A rule $\rho \in \mathcal{R}$ can be written according to the chemical notation of reaction rules in the form

$$l_{a_1,\rho}a_1 + l_{a_2,\rho}a_2 + \dots \rightarrow r_{a_1,\rho}a_1 + r_{a_2,\rho}a_2 + \dots \quad (1)$$

The stoichiometric coefficients $l_{a,\rho}$ and $r_{a,\rho}$ describe the amount of molecular species $a \in \mathcal{M}$ in reaction $\rho \in \mathcal{R}$ on the left-hand and right-hand side, respectively. Together, the stoichiometric coefficients define the **stoichiometric matrix**

$$\mathbf{S} = (s_{a,\rho}) = (r_{a,\rho} - l_{a,\rho}) \quad (2)$$

An entry $s_{a,\rho}$ of the stoichiometric matrix denotes the net amount of molecules of type a produced in reaction ρ .

A reaction rule determines a multiset of molecules (the left-hand side) that can react and subsequently be replaced by the molecules on the right-hand side. Note that the sign “+” is not an operator here, but should only separate the components on either side. The set of all molecules \mathcal{M} and a set of reaction rules \mathcal{R} define the **reaction network** $\langle \mathcal{M}, \mathcal{R} \rangle$ of the AC. The reaction network is equivalent to a Petri net [71]. It can be represented by two matrices, $(l_{a,\rho})$ and $(r_{a,\rho})$, made up of the stoichiometric coefficients. Equivalently we can represent the reaction network as a hyper-graph or a directed bipartite graph with two node types denoting reaction rules and molecular species, respectively, and edge labels for the stoichiometry.

A rule is applicable only if certain conditions are fulfilled. The major condition is that all of the left-hand side components must be available. This condition can

be broadened easily to include other parameters such as neighborhood, rate constants, probability for a reaction, influence of modifier species, or energy consumption. In such a case, a reaction rule would also contain additional information or further parameters. Whether or not these additional predicates are taken into consideration depends on the objectives of the artificial chemistry. If it is meant to simulate real chemistry as accurate as possible, it is necessary to integrate these parameters into the simulation system. If the goal is to build an abstract model, many of these parameters can be omitted.

Like for the set of molecules we can define the set of reaction rules explicitly by enumerating all rules symbolically [20], or we can define it implicitly by referring to the structure of the molecules. Implicit definitions of reaction rules use, for example, string matching/string concatenation [3,60,64], lambda-calculus [26,27], Turing machines [40,90], finite state machines [98] or machine code language [1,16,78,87], proof theory [27], matrix multiplication [5], swarm dynamics [79], or simple arithmetic operations [8]. Note that in some cases the reactions can emerge as the result of the interaction of many atomic particles, whose dynamics is specified by force fields or rules [79]. Section “Structure-to-Function Mapping” will discuss implicitly defined reactions in more details.

Dynamics

The third element of an artificial chemistry is its specification of how the reaction rules give rise to a dynamic process of molecules reacting in some kind of reaction vessel. It is assumed that the reaction vessel contains multiple copies of molecules, which can be seen as instances of the molecular species \mathcal{M} .

This section summarizes how the dynamics of a reaction vessel (which usually contains a huge number of molecules) can be modeled and simulated. The approaches can be characterized roughly by whether each molecule is treated explicitly or whether all molecules of one type are represented by a number denoting their frequency or concentration.

The Chemical Differential Equation A reaction network $\langle \mathcal{M}, \mathcal{R} \rangle$ specifies the structure of a reaction system, but does not contain any notion of time. A common way to specify the dynamics of the reaction system is by using a system of ordinary differential equations of the following form:

$$\dot{\mathbf{x}}(t) = \mathbf{S}\mathbf{v}(\mathbf{x}(t)), \quad (3)$$

where $\mathbf{x} = (x_1, \dots, x_m)^T \in \mathbb{R}^m$ is a concentration vector

depending on time t , \mathbf{S} the stoichiometric matrix derived from \mathcal{R} (Eq. (2)), and $\mathbf{v} = (v_1, \dots, v_r)^T \in \mathbb{R}^r$ a flux vector depending on the current concentration vector. A flux $v_\rho \geq 0$ describes the velocity or turnover rate of reaction $\rho \in \mathcal{R}$. The actual value of v_ρ depends usually on the concentration of the species participating in the reaction ρ (i. e., $\text{LHS}(\rho) \equiv \{a \in \mathcal{M}: l_{a,\rho} > 0\}$) and sometimes on additional modifier species, whose concentration is not influenced by that reaction. There are two important assumptions that are due to the nature of reaction systems. These assumptions relate the kinetic function \mathbf{v} to the reaction rules \mathcal{R} :

Assumption 1 A reaction ρ can only take place if all species of its left-hand side $\text{LHS}(\rho)$ are present. This implies that for all molecules $a \in \mathcal{M}$ and reactions $\rho \in \mathcal{R}$ with $a \in \text{LHS}(\rho)$, if $x_a = 0$ then $v_\rho = 0$. The flux v_ρ must be zero, if the concentration x_a of a molecule appearing on the left-hand side of this reaction is zero. This assumption meets the obvious fact that a molecule has to be present to react.

Assumption 2 If all species $\text{LHS}(\rho)$ of a reaction $\rho \in \mathcal{R}$ are present in the reactor (e. g. for all $a \in \text{LHS}(\rho)$, $x_a > 0$) the flux of that reaction is positive, (i. e., $v_\rho > 0$). In other words, the flux v_ρ must be positive, if all molecules required for that reaction are present, even in small quantities. Note that this assumption implies that a modifier species necessary for that reaction must appear as a species on the left- (and right-) hand side of ρ .

In short, taking Assumption 1 and 2 together, we demand for a chemical differential equation:

$$v_\rho > 0 \Leftrightarrow \forall a \in \text{LHS}(\rho), x_a > 0. \quad (4)$$

Note that Assumption 1 is absolutely necessary. Although Assumption 2 is very reasonable, there might be situations where it is not true. Assume, for example, a reaction of the form $a + a \rightarrow b$. If there is only one single molecule a left in the reaction vessel, the reaction cannot take place. Note however, if we want to model this effect, an ODE is the wrong approach and we should choose a method like those described in the following sections.

There are a large number of kinetic laws fulfilling these assumptions (Eq. (4)), including all laws that are usually applied in practice. The most fundamental of such kinetic laws is **mass-action kinetics**, which is just the product of the concentrations of the interacting molecules (cf. [32]):

$$v_\rho(\mathbf{x}) = \prod_{a \in \mathcal{M}} x_a^{l_{a,\rho}}. \quad (5)$$

The sum of the exponents in the kinetic law $\sum_{a \in \mathcal{M}} l_{a,\rho}$ is called the **order** of the reaction. Most more compli-

cated laws like Michaelis–Menten kinetics are derived from mass-action kinetics. In this sense, mass-action kinetics is the most general approach, allowing one to emulate all more complicated laws derived from it, but at the expense of a larger amount of molecular species.

Explicitly Simulated Collisions The chemical differential equation is a time and state continuous approximation of a discrete system where molecules collide stochastically and react. In the introduction we saw how this can be simulated by a rather simple algorithm, which is widely used in AC research [5,16,26]. The algorithm presented in the introduction simulates a second-order catalytic flow system with constant population size. It is relatively easy to extend this algorithm to include reaction rates and arbitrary orders.

First, the algorithm chooses a subset from P , which simulates a collision among molecules. Note that the resulting subset could be empty, which can be used to simulate an inflow (i.e., reactions of the form $\rightarrow A$, where the left-hand side is empty). By defining the probability of obtaining a subset of size n , we can control the rate of reactions of order n . If the molecules react, the reactants are removed from the population and the products are added. Note that this algorithm can be interpreted as a stochastic rewriting system operating on a multiset of molecules [70,88].

For large population size k , the dynamics created by the algorithm tends to the continuous dynamics of the chemical differential equation assuming mass-action kinetics. For the special case of second-order catalytic flow, as described by the algorithm in the introduction, the dynamics can be described by the catalytic network equation [85]

$$\begin{aligned}\dot{x}_k &= \sum_{i,j \in \mathcal{M}} \alpha_{i,j}^k x_i x_j - x_k \Phi(\mathbf{x}), \quad \text{with} \\ \Phi(\mathbf{x}) &= \sum_{i,j,k \in \mathcal{M}} \alpha_{i,j}^k x_i x_j,\end{aligned}\quad (6)$$

which is a generalization of the replicator equation [35]. If the reaction function is deterministic, we can set the kinetic constants to

$$\alpha_{i,j}^k = \begin{cases} 1 & \text{reaction}(i, j) = k, \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

This dynamic is of particular interest, because competition is generated due to a limited population size. Note that the limited population size is equivalent to an unlimited population size with a limited substrate, e.g., [3].

Discrete-Event Simulation If the copy number of a particular molecular species becomes high or if the reaction rates differ strongly, it is more efficient to use discrete-event simulation techniques. The most famous technique is Gillespie's algorithm [32]. Roughly, in each step, the algorithm generates two random numbers depending on the current population state to determine the next reaction to occur as well as the time interval Δt when it will occur. Then the simulated time is advanced $t := t + \Delta t$, and the molecule count of the population updated according to the reaction that occurred.

The Gillespie algorithm generates a statistically correct trajectory of the chemical master equation. The chemical master equation is the most general way to formulate the stochastic dynamics of a chemical system. The chemical master equation is a first-order differential equation describing the time-evolution of the probability of a reaction system to occupy each one of its possible discrete set of states. In a well-stirred (non spatial) AC a state is equivalent to a multiset of molecules.

Structure-to-Function Mapping

A fundamental “logic” of real chemical systems is that molecules possess a structure and that this structure determines how the molecule interacts with other molecules. Thus, there is a mapping from the structure of a molecule to its function, which determines the reaction rules.

In real chemistry the mapping from structure to function is given by natural or physical laws. In artificial chemistries the structure-to-function mapping is usually given by some kind of algorithm. In most cases, an artificial chemistry using a structure-to-function mapping does not aim at modeling real molecular dynamics in detail. Rather the aim is to capture the fact that there is a structure, a function, and a relation between them.

Having an AC with a structure-to-function mapping at hand, we can study strongly constructive dynamical systems [27,28]. These are systems where, through the interaction of its components, novel molecular species appear.

Example: Lambda-Calculus (AlChemY)

This section demonstrates a structure-to-function that employs a concept from computer science: the λ -calculus. The λ -calculus has been used by Fontana [26] and Fontana and Buss [27] to define a constructive artificial chemistry.

In the so-called *AlChemY*, a molecule is a normalized λ -expression. A λ -expression is a word over an alphabet $A = \{\lambda, ., (,)\} \cup V$ where $V = \{x_1, x_2, \dots\}$ is an infinite set of available variable names. The set of λ expres-

```

while not terminate() do
  reactands := choseASubsetFrom(P);
  if randomReal(0, 1) < reactionProbability(reactands);
    products = reaction(reactands);
    P := remove(P, reactands);
    P := insert(P, products);
  fi
  t := t + 1/ sizeOf(P)  ;; increment simulated time
od

```

Artificial Chemistry, Algorithm 2

sions Γ is defined for $x \in V, s_1 \in \Gamma, s_2 \in \Gamma$ by

$x \in \Gamma$ variable name ,
 $\lambda x.s_2 \in \Gamma$ abstraction ,
 $(s_2)s_1 \in \Gamma$ application .

An abstraction $\lambda x.s_2$ can be interpreted as a function definition, where x is the parameter in the “body” s_2 . The expression $(s_2)s_1$ can be interpreted as the application of s_2 on s_1 , which is formalized by the rewriting rule

$$(\lambda x.s_2)s_1 = s_2[x \leftarrow s_1] , \quad (8)$$

where $s_2[x \leftarrow s_1]$ denotes the term which is generated by replacing every unbounded occurrence of x in s_2 by s_1 . A variable x is bounded if it appears in a form like $\dots(\lambda x.\dots x\dots)\dots$. It is also not allowed to apply the rewriting rule if a variable becomes bounded. Example: Let $s_1 = \lambda x_1.(x_1)\lambda x_2.x_2$ and $s_2 = \lambda x_3.x_3$ then we can derive:

$$\begin{aligned}
 (s_1)s_2 &\Rightarrow (\lambda x_1.(x_1)\lambda x_2.x_2)\lambda x_3.x_3 \\
 &\Rightarrow (\lambda x_3.x_3)\lambda x_2.x_2 \Rightarrow \lambda x_2.x_2 . \quad (9)
 \end{aligned}$$

The simplest way to define a set of second-order catalytic reactions \mathcal{R} by applying a molecule s_1 to another molecule s_2 :

$$s_1 + s_2 \rightarrow s_1 + s_2 + \text{normalForm}((s_1)s_2) . \quad (10)$$

The procedure *normalForm* reduces its argument term to normal form, which is in practice bounded by a maximum of available time and memory; if these resources are exceeded before termination, the collision is considered to be elastic. It should be noted that the λ -calculus allows an elegant generalization of the collision rule by defining it by λ -expression $\Phi \in \Gamma: s_1 + s_2 \rightarrow s_1 + s_2 + \text{normalForm}(((\Phi)s_1)s_2)$.

In order to simulate the artificial chemistry, Fontana and Buss [27] used an algorithm like the algorithm described in Subsect. “Explicitly Simulated Collisions”,

which simulates second-order catalytic mass-action kinetics in a well-stirred population under flow condition.

*AlChem*y possesses a couple of important general properties: Molecules come in two forms: as passive data possessing a structure and as active machines operating on those structures. This generates a “strange loop” like in typogenetics [36,94], which allows molecules to refer to themselves and to operate on themselves. Structures operate on structures, and by doing so change the way structures operate on structures, and so on; creating a “strange”, self-referential dynamic loop. Obviously, there are always some laws that cannot be changed, which are here the rules of the lambda-calculus, defining the structure-to-function mapping. Thus, we can interpret these fixed and predefined laws of an artificial chemistry as the natural or physical laws.

Arithmetic and Logic Operations

One of the most easy ways to define reaction rules implicitly is to apply arithmetic operations taken from mathematics. Even the simplest rules can generate interesting behaviors. Assume, for example, that the molecules are natural numbers: $\mathcal{M} = \{0, 1, 2, \dots, n-1\}$ and the reaction rules are defined by division: $\text{reaction}(a_1, a_2) = a_1/a_2$ if a_1 is a multiple of a_2 , otherwise the molecules do not react. For the dynamics we assume a finite, discrete population and an algorithm like the one described in Subsect. “Explicitly Simulated Collisions”. With increasing population size the resulting reaction system displays a phase transition, at which the population is able to produce prime numbers with high probability (see [8] for details). In a typical simulation, the initially high diversity of a random population is reduced leaving a set of non-reacting prime numbers.

A quite different behavior can be obtained by simply replacing the division by addition: $\text{reaction}(a_1, a_2) = a_1 + a_2 \bmod n$ with $n = |\mathcal{M}|$ the number of molecular species. Initializing the well-stirred tank reactor only with

the molecular species 1, the diversity rapidly increases towards its maximum and the reactor behaves apparently totally chaotic after a very short transient phase (for a sufficiently large population size). However, there are still regularities, which depend on the prime factors of n (cf. [15]).

Matrix-Multiplication Chemistry More complex reaction operators can be defined that operate on vectors or strings instead of scalar numbers as molecules. The matrix multiplication chemistry introduced by Banzhaf [5,6,7] uses binary strings as molecules. The reaction between two binary strings is performed by folding one of them into a matrix which then operates on the other string by multiplication.

Example of a reaction for 4-bit molecules Assume a reaction $s_1 + s_2 \Rightarrow s_3$. The general approach is:

1. Fold s_1 to matrix M . Example: $s_1 = (s_1^1, s_1^2, s_1^3, s_1^4)$

$$M = \begin{pmatrix} s_1^1 & s_1^2 \\ s_1^3 & s_1^4 \end{pmatrix}. \quad (11)$$

2. Multiply M with subsequences of s_2 . Example: Let $s_2 = (s_2^1, s_2^2, s_2^3, s_2^4)$ be divided into two subsequences $s_2^{12} = (s_2^1, s_2^2)$ and $s_2^{34} = (s_2^3, s_2^4)$. Then we can multiply M with the subsequences:

$$s_3^{12} = M \odot s_2^{12}, \quad s_3^{34} = M \odot s_2^{34}. \quad (12)$$

3. Compose s_3 by concatenating the products. Example: $s_3 = s_3^{12} \oplus s_3^{34}$.

There are various ways of defining the vector matrix product \odot . It was mainly used with the following **threshold multiplication**. Given a bit vector $x = (x_1, \dots, x_n)$ and a bit matrix $M = (M_{ij})$ then the term $y = M \odot x$ is defined by:

$$y_j = \begin{cases} 0 & \text{if } \sum_{i=1}^n x_i M_{i,j} \leq \Phi. \\ 1 & \text{otherwise.} \end{cases} \quad (13)$$

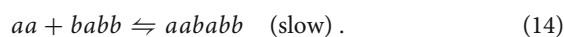
The threshold multiplication is similar to the common matrix-vector product, except that the resulting vector is mapped to a binary vector by using the threshold Φ .

Simulating the dynamics by an ODE or explicit molecular collisions (Subsect. "Explicitly Simulated Collisions"), we can observe that such a system would develop into a steady state where some string species support each other in production and thus become a stable autocatalytic cycle, whereas others would disappear due to the competition in the reactor.

The system has a couple of interesting properties: Despite the relatively simple definition of the basic reaction mechanism, the resulting complexity of the reaction system and its dynamics is surprisingly high. Like in typogenetics and Fontana's lambda-chemistry, molecules appear in two forms; as passive data (binary strings) and as active operators (binary matrix). The folding of a binary string to a matrix is the central operation of the structure-to-function mapping. The matrix is multiplied with substrings of the operand and thus some kind of locality is preserved, which mimics local operation of macromolecules (e.g. ribosomes or restriction enzymes) on polymers (e.g. RNA or DNA). Locality is also conserved in the folding, because bits that are close in the string are also close in the matrix.

Autocatalytic Polymer Chemistries

In order to study the emergence and evolution of autocatalytic sets [19,46,75] Bagley, Farmer, Fontana, Kauffman and others [3,23,47,48,59] have used artificial chemistries where the molecules are character sequences (e.g., $\mathcal{M} = \{a, b, aa, ab, ba, bb, aaa, aab, \dots\}$) and the reactions are concatenation and cleavage, for example:



Additionally, each molecule can act as a catalyst enhancing the rate of a concatenation reaction.

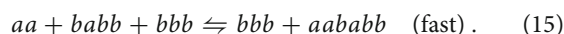
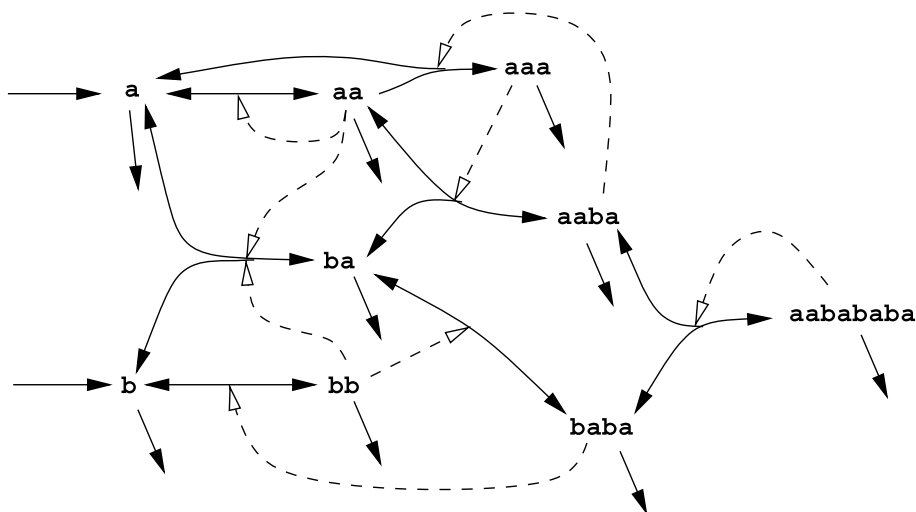


Figure 1 shows an example of an autocatalytic network that appeared. There are two time scales. Reactions that are not catalyzed are assumed to be very slow and not depicted, whereas catalyzed reactions, inflow, and decay are fast.

Typical experiments simulate a well-stirred flow reactor using meta-dynamical ODE framework [3]; that is, the ODE model is dynamically changed when new molecular species appear or present species vanish. Note that the structure of the molecules does not fully determine the reaction rules. Which molecule catalyzes which reaction (the dotted arrows in Fig. 1) is assigned explicitly randomly. An interesting aspect is that the catalytic or autocatalytic polymer sets (or reaction networks) evolve without having a genome [47,48]. The simulation studies show how small, spontaneous fluctuations can be amplified by an autocatalytic network, possibly leading to a modification of the entire network [3]. Recent studies by Fernando and Rowe [24] include further aspects like energy conservation and compartmentalization.



Artificial Chemistry, Figure 1

Example of an autocatalytic polymer network. The *dotted lines* represent catalytic links, e.g., $aa + ba + aaa \rightarrow aaba + aaa$. All molecules are subject to a dilution flow. In this example, all polymerization reactions are reversible and there is a continuous inflow of the monomers a and b . Note that the network contains further auto-catalytic networks, for example, $\{a, b, aa, ba\}$ and $\{a, aa\}$, of which some are closed and thus organizations, for example, $\{a, b, aa, ba\}$. The set $\{a, aa\}$ is not closed, because b and ba can be produced

Bagley and Farmer [3] showed that autocatalytic sets can be silhouetted against a noisy background of spontaneously reacting molecules under moderate (that is, neither too low nor too high) flow.

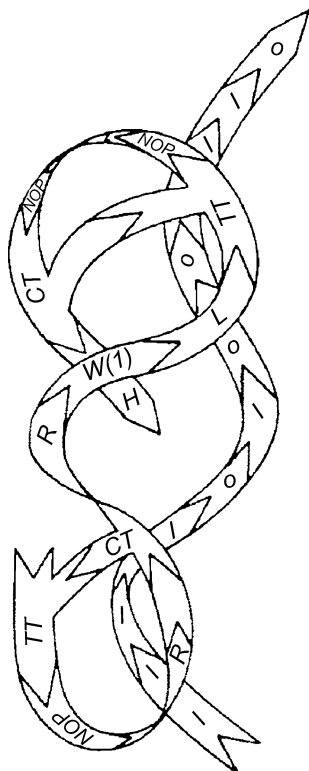
Artificial Chemistries Inspired by Turing Machines

The concept of an abstract automaton or Turing machine provides a base for a variety of structure-function mappings. In these approaches molecules are usually represented as a sequence of bits, characters, or instructions [36,51,63,86]. A sequence of bits specifies the behavior of a machine by coding for its state transition function. Thus, like in the matrix-multiplication chemistry and the lambda-chemistry, a molecule appears in two forms, namely as passive data (e.g., a binary string) and as an active machine. Also here we can call the mapping from a binary string into its machine **folding**, which might be indeterministic and may depend on other molecules (e.g., [40]).

In the early 1970s Laing [51] argued for abstract, non-analogous models in order to develop a general theory for living systems. For developing such general theory, so-called **artificial organisms** would be required. Laing suggested a series of artificial organisms [51,52,53] that should allow one to study general properties of life and thus would allow one to derive a theory which is not re-

stricted to the instance of life we observe on earth. The artificial organisms consist of different compartments, e.g., a “brain” plus “body” parts. These compartments contain binary strings as molecules. Strings are translated to a sequence of instructions forming a three-dimensional shape (cf. [78,86,94]). In order to perform a reaction, two molecules are attached such that they touch at one position (Fig. 2). One of the molecules is executed like a Turing machine, manipulating the passive data molecule as its tape. Laing proved that his artificial organisms are able to perform universal computation. He also demonstrated different forms of self-reproduction, self-description and self-inspection using his molecular machines [53].

Typogenetics is a similar approach, which was introduced in 1979 by Hofstadter in order to illustrate the “formal logic of life” [36]. Later, *typogenetics* was simulated and investigated in more detail [67,94,95]. The molecules of *typogenetics* are character sequences (called strands) over the alphabet A, C, G, T . The reaction rules are “typographic” manipulations based on a set of predefined basic operations like cutting, insertion, or deletion of characters. A sequence of such operations forms a unit (called an enzyme) which may operate on a character sequence like a Turing machine on its tape. A character string can be “translated” to an enzyme (i.e., a sequence of operations) by mapping two characters to an operation according to a predefined “genetic code”.



Artificial Chemistry, Figure 2

Illustration of Laing's molecular machines. A program molecule is associated with a data molecule. Figure from [52]

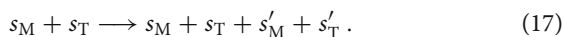
Another artificial chemistry whose reaction mechanism is inspired by the Turing machine was suggested by McCaskill [63] in order to study the self-organization of complex chemical systems consisting of catalytic polymers. Variants of this AC were realized later in a specially designed parallel reconfigurable computer based on FPGAs – Field Programmable Gate Arrays [12,18,89]. In this approach, molecules are binary strings of fixed length (e. g., 20 [63]) or of variable length [12]. As in previous approaches, a string codes for an automaton able to manipulate other strings. And again, pattern matching is used to check whether two molecules can react and to obtain the “binding site”; i. e., the location where the active molecule (machine) manipulates the passive molecule (tape). The general reaction scheme can be written as:



In experimental studies, self-replicating strings appeared frequently. In coevolution with parasites, an evolutionary arms race started among these species and the self-replicating string diversified to an extent that the para-

sites could not coadapt and went extinct. In a spatial environment (e. g., a 2-D lattice), sets of cooperating polymers evolved, interacting in a hypercyclic fashion. The authors also observed a *chemoton*-like [30] cooperative behavior, with spatially isolated, membrane-bounded evolutionary stable molecular organizations.

Machines with Fixed Tape Size In order to perform large-scale systematic simulation studies like the investigation of noise [40] or intrinsic evolution [16], it makes sense to limit the study to molecules of fixed tractable length. Ikegami and Hashimoto [40] developed an abstract artificial chemistry with two types of molecular species: 7 bit long tapes, which are mapped to 16 bit long machines. Tapes and machines form two separated populations, simulated in parallel. Reactions take place between a tape and a machine according to the following reaction scheme:



A machine s_M can react with a tape s_T , if its head matches a substring of the tape s_T and its tail matches a different substring of the tape s_T . The machine operates only between these two substrings (called reading frame) which results in a tape s'_T . The tape s'_T is then “translated” (folded) into a machine s'_M .

Ikegami and Hashimoto [40] showed that under the influence of low noise, simple autocatalytic loops are formed. When the noise level is increased, the reaction network is destabilized by parasites, but after a relatively long transient phase (about 1000 generations) a very stable, dense reaction network appears, called **core network** [40]. A core network maintains its relatively high diversity even if the noise is deactivated. The active mutation rate is high. When the noise level is very high, only small, degenerated core networks emerge with a low diversity and very low (even no) active mutation. The core networks which emerged under the influence of external noise are very stable so that their is no further development after they have appeared.

Assembler Automata

An **Assembler automaton** is like a parallel von Neumann machine. It consists of a core memory and a set of processing units running in parallel. Inspired by *Core Wars* [14], assembler automata have been used to create certain artificial life systems, such as *Coreworld* [72,73], *Tierra* [74], *Avida* [1], and *Amoeba* [69]. Although these machines have been classified as artificial chemistries [72], it is in general difficult to identify molecules or reactions. Furthermore, the assembler automaton *Tierra* has explicitly

been designed to imitate the Cambrian explosion and *not* a chemical process. Nevertheless, in some cases we can interpret the execution of an assembler automaton as a chemical process, which is especially possible in a clear way in experiments with Avida. Here, a molecule is a single assembler program, which is protected by a memory management system. The system is initialized with manually written programs that are able to self-replicate. Therefore, in a basic version of Avida only unimolecular first-order reactions occur, which are of replicator type. The reaction scheme can be written as $a \rightarrow a + \text{mutation}(a)$. The function mutation represents the possible errors that can occur while the program is self-replicating.

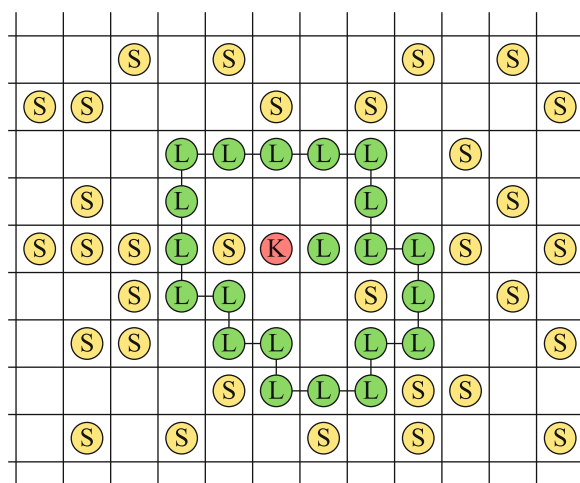
Lattice Molecular Systems

In this section systems are discussed which consist of a regular lattice, where each lattice site can hold a part (e. g. atom) of a molecule. Between parts, bonds can be formed so that a molecule covers many lattice sites. This is different to systems where a lattice site holds a complete molecule, like in Avida. The important difference is that in lattice molecular systems, the space of the molecular structure is identical to the space in which the molecules are floating around. In systems where a molecule covers just one lattice site, the molecular structure is described in a different space independently from the space in which the molecule is located.

Lattice molecular systems have been intensively used to model polymers [92], protein folding [82], and RNA structures. Besides approaches which intend to model real molecular dynamics as accurately as possible, there are also approaches which try to build abstract models. These models should give insight into statistical properties of polymers like their energy landscape and folding processes [76,77], but should not give insight into questions concerning origin and evolution self-maintaining organizations or molecular information processing. For these questions more abstract systems are studied as described in the following.

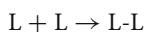
Varela, Maturana, and Uribe introduced in [93] a lattice molecular system to illustrate their concept of autopoiesis (cf. [65,99]). The system consists of a 2-D square lattice. Each lattice site can be occupied by one of the following atoms: Substrate *S*, catalyst *K*, and monomer *L*. Atoms may form bonds and thus form molecular structures on the lattice. If molecules come close to each other they may react according to the following reaction rules:

$K + 2S \rightarrow K + L$ (1) Composition: Formation of a monomer.
 $\dots -L-L + L \rightarrow \dots -L-L-L$ (2) Concatenation:



Artificial Chemistry, Figure 3

Illustration of a lattice molecular automaton that contains an autopoietic entity. Its membrane is formed by a chain of *L* monomers and encloses a catalyst *K*. Only substrate *S* may diffuse through the membrane. Substrate inside is catalyzed by *K* to form free monomers. If the membrane is damaged by disintegration of a monomer *L* it can be quickly repaired by a free monomer. See [65,93] for details



Formation of a bond between a monomer and another monomer with no more than one bond. This reaction is inhibited by a double-bounded monomer [65].

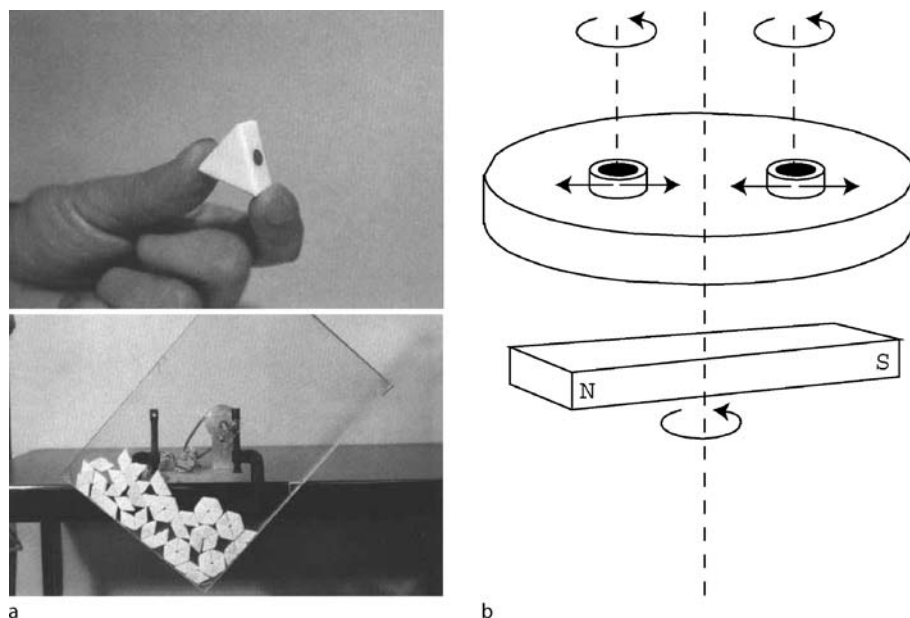


(3) Disintegration: Decomposition of a monomer

Figure 3 illustrates an autopoietic entity that may arise. Note that it is quite difficult to find the right dynamical conditions under which such autopoietic structures are stable [65,68].

Cellular Automata

Cellular automata ► [Mathematical Basis of Cellular Automata](#), [Introduction](#) can be used as a medium to simulate chemical-like processes in various ways. An obvious approach is to use the cellular automata to model space where each cell can hold a molecule (like in Avida) or atom (like in lattice molecular automata). However there are approaches where it is not clear at the onset what a molecule or reaction is. The model specification does not contain any notion of a molecule or reaction, so that an observer has to identify them. Molecules can be equated



Artificial Chemistry, Figure 4

Example of mechanical artificial chemistries. **a** Self-assembling magnetic tiles by Hosokawa et al. [38]. **b** Rotating magnetic discs by Grzybowski et al. [33]. Figures taken (and modified) from [33,38]

with self-propagating patterns like gliders, self-reproducing loops [54,80], or with the moving boundary between two homogeneous domains [37]. Note that in the latter case, particles become visible as space-time structures, which are defined by boundaries and not by connected set of cells with specific states. An interaction between these boundaries is then interpreted as a reaction.

Mechanical Artificial Chemistries

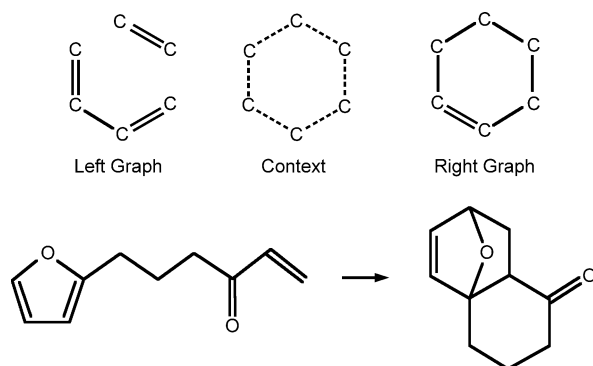
There are also physical systems that can be regarded as artificial chemistries. Hosokawa et al. [38] presented a mechanical self-assembly system consisting of triangular shapes that form bonds by permanent magnets. Interpreting attachment and detachment as chemical reactions, Hosokawa et al. [38] derived a chemical differential equation (Subject. “The Chemical Differential Equation”) modeling the kinetics of the structures appearing in the system. Another approach uses millimeter-sized magnetic discs at a liquid-air interface, subject to a magnetic field produced by a rotating permanent magnet [33]. These magnetic discs exhibit various types of structures, which might even interact resembling chemical reactions. The important difference to the first approach is that the rotating magnetic discs form dissipative structures, which require continuous energy supply.

Semi-Realistic Molecular Structures and Graph Rewriting

Recent development in artificial chemistry aims at more realistic molecular structures and reactions, like *oo chemistry* by Bersini [10] (see also [57]) or *toy chemistry* [9]. These approaches apply graph rewriting, which is a powerful and quite universal approach for defining transformations of graph-like structures [50]. In toy chemistry, molecules are represented as labeled graphs; i. e., by their structural formulas; their basic properties are derived by a simplified version of the extended Hückel MO theory that operates directly on the graphs; chemical reaction mechanisms are implemented as graph rewriting rules acting on the structural formulas; reactivities and selectivities are modeled by a variant of the frontier molecular orbital theory based on the extended Hückel scheme. Figure 5 shows an example of a graph-rewriting rule for unimolecular reactions.

Space

Many approaches discussed above assume that the reaction vessel is well-stirred. However, especially in living systems, space plays an important role. Cells, for example, are not a bag of enzymes, but spatially highly structured.



Artificial Chemistry, Figure 5

An example of a graph-rewriting rule (top) and its application to the synthesis of a bridge ring system (bottom). Figure from [9]

Moreover, it is assumed that space has played an important role in the origin of life.

Techniques for Modeling Space

In chemistry, space is usually modeled by assuming an Euclidean space (partial differential equation or particle-based) or by compartments, which we can find also in many artificial chemistries [29,70]. However, some artificial chemistries use more abstract, “computer friendly” spaces, such as a core-memory like in *Tierra* [74] or a planar triangular graph [83]. There are systems like MGS [31] that allow to specify various topological structures easily. Approaches like P-systems and membrane computing [70] allow one even to change the spatial structure dynamically ► [Membrane Computing](#).

Phenomena in Space

In general, space delays the extinction of species by fitter species, which is for example, used in *Avida* to obtain more complex evolutionary patterns [1]. Space leads usually to higher diversity. Moreover, systems that are unstable in a well-stirred reaction vessel can become stable in a spatial situation. An example is the hypercycle [20], which stabilizes against parasites when space is introduced [11]. Conversely, space can destabilize an equilibrium, leading to symmetry breaking, which has been suggested as an important mechanism underlying morphogenesis [91]. Space can support the co-existence of chemical species, which would not co-exist in a well-stirred system [49]. Space is also necessary for the formation of autopoietic structures [93] and the formation of units that undergo Darwinian evolution [24].

Theory

There is a large body of theory from domains like chemical reaction system theory [21], Petri net theory [71], and rewriting system theory (e. g., P-systems [70]), which applies also to artificial chemistries. In this section, however, we shall investigate in more detail those theoretical concepts which have emerged in artificial chemistry research.

When working with complex artificial chemistries, we are usually more interested in the qualitative than quantitative nature of the dynamics. That is, we study how the set of molecular species present in the reaction vessel changes over time, rather than studying a more detailed trajectory in concentration space.

The most prominent qualitative concept is the autocatalytic set, which has been proposed as an important element in the origin of life [19,46,75]. An **autocatalytic set** is a set of molecular species where each species is produced by at least one catalytic reaction within the set [41,47] (Fig. 1). This property has also been called self-maintaining [27] or (catalytic) closure [48]. Formally: A set of species $A \subseteq \mathcal{M}$ is called an autocatalytic set (or sometimes, catalytically closed set), if for all species¹ $a \in A$ there is a catalyst $a' \in A$, and a reaction $\rho \in \mathcal{R}$ such that a' is catalyst in ρ (i. e., $a' \in \text{LHS}(\rho)$ and $a' \in \text{RHS}(\rho)$) and ρ can take place in A (i. e., $\text{LHS}(\rho) \subseteq A$).

Example 1 (autocatalytic sets) $\mathcal{R} = \{a \rightarrow 2a, a \rightarrow a + b, a \rightarrow, b \rightarrow\}$. Molecule a catalyzes its own production and the production of b . Both molecules are subject to a dilution flow (or, equivalently, spontaneous decay), which is usually assumed in ACs studying autocatalytic sets. In this example there are three autocatalytic sets: two non-trivial autocatalytic sets $\{a\}$ and $\{a, b\}$, and the empty set $\{\}$, which is from a mathematical point of view also an autocatalytic set.

The term “autocatalytic set” makes sense only in ACs where catalysis is possible, it is not useful when applied to arbitrary reaction networks. For this reason Dittrich and Speroni di Fenizio [17] introduced a general notion of self-maintenance, which includes the autocatalytic set as a special case: Formally, given a reaction network $(\mathcal{M}, \mathcal{R})$ with $m = |\mathcal{M}|$ molecules and $r = |\mathcal{R}|$ reactions, and let $\mathbf{S} = (s_{a,\rho})$ be the $(m \times r)$ stoichiometric matrix implied by the reaction rules \mathcal{R} , where $s_{a,\rho}$ denotes the number

¹For general reaction systems, this definition has to be refined. When A contains species that are part of the inflow, like a and b in Fig. 1, but which are not produced in a catalytic way, we might want them to be part of an autocatalytic set. Assume, for example, the set $A = \{a, b, aa, ba\}$ from Fig. 1, where aa catalyzes the production of aa and ba , while using up “substrate” a and b , which are not catalytically produced.

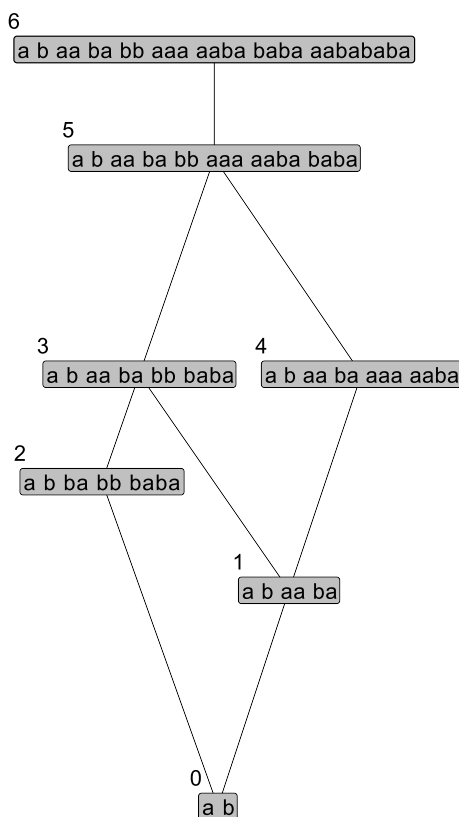
of molecules of type a produced in reaction ρ . A set of molecules $C \subseteq \mathcal{M}$ is called **self-maintaining**, if there exists a flux vector $\mathbf{v} \in \mathbb{R}^r$ such that the following three conditions apply: (1) for all reactions ρ that can take place in C (i. e., $\text{LHS}(\rho) \subseteq C$) the flux $v_\rho > 0$; (2) for all remaining reactions ρ (i. e., $\text{LHS}(\rho) \not\subseteq C$), the flux $v_\rho = 0$; and (3) for all molecules $a \in C$, the production rate $(\mathbf{Sv})_a \geq 0$. v_ρ denotes the element of \mathbf{v} describing the flux (i. e. rate) of reaction ρ . $(\mathbf{Sv})_a$ is the production rate of molecule a given flux vector \mathbf{v} . In Example 1 there are three self-maintaining (autocatalytic) sets.

Interestingly, there are self-maintaining sets that cannot make up a stationary state. In our Example 1 only two of the three self-maintaining sets are species combinations that can make up a stationary state (i. e., a state \mathbf{x}^0 for which $\mathbf{0} = \mathbf{Sv}(\mathbf{x}^0)$ holds). The self-maintaining set $\{a\}$ cannot make up a stationary state because a generates b through the reaction $a \rightarrow a + b$. Thus, there is no stationary state (of Eq. (3) with Assumptions 1 and 2) in which the concentration of a is positive while the concentration of b is zero. In order to filter out those less interesting self-maintaining sets, Fontana and Buss [27] introduced a concept taken from mathematics called closure: Formally, a set of species $A \subseteq \mathcal{M}$ is **closed**, if for all reactions ρ with $\text{LHS}(\rho) \subseteq A$ (the reactions that can take place in A), the products are also contained in A , i. e., $\text{RHS}(\rho) \subseteq A$.

Closure and self-maintenance lead to the important concept of a chemical organization [17,27]: Given an arbitrary reaction network $\langle \mathcal{M}, \mathcal{R} \rangle$, a set of molecular species that is closed and self-maintaining is called an **organization**. The importance of an organization is illustrated by a theorem roughly saying that given a fixed point of the chemical ODE (Eq. (3)), then the species with positive concentrations form an organization [17]. In other words, we have only to check those species combinations for stationary states that are organizations. The set of all organizations can be visualized nicely by a Hasse-diagram, which sketches the hierarchical (organizational) structure of the reaction network (Fig. 6). The dynamics of the artificial chemistry can then be explained within this Hasse-diagram as a movement from organization to organization [61,84].

Note that in systems under flow condition, the (finite) set of all organizations forms an algebraic lattice [17]. That is, given two organizations, there is always a unique organization union and organization intersection.

Although the concept of **autopoiesis** [93] has been described informally in quite some detail, a stringent formal definition is lacking. In a way, we can interpret the formal concepts above as an attempt to approach necessary prop-



Artificial Chemistry, Figure 6

Lattice of organizations of the autocatalytic network shown in Fig. 1. An organization is a closed and self-maintaining set of species. Two organizations are connected by a line, if one is contained in the other and there is no organization in between. The vertical position of an organization is determined by the number of species it contains

erties of an autopoietic system step by step by precise formal means. Obviously, being (contained in) at least one organization is a necessary condition for a chemical autopoietic system. But it is not sufficient. Missing are the notion of robustness and a spatial concept that formalizes a system's ability to maintain (and self-create) its own identity, e. g., through maintaining a membrane (Fig. 3).

Evolution

With artificial chemistries we can study chemical evolution. Chemical evolution (also called “pre-biotic evolution”) describes the first step in the development of life, such as the formation of complex organic molecules from simpler (in-)organic compounds [62]. Because there are no pre-biotic fossils, the study of chemical evolution has to rely on experiments [56,66] and theoretic (simulation) models [19]. Artificial chemistries aim at capturing the

constructive nature of these chemical systems and try to reproduce their evolution in computer simulations. The approaches can be distinguished by whether the evolution is driven by external operators like mutation, or whether variation and selection appears intrinsically through the chemical dynamics itself.

Extrinsic Evolution

In extrinsic approaches, an external variation operator changes the reaction network by adding, removing, or manipulating a reaction, which may also lead to the addition or removal of chemical species. In this approach, a molecule does not need to possess a structure. The following example by Jain and Krishna [41] shows that this approach allows one to create an evolving system quite elegantly:

Let us assume that the reaction network consists of m species $\mathcal{M} = \{1, \dots, m\}$. There are first-order catalytic reaction rules of the form $(i \rightarrow i + j) \in \mathcal{R}$ and a general dilution flow $(a \rightarrow) \in \mathcal{R}$ for all species $a \in \mathcal{M}$. The reaction network is completely described by a directed graph represented by the adjacency matrix $C = (c_{i,j})$, $i, j \in \mathcal{M}$, $c_{i,j} \in \{0, 1\}$, with $c_{i,j} = 1$ if molecule j catalyzes the production of i , and $c_{i,j} = 0$ otherwise. In order to avoid that self-replicators dominate the system, Jain and Krishna assume $c_{i,i} = 0$ for all molecules $i \in \mathcal{M}$.

At the beginning, the reaction network is randomly initialized, that is, (for $i \neq j$) $c_{i,j} = 1$ with probability p and $c_{i,j} = 0$, otherwise. In order to simulate the dynamics, we assume a population of molecules represented by the concentration vector $\mathbf{x} = (x_1, \dots, x_m)$, where x_i represents the current concentration of species i . The whole system is simulated in the following way:

Step 1: Simulate the chemical differential equation

$$\dot{x}_i = \sum_{j \in \mathcal{M}} c_{j,i} x_j - x_i \sum_{k \in \mathcal{M}} c_{i,k} x_k, \quad (18)$$

until a steady state is reached. Note that this steady state is usually independent of the initial concentrations, cf. [85].

Step 2: Select the “mutating” species i , which is the species with smallest concentration in the steady state.

Step 3: Update the reaction network by replacing the mutating species by a new species, which is created randomly in the same way as the initial species. That is, the entries of the i th row and i th column of the adjacency matrix C are replaced by randomly chosen entries with the same probability p as during initialization.

Step 4: Go to Step 1.

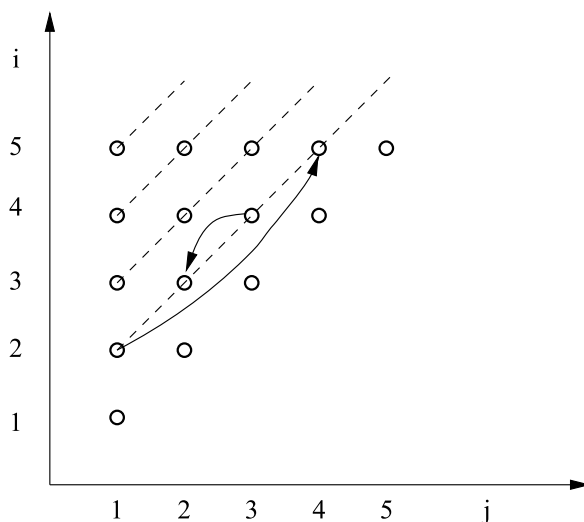
Note that there are two explicitly simulated time scales:

a slow, evolutionary time scale at which the reaction network evolves (Step 2 and 3), and a fast time scale at which the molecules catalytically react (Step 1).

In this model we can observe how an autocatalytic set inevitably appears after a period of disorder. After its arrival the largest autocatalytic set increases rapidly its connectivity until it spans the whole network. Subsequently, the connectivity converges to a steady state determined by the rate of external perturbations. The resulting highly non-random network is not fully stable, so that we can also study the causes for crashes and recoveries. For example, Jain and Krishna [44] identified that in the absence of large external perturbation, the appearance of a new *viable* species is a major cause of large extinction and recoveries. Furthermore, crashes can be caused by extinction of a “keystone species”. Note that for these observations, the new species created in Step 3 do not need to inherit any information from other species.

Intrinsic Evolution

Evolutionary phenomena can also be caused by the intrinsic dynamics of the (artificial) chemistry. In this case, external operators like those mutating the reaction network are not required. As opposed to the previous approach, we can define the whole reaction network at the onset and let only the composition of molecules present in the reaction vessel evolve. The reaction rules are (usually) de-



Artificial Chemistry, Figure 7

Illustration of syntactically and semantically closed organizations. Each organization consists of an infinite number of molecular species (connected by a dotted line). A circle represents a molecule having the structure: $\lambda x_1. \lambda x_2. \dots 27$

finied by a structure-to-function mapping similar to those described in Sect. “Structure-to-Function Mapping”. The advantage of this approach is that we need not define an external operator changing the reaction network’s topology. Furthermore, the evolution can be more realistic because, for example, molecular species that have vanished can reenter at a later time, which does not happen in an approach like the one described previously. Also, when we have a structure-to-function mapping, we can study how the structure of the molecules is related to the emergence of chemical organizations.

There are various approaches using, for example, Turing machines [63], lambda-calculus [26,27], abstract automata [16], or combinators [83] for the structure-to-function mapping.

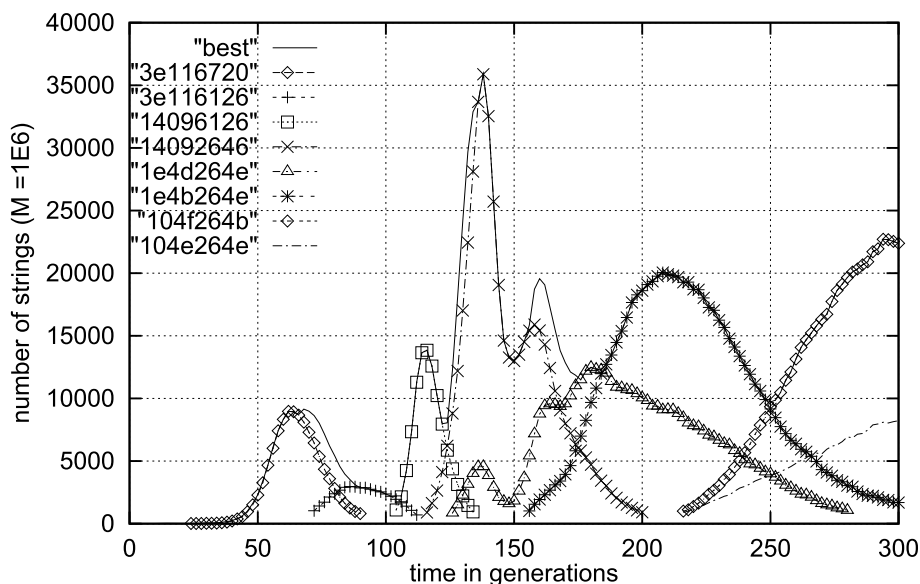
In all of those approaches we can observe an important phenomenon: While the AC evolves autocatalytic sets or more general, chemical organizations become visible. Like in the system by Jain and Krishna this effect is indicated also by an increase of the connectivity of the species within the population.

When an organization becomes visible, the system has focused on a sub-space of the whole set of possible molecules. Those molecules, belonging to the emerged or-

ganization, posses usually relatively high concentrations, because they are generated by many reactions. The closure of this set is usually smaller than the universe \mathcal{M} . Depending on the setting the emerged organization can consist of a single self-replicator, a small set of mutually producing molecules, or a large set of different molecules. Note that in the latter case the organization can be so large (and even infinite) that not all its molecules are present in the population (see Fig. 7 for an example). Nevertheless the population can carry the organization, if the system can keep a generating set of molecules within the population.

An emerged organization can be quite stable but can also change either by external perturbations like randomly inserted molecules or by internal fluctuations caused by reactions among molecules that are not part of the emerged organization, but which have remained in small quantities in the population, cf. for an example [61].

Dittrich and Banzhaf [16] have shown that it is even possible to obtain evolutionary behavior without any externally generated variation, that is, even without any inflow of random molecules. And without any explicitly defined fitness function or selection process. Selection emerges as a result of the dilution flow and the limited population size.



Artificial Chemistry, Figure 8

Example of a self-evolving artificial chemistry. The figure shows the concentration of some selected species of a reaction vessel containing approximately 10^4 different species. In this example, molecules are binary strings of length 32 bit. Two binary strings react by mapping one of them to a finite state machine operating on the second binary string. There is only a general, non-selective dilution flow. No other operators like mutation, variation, or selection are applied. Note that the structure of a new species tends to be similar to the structure of the species it has been created from. The dynamics simulated by the algorithm of the introduction. Population size $k = 10^6$ molecules. Figure from [16]

Syntactic and Semantic Closure

In many “constructive” implicitly defined artificial chemistries we can observe species appearing that share syntactical and functional similarities that are invariant under the reaction operation. Fontana and Buss [27] called this phenomenon syntactic and semantic closure.

Syntactic closure refers to the observation that the molecules within an emerged organization $O \subset \mathcal{M}$ are structurally similar. That is, they share certain structural features. This allows one to describe the set of molecules O by a formal language or grammar in a compact way. If we would instead pick a subset A from \mathcal{M} randomly, the expected length of A 's description is on the order of its size. Assume, for example, if we pick one million strings of length 100 randomly from the set of all strings of length 100, then we would need about 100 million characters to describe the resulting set. Interestingly, the organizations that appear in implicitly defined ACs can be described much more compactly by a grammar.

Syntactic and semantic closure should be illustrated with an example taken from [27], where $O \subset \mathcal{M}$ is even infinite in size. In particular experiments with the lambda-chemistry, molecules appeared that possess the following structure:

$$A_{i,j} \equiv \lambda x_1. \lambda x_2. \dots \lambda x_i. x_j \quad \text{with} \quad j \leq i, \quad (19)$$

for example $\lambda x_1. \lambda x_2. \lambda x_3. x_2$. Syntactical closure means that we can specify such structural rules specifying the molecules of O and that these structural rules are invariant under the reaction mechanism. If molecules with the structure given by Eq. (19) react, their product can also be described by Eq. (19).

Semantic closure means that we can describe the reactions taking place within O by referring to the molecule's grammatical structure (e.g. Eq. (19)). In our example, all reactions within O can be described by the following laws (illustrated by Fig. 7):

$$\begin{aligned} \forall i, j > 1, k, l: A_{i,j} + A_{k,l} &\Longrightarrow A_{i-1,j-1}, \\ \forall i > 1, j = 1, k, l: A_{i,j} + A_{k,l} &\Longrightarrow A_{k+i-1,l+i-1}. \end{aligned} \quad (20)$$

For example $\lambda x_1. \lambda x_2. \lambda x_3. \lambda x_4. x_3 + \lambda x_1. \lambda x_2. x_2 \Longrightarrow \lambda x_1. \lambda x_2. \lambda x_3. x_2$ and $\lambda x_1. \lambda x_2. x_1 + \lambda x_1. \lambda x_2. \lambda x_3. \lambda x_4. x_3 \Longrightarrow \lambda x_1. \lambda x_2. \lambda x_3. \lambda x_4. \lambda x_5. x_4$, respectively (Fig. 7). As a result of the semantic closure we need not to refer to the underlying reaction mechanism (e.g., the lambda-calculus). Instead, we can explain the reactions within O on a more abstract level by referring to the grammatical structure (e.g., Eq. (19)).

Note that syntactical and semantical closure appears also in real chemical system and is exploited by chemistry to organize chemical explanations. It might even be stated that without this phenomenon, chemistry as we know it would not be possible.

Information Processing

The chemical metaphor gives rise to a variety of computing concepts, which are explained in detail in further articles of this encyclopedia. In approaches like amorphous computing ► [Amorphous Computing](#) chemical mechanisms are used as a sub-system for communication between a huge number of simple, spatially distributed processing units. In membrane computing ► [Membrane Computing](#), rewriting operations are not only used to react molecules, but also to model active transport between compartments and to change the spatial compartment structure itself.

Beside these theoretical and in-silico artificial chemistries there are approaches that aim at using real molecules: Spatial patterns like waves and solitons in excitable media can be exploited to compute, see ► [Reaction-Diffusion Computing](#) and ► [Computing in Geometrical Constrained Excitable Chemical Systems](#). Whereas other approaches like conformational computing and DNA computing ► [DNA Computing](#) do not rely on spatially structured reaction vessels. Other closely related topics are molecular automata ► [Molecular Automata](#) and bacterial computing ► [Bacterial Computing](#).

Future Directions

Currently, we can observe a convergence of artificial chemistries and chemical models towards more realistic artificial chemistries [9]. On the one hand, models of systems biology are inspired by techniques from artificial chemistries, e.g., in the domain of rule-based modeling [22,34]. On the other hand, artificial chemistries become more realistic, for example, by adopting more realistic molecular structures or using techniques from computational chemistry to calculate energies [9,57]. Furthermore, the relation between artificial chemistries and real biological systems is made more and more explicit [25,45,58].

In the future, novel theories and techniques have to be developed to handle complex, implicitly defined reaction systems. This development will especially be driven by the needs of system biology, when implicitly defined models (i.e., rule-based models) become more frequent.

Another open challenge lies in the creation of realistic (chemical) open-ended evolutionary systems. For exam-

ple, an artificial chemistry with “true” open-ended evolution or the ability to show a satisfying long transient phase, which would be comparable to the natural process where novelties continuously appear, has not been presented yet. The reason for this might be lacking computing power, insufficient man-power to implement what we know, or missing knowledge concerning the fundamental mechanism of (chemical) evolution. Artificial chemistries could provide a powerful platform to study whether the mechanisms that we believe explain (chemical) evolution are indeed candidates.

As sketched above, there is a broad range of currently explored practical application domains of artificial chemistries in technical artifacts, such as ambient computing, amorphous computing, organic computing, or smart materials. And eventually, artificial chemistry might come back to the realm of real chemistry and inspire the design of novel computing reaction systems like in the field of molecular computing, molecular communication, bacterial computing, or synthetic biology.

Bibliography

Primary Literature

- Adami C, Brown CT (1994) Evolutionary learning in the 2D artificial life system *avida*. In: Brooks RA, Maes P (eds) *Prof artificial life IV*. MIT Press, Cambridge, pp 377–381. ISBN 0-262-52190-3
- Adleman LM (1994) Molecular computation of solutions to combinatorial problems. *Science* 266:1021
- Bagley RJ, Farmer JD (1992) Spontaneous emergence of a metabolism. In: Langton CG, Taylor C, Farmer JD, Rasmussen S (eds) *Artificial life II*. Addison-Wesley, Redwood City, pp 93–140. ISBN 0-201-52570-4
- Banâtre J-P, Métayer DL (1986) A new computational model and its discipline of programming. Technical Report RR-0566. INRIA, Rennes
- Banzhaf W (1993) Self-replicating sequences of binary numbers – foundations I and II: General and strings of length $n = 4$. *Biol Cybern* 69:269–281
- Banzhaf W (1994) Self-replicating sequences of binary numbers: The build-up of complexity. *Complex Syst* 8:215–225
- Banzhaf W (1995) Self-organizing algorithms derived from RNA interactions. In: Banzhaf W, Eeckman FH (eds) *Evolution and Biocomputing*. LNCS, vol 899. Springer, Berlin, pp 69–103
- Banzhaf W, Dittrich P, Rauhe H (1996) Emergent computation by catalytic reactions. *Nanotechnology* 7(1):307–314
- Benkő G, Flamm C, Stadler PF (2003) A graph-based toy model of chemistry. *J Chem Inf Comput Sci* 43(4):1085–1093. doi:10.1021/ci0200570
- Bersini H (2000) Reaction mechanisms in the oo chemistry. In: Bedau MA, McCaskill JS, Packard NH, Rasmussen S (eds) *Artificial life VII*. MIT Press, Cambridge, pp 39–48
- Boerlijst MC, Hogeweg P (1991) Spiral wave structure in prebiotic evolution: Hypercycles stable against parasites. *Physica D* 48(1):17–28
- Breyer J, Ackermann J, McCaskill J (1999) Evolving reaction-diffusion ecosystems with self-assembling structure in thin films. *Artif Life* 4(1):25–40
- Conrad M (1992) Molecular computing: The key-lock paradigm. *Computer* 25:11–22
- Dewdney AK (1984) In the game called core war hostile programs engage in a battle of bits. *Sci Amer* 250:14–22
- Dittrich P (2001) On artificial chemistries. Ph D thesis, University of Dortmund
- Dittrich P, Banzhaf W (1998) Self-evolution in a constructive binary string system. *Artif Life* 4(2):203–220
- Dittrich P, Speroni di Fenizio P (2007) Chemical organization theory. *Bull Math Biol* 69(4):1199–1231. doi:10.1007/s11538-006-9130-8
- Ehrlich R, Ellinger T, McCaskill JS (1997) Cooperative amplification of templates by cross-hybridization (CATCH). *Eur J Biochem* 243(1/2):358–364
- Eigen M (1971) Selforganization of matter and the evolution of biological macromolecules. *Naturwissenschaften* 58(10):465–523
- Eigen M, Schuster P (1977) The hypercycle: A principle of natural self-organisation, part A. *Naturwissenschaften* 64(11):541–565
- Érdi P, Tóth J (1989) Mathematical models of chemical reactions: Theory and applications of deterministic and stochastic models. Princeton University Press, Princeton
- Faeder JR, Blinov ML, Goldstein B, Hlavacek WS (2005) Rule-based modeling of biochemical networks. *Complexity*. doi:10.1002/cplx.20074
- Farmer JD, Kauffman SA, Packard NH (1986) Autocatalytic replication of polymers. *Physica D* 22:50–67
- Fernando C, Rowe J (2007) Natural selection in chemical evolution. *J Theor Biol* 247(1):152–167. doi:10.1016/j.jtbi.2007.01.028
- Fernando C, von Kiedrowski G, Szathmáry E (2007) A stochastic model of nonenzymatic nucleic acid replication: Elongators sequester replicators. *J Mol Evol* 64(5):572–585. doi:10.1007/s00239-006-0218-4
- Fontana W (1992) Algorithmic chemistry. In: Langton CG, Taylor C, Farmer JD, Rasmussen S (eds) *Artificial life II*. Addison-Wesley, Redwood City, pp 159–210
- Fontana W, Buss LW (1994) ‘The arrival of the fittest’: Toward a theory of biological organization. *Bull Math Biol* 56:1–64
- Fontana W, Buss LW (1996) The barrier of objects: From dynamical systems to bounded organization. In: Casti J, Karlqvist A (eds) *Boundaries and barriers*. Addison-Wesley, Redwood City, pp 56–116
- Furusawa C, Kaneko K (1998) Emergence of multicellular organisms with dynamic differentiation and spatial pattern. *Artif Life* 4:79–93
- Gánti T (1975) Organization of chemical reactions into dividing and metabolizing units: The chemotons. *Biosystems* 7(1):15–21
- Giavitto J-L, Michel O (2001) MGS: A rule-based programming language for complex objects and collections. *Electron Note Theor Comput Sci* 59(4):286–304
- Gillespie DT (1976) General method for numerically simulating stochastic time evolution of coupled chemical-reaction. *J Comput Phys* 22(4):403–434
- Grzybowski BA, Stone HA, Whitesides GM (2000) Dynamic self-

- assembly of magnetized, millimetre-sized objects rotating at a liquid-air interface. *Nature* 405(6790):1033–1036
34. Hlavacek W, Faeder J, Blinov M, Posner R, Hucka M, Fontana W (2006) Rules for modeling signal-transduction systems. *Sci STKE* 2006:re6
 35. Hofbauer J, Sigmund K (1988) Dynamical systems and the theory of evolution. University Press, Cambridge
 36. Hofstadter DR (1979) Gödel, Escher, Bach: An eternal golden braid. Basic Books Inc, New York. ISBN 0-465-02685-0
 37. Hordijk W, Crutchfield JP, Mitchell M (1996) Embedded-particle computation in evolved cellular automata. In: Toffoli T, Buafore M, Leão J (eds) *PhysComp96*. New England Complex Systems Institute, Cambridge, pp 153–8
 38. Hosokawa K, Shimoyama I, Miura H (1994) Dynamics of self-assembling systems: Analogy with chemical kinetics. *Artif Life* 1(4):413–427
 39. Hutton TJ (2002) Evolvable self-replicating molecules in an artificial chemistry. *Artif Life* 8(4):341–356
 40. Ikegami T, Hashimoto T (1995) Active mutation in self-reproducing networks of machines and tapes. *Artif Life* 2(3):305–318
 41. Jain S, Krishna S (1998) Autocatalytic sets and the growth of complexity in an evolutionary model. *Phys Rev Lett* 81(25):5684–5687
 42. Jain S, Krishna S (1999) Emergence and growth of complex networks in adaptive systems. *Comput Phys Commun* 122:116–121
 43. Jain S, Krishna S (2001) A model for the emergence of cooperation, interdependence, and structure in evolving networks. *Proc Natl Acad Sci USA* 98(2):543–547
 44. Jain S, Krishna S (2002) Large extinctions in an evolutionary model: The role of innovation and keystone species. *Proc Natl Acad Sci USA* 99(4):2055–2060. doi:10.1073/pnas.032618499
 45. Kaneko K (2007) *Life: An introduction to complex systems biology*. Springer, Berlin
 46. Kauffman SA (1971) Cellular homeostasis, epigenesis and replication in randomly aggregated macromolecular systems. *J Cybern* 1:71–96
 47. Kauffman SA (1986) Autocatalytic sets of proteins. *J Theor Biol* 119:1–24
 48. Kauffman SA (1993) *The origins of order: Self-organization and selection in evolution*. Oxford University Press, New York
 49. Kirner T, Ackermann J, Ehrlich R, McCaskill JS (1999) Complex patterns predicted in an in vitro experimental model system for the evolution of molecular cooperation. *Biophys Chem* 79(3):163–86
 50. Kniemeyer O, Buck-Sorlin GH, Kurth W (2004) A graph grammar approach to artificial life. *Artif Life* 10(4):413–431. doi:10.1162/1064546041766451
 51. Laing R (1972) Artificial organisms and autonomous cell rules. *J Cybern* 2(1):38–49
 52. Laing R (1975) Some alternative reproductive strategies in artificial molecular machines. *J Theor Biol* 54:63–84
 53. Laing R (1977) Automaton models of reproduction by self-inspection. *J Theor Biol* 66:437–56
 54. Langton CG (1984) Self-reproduction in cellular automata. *Physica D* 10D(1–2):135–44
 55. Langton CG (1989) Artificial life. In: Langton CG (ed) *Proc of artificial life*. Addison-Wesley, Redwood City, pp 1–48
 56. Lazcano A, Bada JL (2003) The 1953 Stanley L. Miller experiment: Fifty years of prebiotic organic chemistry. *Orig Life Evol Biosph* 33(3):235–42
 57. Lenaerts T, Bersini H (2009) A synthon approach to artificial chemistry. *Artif Life* 9 (in press)
 58. Lenski RE, Ofria C, Collier TC, Adami C (1999) Genome complexity, robustness and genetic interactions in digital organisms. *Nature* 400(6745):661–4
 59. Lohn JD, Colombano S, Scargle J, Stassinopoulos D, Haith GL (1998) Evolution of catalytic reaction sets using genetic algorithms. In: *Proc IEEE International Conference on Evolutionary Computation*. IEEE, New York, pp 487–492
 60. Lugowski MW (1989) Computational metabolism: Towards biological geometries for computing. In: Langton CG (ed) *Artificial Life*. Addison-Wesley, Redwood City, pp 341–368. ISBN 0-201-09346-4
 61. Matsumaru N, Speroni di Fenizio P, Centler F, Dittich P (2006) On the evolution of chemical organizations. In: Artmann S, Dittich P (eds) *Proc of the 7th german workshop of artificial life*. IOS Press, Amsterdam, pp 135–146
 62. Maynard Smith J, Szathmáry E (1995) *The major transitions in evolution*. Oxford University Press, New York
 63. McCaskill JS (1988) Polymer chemistry on tape: A computational model for emergent genetics. Internal report. MPI for Biophysical Chemistry, Göttingen
 64. McCaskill JS, Chorongiewski H, Mekelburg D, Tangen U, Gemm U (1994) Configurable computer hardware to simulate long-time self-organization of biopolymers. *Ber Bunsenges Phys Chem* 98(9):1114–1114
 65. McMullin B, Varela FJ (1997) Rediscovering computational autopoiesis. In: Husbands P, Harvey I (eds) *Fourth european conference on artificial life*. MIT Press, Cambridge, pp 38–47
 66. Miller SL (1953) A production of amino acids under possible primitive earth conditions. *Science* 117(3046):528–9
 67. Morris HC (1989) Typogenetics: A logic for artificial life. In: Langton CG (ed) *Artif life*. Addison-Wesley, Redwood City, pp 341–368
 68. Ono N, Ikegami T (2000) Self-maintenance and self-reproduction in an abstract cell model. *J Theor Biol* 206(2):243–253
 69. Pargellis AN (1996) The spontaneous generation of digital “life”. *Physica D* 91(1–2):86–96
 70. Păun G (2000) Computing with membranes. *J Comput Syst Sci* 61(1):108–143
 71. Petri CA (1962) *Kommunikation mit Automaten*. Ph D thesis, University of Bonn
 72. Rasmussen S, Knudsen C, Feldberg R, Hindsholm M (1990) The coreworld: Emergence and evolution of cooperative structures in a computational chemistry. *Physica D* 42:111–134
 73. Rasmussen S, Knudsen C, Feldberg R (1992) Dynamics of programmable matter. In: Langton CG, Taylor C, Farmer JD, Rasmussen S (eds) *Artificial life II*. Addison-Wesley, Redwood City, pp 211–291. ISBN 0-201-52570-4
 74. Ray TS (1992) An approach to the synthesis of life. In: Langton CG, Taylor C, Farmer JD, Rasmussen S (eds) *Artificial life II*. Addison-Wesley, Redwood City, pp 371–408
 75. Rössler OE (1971) A system theoretic model for biogenesis. *Z Naturforsch B* 26(8):741–746
 76. Sali A, Shakhnovich E, Karplus M (1994) How does a protein fold? *Nature* 369(6477):248–251
 77. Sali A, Shakhnovich E, Karplus M (1994) Kinetics of protein folding: A lattice model study of the requirements for folding to the native state. *J Mol Biol* 235(5):1614–1636
 78. Salzberg C (2007) A graph-based reflexive artificial chemistry. *Biosystems* 87(1):1–12

79. Sayama H (2009) Swarm chemistry. *Artif Life*. (in press)
80. Sayama H (1998) Introduction of structural dissolution into Langton's self-reproducing loop. In: Adami C, Belew R, Kitano H, Taylor C (eds) *Artificial life VI*. MIT Press, Cambridge, pp 114–122
81. Segre D, Ben-Eli D, Lancet D (2000) Compositional genomes: Prebiotic information transfer in mutually catalytic noncovalent assemblies. *Proc Natl Acad Sci USA* 97(8):4112–4117
82. Socci ND, Onuchic JN (1995) Folding kinetics of proteinlike heteropolymers. *J Chem Phys* 101(2):1519–1528
83. Speroni di Fenizio P (2000) A less abstract artificial chemistry. In: Bedau MA, McCaskill JS, Packard NH, Rasmussen S (eds) *Artificial life VII*. MIT Press, Cambridge, pp 49–53
84. Speroni Di Fenizio P, Dittrich P (2002) Artificial chemistry's global dynamics. Movement in the lattice of organisation. *J Three Dimens Images* 16(4):160–163. ISSN 1342-2189
85. Stadler PF, Fontana W, Miller JH (1993) Random catalytic reaction networks. *Physica D* 63:378–392
86. Suzuki H (2007) Mathematical folding of node chains in a molecular network. *Biosystems* 87(2–3):125–135. doi:10.1016/j.biosystems.2006.09.005
87. Suzuki K, Ikegami T (2006) Spatial-pattern-induced evolution of a self-replicating loop network. *Artif Life* 12(4):461–485. doi:10.1162/artl.2006.12.4.461
88. Suzuki Y, Tanaka H (1997) Symbolic chemical system based on abstract rewriting and its behavior pattern. *Artif Life Robotics* 1:211–219
89. Tangen U, Schulte L, McCaskill JS (1997) A parallel hardware evolvable computer polyp. In: Pocek KL, Arnold J (eds) *IEEE symposium on FPGAs for custom computing machines*. IEEE Computer Society, Los Alamitos
90. Thürk M (1993) Ein Modell zur Selbstorganisation von Automatenalgorithmen zum Studium molekularer Evolution. Ph D thesis, Universität Jena
91. Turing AM (1952) The chemical basis of morphogenesis. *Phil Trans R Soc London B* 237:37–72
92. Vanderzande C (1998) *Lattice models of polymers*. Cambridge University Press, Cambridge
93. Varela FJ, Maturana HR, Uribe R (1974) Autopoiesis: The organization of living systems. *BioSystems* 5(4):187–196
94. Varetto L (1993) Typogenetics: An artificial genetic system. *J Theor Biol* 160(2):185–205
95. Varetto L (1998) Studying artificial life with a molecular automaton. *J Theor Biol* 193(2):257–285
96. Vico G (1710) *De antiquissima Italorum sapientia ex linguae originibus eruenda libris tres*. Neapel
97. von Neumann J, Burks A (ed) (1966) *The theory of self-reproducing automata*. University of Illinois Press, Urbana
98. Zauner K-P, Conrad M (1996) Simulating the interplay of structure, kinetics, and dynamics in complex biochemical networks. In: Hofstad R, Lengauer T, Löffler M, Schomburg D (eds) *Computer science and biology GCB'96*. University of Leipzig, Leipzig, pp 336–338
99. Zeleny M (1977) Self-organization of living systems: A formal model of autopoiesis. *Int J General Sci* 4:13–28

Books and Reviews

- Adami C (1998) *Introduction to artificial life*. Springer, New York
- Dittrich P, Ziegler J, Banzhaf W (2001) Artificial chemistries – a review. *Artif Life* 7(3):225–275

- Hofbauer J, Sigmund K (1998) *Evolutionary games and population dynamics*. Cambridge University Press, Cambridge

Artificial Intelligence in Modeling and Simulation

BERNARD ZEIGLER¹, ALEXANDRE MUZY²,
LEVENT YILMAZ³

¹ Arizona Center for Integrative Modeling
and Simulation, University of Arizona, Tucson, USA

² CNRS, Université di Corsica, Corte, France

³ Auburn University, Alabama, USA

Article Outline

Glossary

Definition of the Subject

Introduction

Review of System Theory and Framework for Modeling
and Simulation

Fundamental Problems in M&S

AI-Related Software Background

AI Methods in Fundamental Problems of M&S

Automation of M&S

SES/Model Base Architecture

for an Automated Modeler/Simulationist

Intelligent Agents in Simulation

Future Directions

Bibliography

Glossary

Behavior *The observable manifestation of an interaction with a system.*

DEVS Discrete Event System Specification formalism describes models developed for simulation; applications include simulation based testing of collaborative services.

Endomorphic agents *Agents that contain models of themselves and/or of other endomorphic Agents.*

Levels of interoperability Levels at which systems can interoperate such as syntactic, semantic and pragmatic. The higher the level, the more effective is information exchange among participants.

Levels of system specification Levels at which dynamic input/output systems can be described, known, or specified ranging from behavioral to structural.

Metadata Data that describes other data; a hierarchical concept in which metadata are a descriptive abstraction above the data it describes.

Model-based automation Automation of system development and deployment that employs models or system specifications, such as DEVS, to derive artifacts.

Modeling and simulation ontology The SES is interpreted as an ontology for the domain of hierarchical, modular simulation models specified with the DEVS formalism.

Net-centric environment Network Centered, typically Internet-centered or web-centered information exchange medium.

Ontology Language that describes a state of the world from a particular conceptual view and usually pertains to a particular application domain.

Pragmatic frame A means of characterizing the consumer's use of the information sent by a producer; formalized using the concept of processing network model.

Pragmatics Pragmatics is based on Speech Act Theory and focuses on elucidating the intent of the semantics constrained by a given context. Metadata tags to support pragmatics include Authority, Urgency/Consequences, Relationship, Tense and Completeness.

Predicate logic An expressive form of declarative language that can describe ontologies using symbols for individuals, operations, variables, functions with governing axioms and constraints.

Schema An advanced form of XML document definition, extends the DTD concept.

Semantics Semantics determines the content of messages in which information is packaged. The meaning of a message is the eventual outcome of the processing that it supports.

Sensor Device that can sense or detect some aspect of the world or some change in such an aspect.

System specification Formalism for describing or specifying a system. There are levels of system specification ranging from behavior to structure.

Service-oriented architecture Web service architecture in which services are designed to be (1) accessed without knowledge of their internals through well-defined interfaces and (2) readily discoverable and composable.

Structure The internal mechanism that produces the behavior of a system.

System entity structure Ontological basis for modeling and simulation. Its pruned entity structures can describe both static data sets and dynamic simulation models.

Syntax Prescribes the form of messages in which information is packaged.

UML Unified Modeling Language is a software development language and environment that can be used for ontology development and has tools that map UML specifications into XML.

XML eXtensible Markup Language provides a syntax for document structures containing tagged information where tag definitions set up the basis for semantic interpretation.

Definition of the Subject

This article discusses the role of Artificial Intelligence (AI) in Modeling and Simulation (M&S). AI is the field of computer science that attempts to construct computer systems that emulate human problem solving behavior with the goal of understanding human intelligence. M&S is a multidisciplinary field of systems engineering, software engineering, and computer science that seeks to develop robust methodologies for constructing computerized models with the goal of providing tools that can assist humans in all activities of the M&S enterprise. Although each of these disciplines has its core community there have been numerous intersections and cross-fertilizations between the two fields. From the perspective of this article, we view M&S as presenting some fundamental and very difficult problems whose solution may benefit from the concepts and techniques of AI.

Introduction

To state the M&S problems that may benefit from AI we first briefly review a system-theory based framework for M&S that provides a language and concepts to facilitate definitive problem statement. We then introduce some key problem areas: verification and validation, reuse and composability, and distributed simulation and systems of systems interoperability. After some further review of software and AI-related background, we go on to outline some areas of AI that have direct applicability to the just given problems in M&S. In order to provide a unifying theme for the problem and solutions, we then raise the question of whether all of M&S can be automated into an integrated autonomous artificial modeler/simulationist. We then proceed to explore an approach to developing such an intelligent agent and present a concrete means by which such an agent could engage in M&S. We close with consideration of an advanced feature that such an agent must have if it is to fully emulate human capability—the ability, to a limited, but significant extent, to construct and employ models of its own “mind” as well of the “minds” of other agents.

Review of System Theory and Framework for Modeling and Simulation

Hierarchy of System Specifications

Systems theory [1] deals with a hierarchy of system specifications which defines levels at which a system may be known or specified. Table 1 shows this Hierarchy of System Specifications (in simplified form, see [2] for full exposition).

- At level 0 we deal with the input and output interface of a system.
- At level 1 we deal with purely observational recordings of the behavior of a system. This is an I/O relation which consists of a set of pairs of input behaviors and associated output behaviors.
- At level 2 we have knowledge of the initial state when the input is applied. This allows partitioning the input/output pairs of level 1 into non-overlapping subsets, each subset associated with a different starting state.
- At level 3 the system is described by state space and state transition functions. The transition function describes the state-to-state transitions caused by the inputs and the outputs generated thereupon.
- At level 4 a system is specified by a set of components and a coupling structure. The components are systems on their own with their own state set and state transition functions. A coupling structure defines how those interact. A property of a coupled system that is called “closure under coupling” guarantees that a coupled system at level 3 itself specifies a system. This property allows hierarchical construction of systems, i.e., that coupled systems can be used as components in larger coupled systems.

As we shall see in a moment, the system specification hierarchy provides a mathematical underpinning to define a framework for modeling and simulation. Each of the en-

tities (e.g., real world, model, simulation, and experimental frame) will be described as a system known or specified at some level of specification. The essence of modeling and simulation lies in establishing relations between pairs of system descriptions. These relations pertain to the validity of a system description at one level of specification relative to another system description at a different (higher, lower, or equal) level of specification.

On the basis of the arrangement of system levels as shown in Table 1, we distinguish between vertical and horizontal relations. A vertical relation is called an association mapping and takes a system at one level of specification and generates its counterpart at another level of specification. The downward motion in the structure-to-behavior direction, formally represents the process by which the behavior of a model is generated. This is relevant in simulation and testing when the model generates the behavior which then can be compared with the desired behavior.

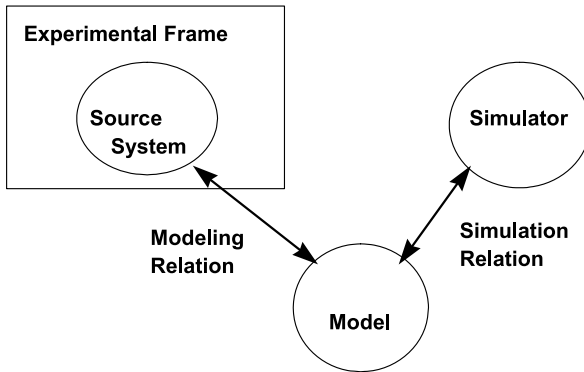
The opposite upward mapping relates a system description at a lower level with one at a higher level of specification. While the downward association of specifications is straightforward, the upward association is much less so. This is because in the upward direction information is introduced while in the downward direction information is reduced. Many structures exhibit the same behavior and recovering a unique structure from a given behavior is not possible. The upward direction, however, is fundamental in the design process where a structure (system at level 3) has to be found that is capable of generating the desired behavior (system at level 1).

Framework for Modeling and Simulation

The *Framework for M&S* as described in [2] establishes *entities* and their *relationships* that are central to the M&S enterprise (see Fig. 1. The entities of the Framework are: *source system*, *model*, *simulator*, and *experimental frame*; they are related by the *modeling* and the *simulation* relationships. Each entity is formally characterized as a sys-

Artificial Intelligence in Modeling and Simulation, Table 1
Hierarchy of system specifications

Level	Name	What we specify at this level
4	Coupled systems	System built up by several component systems that are coupled together
3	I/O System	System with state and state transitions to generate the behavior
2	I/O Function	Collection of input/output pairs constituting the allowed behavior partitioned according to the initial state the system is in when the input is applied
1	I/O Behavior	Collection of input/output pairs constituting the allowed behavior of the system from an external Black Box view
0	I/O Frame	Input and output variables and ports together with allowed values



Artificial Intelligence in Modeling and Simulation, Figure 1
Framework entities and relationships

tem at an appropriate level of specification of a generic dynamic system.

Source System

The source system is the real or virtual environment that we are interested in modeling. It is viewed as a source of observable data, in the form of time-indexed trajectories of variables. The data that has been gathered from observing or otherwise experimenting with a system is called the system behavior database. These data are viewed or acquired through experimental frames of interest to the model development and user. As we shall see, in the case of model validation, these data are the basis for comparison with data generated by a model. Thus, these data must be sufficient to enable reliable comparison as well as being accepted by both the model developer and the test agency as the basis for comparison. Data sources for this purpose might be measurement taken in prior experiments, mathematical representation of the measured data, or expert knowledge of the system behavior by accepted subject matter experts.

Experimental Frame

An experimental frame is a specification of the conditions under which the system is observed or experimented with [3]. An experimental frame is the operational formulation of the objectives that motivate a M&S project. A frame is realized as a system that interacts with the system of interest to obtain the data of interest under specified conditions.

An experimental frame specification consists of four major subsections:

Input stimuli Specification of the class of admissible input time-dependent stimuli. This is the class from

which individual samples will be drawn and injected into the model or system under test for particular experiments.

Control Specification of the conditions under which the model or system will be initialized, continued under examination, and terminated.

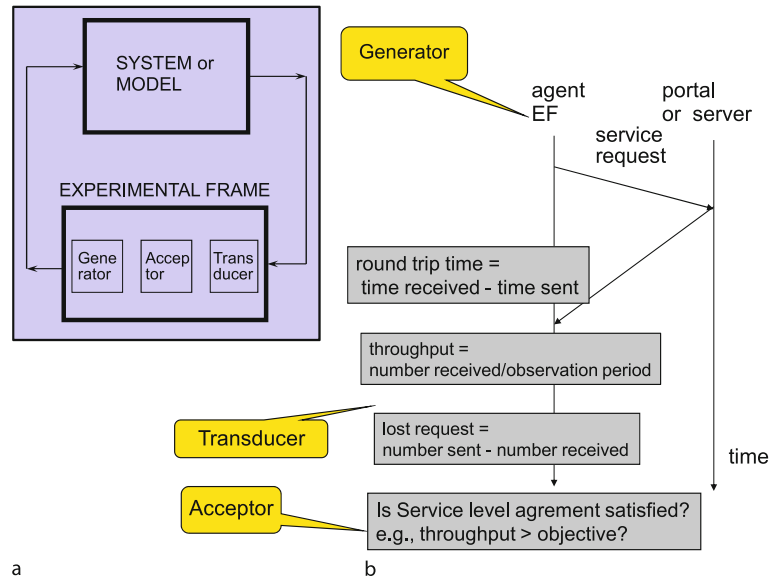
Metrics Specification of the data summarization functions and the measures to be employed to provide quantitative or qualitative measures of the input/output behavior of the model. Examples of such metrics are performance indices, goodness of fit criteria, and error accuracy bound.

Analysis Specification of means by which the results of data collection in the frame will be analyzed to arrive at final conclusions. The data collected in a frame consists of pairs of input/output time functions.

When an experimental frame is realized as a system to interact with the model or system under test the specifications become components of the driving system. For example, a generator of output time functions implements the class of input stimuli.

An experimental frame is the operational formulation of the objectives that motivate a modeling and simulation project. Many experimental frames can be formulated for the same system (both source system and model) and the same experimental frame may apply to many systems. Why would we want to define many frames for the same system? Or apply the same frame to many systems? For the same reason that we might have different objectives in modeling the same system, or have the same objective in modeling different systems. There are two equally valid views of an experimental frame. One, views a frame as a definition of the type of data elements that will go into the database. The second views a frame as a system that interacts with the system of interest to obtain the data of interest under specified conditions. In this view, the frame is characterized by its implementation as a measurement system or observer. In this implementation, a frame typically has three types of components (as shown in Fig. 2 and Fig. 3): a *generator*, that generates input segments to the system; *acceptor* that monitors an experiment to see the desired experimental conditions are met; and *transducer* that observes and analyzes the system output segments.

Figure 2b illustrates a simple, but ubiquitous, pattern for experimental frames that measure typical job processing performance metrics, such as round trip time and throughput. Illustrated in the web context, a generator produces service request messages at a given rate. The time that has elapsed between sending of a request and its return from a server is the round trip time. A transducer notes the departures and arrivals of requests allow-

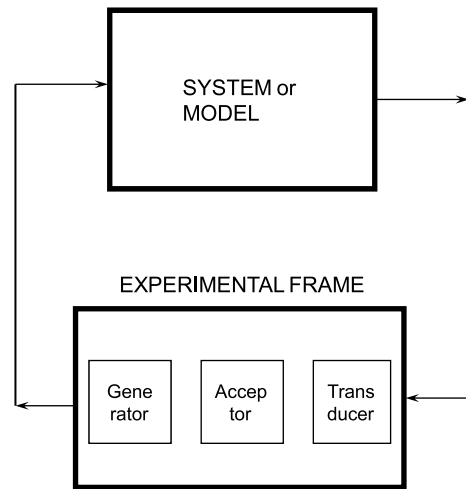


Artificial Intelligence in Modeling and Simulation, Figure 2
Experimental frame and components

ing it to compute the average round trip time and other related statistics, as well as the throughput and unsatisfied (or lost) requests. An acceptor notes whether performance achieves the developer's objectives, for example, whether the throughput exceeds the desired level and/or whether say 99% of the round trip times are below a given threshold.

Objectives for modeling relate to the role of the model in systems design, management or control. Experimental frames translate the objectives into more precise experimentation conditions for the source system or its models. We can distinguish between objectives concerning those for verification and validation of (a) models and (b) systems. In the case of models, experimental frames translate the objectives into more precise experimentation conditions for the source system and/or its models. A model under test is expected to be valid for the source system in each such frame. Having stated the objectives, there is presumably a best level of resolution to answer these questions. The more demanding the questions, the greater the resolution likely to be needed to answer them. Thus, the choice of appropriate levels of abstraction also hinges on the objectives and their experimental frame counterparts.

In the case of objectives for verification and validation of systems, we need to be given, or be able to formulate, the requirements for the behavior of the system at the IO behavior level. The experimental frame then is formulated to translate these requirements into a set of possible experiments to test whether the system actually performs its



Artificial Intelligence in Modeling and Simulation, Figure 3
Experimental frame and its components

required behavior. In addition we can formulate measures of the effectiveness (MOE) of a system in accomplishing its goals. We call such measures, *outcome* measures. In order to compute such measures, the system must expose relevant variables, we'll call these *output* variables, whose values can be observed during execution runs of the system.

Model

A model is a system specification, such as a set of instructions, rules, equations, or constraints for generating in-

put/output behavior. Models may be expressed in a variety of formalisms that may be understood as means for specifying subclasses of dynamic systems. The Discrete Event System Specification (DEVS) formalism delineates the subclass of discrete event systems and it can also represent the systems specified within traditional formalisms such as differential (continuous) and difference (discrete time) equations [4]. In DEVS, as in systems theory, a model can be *atomic*, i. e., not further decomposed, or *coupled*, in which case it consists of components that are coupled or interconnected together.

Simulator

A simulator is any computation system (such as a single processor, or a processor network, or more abstractly an algorithm), capable of executing a model to generate its behavior.

The more general purpose a simulator is the greater the extent to which it can be configured to execute a variety of model types. In order of increasing capability, simulators can be:

- Dedicated to a particular model or small class of similar models
- Capable of accepting all (practical) models from a wide class, such as an application domain (e. g., communication systems)
- Restricted to models expressed in a particular modeling formalism, such as continuous differential equation models
- Capable of accepting multi-formalism models (having components from several formalism classes, such as continuous and discrete event).

A simulator can take many forms such as on a single computer or multiple computers executing on a network.

Fundamental Problems in M&S

We have now reviewed a system-theory-based framework for M&S that provides a language and concepts in which to formulate key problems in M&S. Next on our agenda is to discuss problem areas including: verification and validation, reuse and composability, and distributed simulation and systems of systems interoperability. These are challenging, and heretofore, unsolved problems at the core of the M&S enterprise.

Validation and Verification

The basic concepts of verification and validation (V&V) have been described in different settings, levels of details,

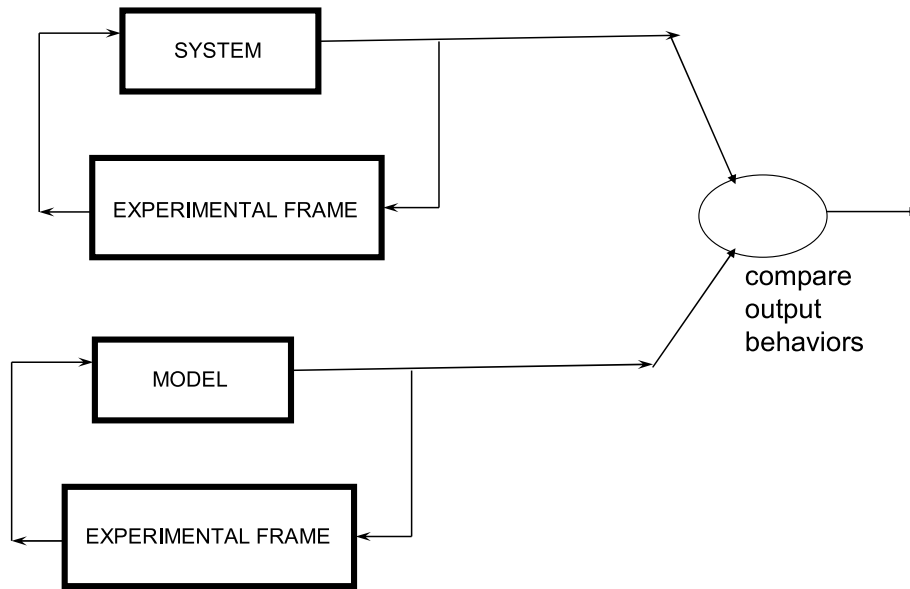
and points of views and are still evolving. These concepts have been studied by a variety of scientific and engineering disciplines and various flavors of validation and verification concepts and techniques have emerged from a modeling and simulation perspective. Within the modeling and simulation community, a variety of methodologies for V&V have been suggested in the literature [5,6,7]. A categorization of 77 verification, validation and testing techniques along with 15 principles has been offered to guide the application of these techniques [8]. However, these methods vary extensively – e. g., alpha testing, induction, cause and effect graphing, inference, predicate calculus, proof of correctness, and user interface testing and are only loosely related to one another. Therefore, such a categorization can only serve as an informal guideline for the development of a process for V&V of models and systems.

Validation and verification concepts are themselves founded on more primitive concepts such as system specifications and homomorphism as discussed in the framework of M&S [2]. In this framework, the entities *system*, *experimental frame*, *model*, *simulator* take on real importance only when properly related to each other. For example, we build a model of a particular system for some objective and only some models, and not others, are suitable. Thus, it is critical to the success of a simulation modeling effort that certain relationships hold. Two of the most important are *validity* and *simulator correctness*.

The basic modeling relation, *validity*, refers to the relation between a model, a source system and an *experimental frame*. The most basic concept, *replicative validity*, is affirmed if, for all the experiments possible within the experimental frame, the behavior of the model and system agree within acceptable tolerance. The term *accuracy* is often used in place of validity. Another term, *fidelity*, is often used for a combination of both validity and detail. Thus, a high fidelity model may refer to a model that is both highly detailed and valid (in some understood experimental frame). However, when used this way, the assumption seems to be that high detail alone is needed for high fidelity, as if validity is a necessary consequence of high detail. In fact, it is possible to have a very detailed model that is nevertheless very much in error, simply because some of the highly resolved components function in a different manner than their real system counterparts.

The basic approach to model validation is comparison of the behavior generated by a model and the source system it represents within a given experimental frame. The basis for comparison serves as the reference Fig. 4 against which the accuracy of the model is measured.

The basic *simulation* relation, simulator correctness, is a relation between a *simulator* and a *model*. A *simulator*



Artificial Intelligence in Modeling and Simulation, Figure 4
Basic approach to model validation

correctly simulates a model if it is guaranteed to faithfully generate the model's output behavior given its initial state and its input trajectory. In practice, as suggested above, simulators are constructed to execute not just one model but also a family of possible models. For example, a network simulator provides both a simulator and a class of network models it can simulate. In such cases, we must establish that a simulator will correctly execute the particular class of models it claims to support. Conceptually, the approach to testing for such execution, illustrated in Fig. 5, is to perform a number of test cases in which the same model is provided to the simulator under test and to a "gold standard simulator" which is known to correctly simulate the model. Of course such test case models must lie within the class supported by the simulator under test as well as presented in the form that it expects to receive them. Comparison of the output behaviors in the same manner as with model validation is then employed to check the agreement between the two simulators.

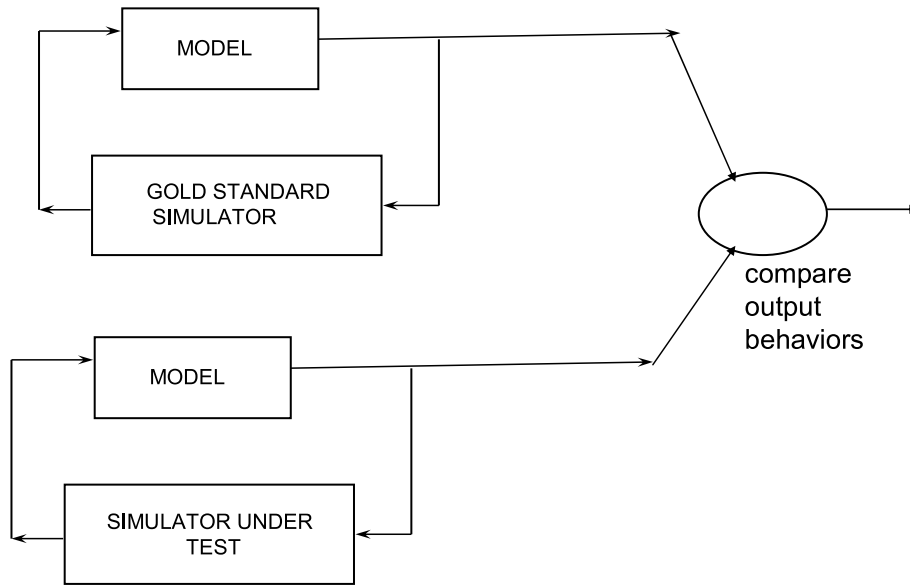
If the specifications of both the simulator and the model are available in separated form where each can be accessed independently, it may be possible to prove correctness mathematically.

The case of system validation is illustrated in Fig. 6. Here the system is considered as a hardware and/or software implementation to be validated against requirements for its input/output behavior. The goal is to develop test models that can stimulate the implemented system with inputs and can observe its outputs to compare them with

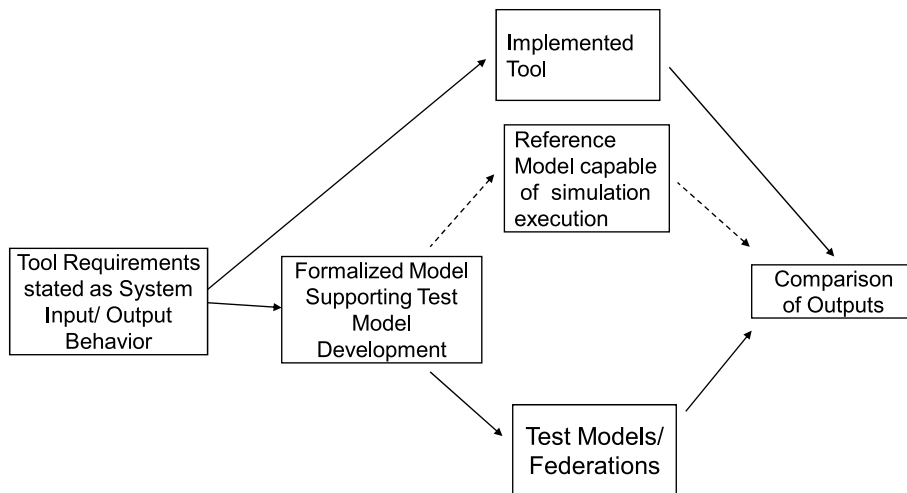
those required by the behavior requirements. Also shown is a dotted path in which a reference model is constructed that is capable of simulation execution. Construction of such a reference model is more difficult to develop than the test models since it requires not only knowing in advance what output to test for, but to actually to generate such an output. Although such a reference model is not required, it may be desirable in situations in which the extra cost of development is justified by the additional range of tests that might be possible and the consequential increased coverage this may provide.

Model Reuse and Composability

Model reuse and composability are two sides of the same coin—it is patently desirable to reuse models, the fruits of earlier or others work. However, typically such models will become components in a larger composite model and must be able to interact meaningfully with them. While software development disciplines are successfully applying a component-based approach to build software systems, the additional systems dynamics involved in simulation models has resisted straight forward reuse and composition approaches. A model is only reusable to the extent that its original dynamic systems assumptions are consistent with the constraints of the new simulation application. Consequently, without contextual information to guide selection and refactoring, a model may not be reused to advantage within a new experimental frame. Davis and



Artificial Intelligence in Modeling and Simulation, Figure 5
Basic approach to simulator verification



Artificial Intelligence in Modeling and Simulation, Figure 6
Basic approach to system validation

Anderson [9] argue that to foster such reuse, model representation methods should distinguish, and separately specify, the model, simulator, and the experimental frame. However, Yilmaz and Oren [10] pointed out that more contextual information is needed beyond the information provided by the set of experimental frames to which a model is applicable [11], namely, the characterization of the context in which the model was constructed. These authors extended the basic model-simulator-experimen-

tal frame perspective to emphasize the role of context in reuse. They make a sharp distinction between the objective context within which a simulation model is originally defined and the intentional context in which the model is being qualified for reuse. They extend the system theoretic levels of specification discussed earlier to define certain behavioral model dependency relations needed to formalize conceptual, realization, and experimental aspects of context.

As the scope of simulation applications grows, it is increasingly the case that more than one modeling paradigm is needed to adequately express the dynamics of the different components. For systems composed of models with dynamics that are intrinsically heterogeneous, it is crucial to use multiple modeling formalisms to describe them. However, combining different model types poses a variety of challenges [9,12,13]. Sarjoughian [14], introduced an approach to multi-formalism modeling that employs an interfacing mechanism called a Knowledge Interchange Broker to compose model components expressed in diverse formalisms. The KIB supports translation from the semantics of one formalism into that of a second to ensure coordinated and correct execution simulation algorithms of distinct modeling formalisms.

Distributed Simulation and System of Systems Interoperability

The problems of model reuse and composability manifest themselves strongly in the context of distributed simulation where the objective is to enable existing geographically dispersed simulators to meaningfully interact, or federate, together. We briefly review experience with interoperability in the distributed simulation context and a linguistically based approach to the System of Systems (SoS) interoperability problem [15]. Sage and Cuppan [16] drew the parallel between viewing the construction of SoS as a federation of systems and the federation that is supported by the High Level Architecture (HLA), an IEEE standard fostered by the DoD to enable composition of simulations [17,18].

HLA is a network middleware layer that supports message exchanges among simulations, called federates, in a neutral format. However, experience with HLA has been disappointing and forced acknowledging the difference between technical interoperability and substantive interoperability [19]. The first only enables heterogeneous simulations to exchange data but does not guarantee the second, which is the desired outcome of exchanging meaningful data, namely, that coherent interaction among federates takes place. Tolk and Muguirra [20] introduced the Levels of Conceptual Interoperability Model (LCIM) which identified seven levels of interoperability among participating systems. These levels can be viewed as a refinement of the *operational* interoperability type which is one of three defined by [15]. The operational type concerns linkages between systems in their interactions with one another, the environment, and with users. The other types apply to the context in which systems are constructed and acquired. They are *constructive*—relating

to linkages between organizations responsible for system construction and *programmatic*—linkages between program offices to manage system acquisition.

AI-Related Software Background

To proceed to the discussion of the role of AI in addressing key problems in M&S, we need to provide some further software and AI-related background. We offer a brief historical account of object-orientation and agent-based systems as a springboard to discuss the upcoming concepts of object frameworks, ontologies and endomorphic agents.

Object-Orientation and Agent-Based Systems

Many of the software technology advances of the last 30 years have been initiated from the field of M&S. Objects, as code modules with both structure and behavior, were first introduced in the SIMULA simulation language [21]. Objects blossomed in various directions and became institutionalized in the widely adopted programming language C++ and later in the infrastructure for the web in the form of Java [22] and its variants. The freedom from straight-line procedural programming that object-orientation championed was taken up in AI in two directions: various forms of knowledge representation and of autonomy. Rule-based systems aggregate modular if-then logic elements—the rules—that can be activated in some form of causal sequence (inference chains) by an execution engine [23]. In their passive state, rules represent static discrete pieces of inferential logic, called declarative knowledge. However, when activated, a rule influences the state of the computation and the activation of subsequent rules, providing the system a dynamic or procedural, knowledge characteristic as well. Frame-based systems further expanded knowledge representation flexibility and inferencing capability by supporting slots and constraints on their values—the frames—as well as their taxonomies based on generalization/specialization relationships [24]. Convergence with object-orientation became apparent in that frames could be identified as objects and their taxonomic organization could be identified with classes within object-style organizations that are based on sub-class hierarchies.

On the other hand, the modular nature of objects together with their behavior and interaction with other objects, led to the concept of agents which embodied increased autonomy and self-determination [25]. Agents represent individual threads of computation and are typically deployable in distributed form over computer networks where they interact with local environments and communicate/coordinate with each other. A wide variety

of agent types exists in large part determined by the variety and sophistication of their processing capacities—ranging from agents that simply gather information on packet traffic in a network to logic-based entities with elaborate reasoning capacity and authority to make decisions (employing knowledge representations just mentioned.) The step from agents to their aggregates is natural, thus leading to the concept of multi-agent systems or societies of agents, especially in the realm of modeling and simulation [26].

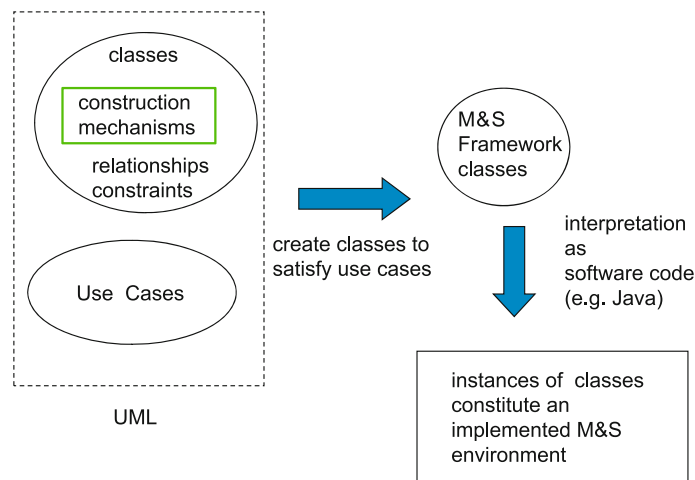
To explore the role of AI in M&S at the present, we project the just-given historical background to the concurrent concepts of object frameworks, ontologies and endomorphic agents. The Unified Modeling Language (UML) is gaining a strong foothold as the defacto standard for object-based software development. Starting as a diagrammatical means of software system representation, it has evolved to a formally specified language in which the fundamental properties of objects are abstracted and organized [27,28]. Ontologies are models of the world relating to specific aspects or applications that are typically represented in frame-based languages and form the knowledge components for logical agents on the Semantic Web [29]. A convergence is underway that re-enforces the commonality of the object-based origins of AI and software engineering. UML is being extended to incorporate ontology representations so that software systems in general will have more explicit models of their domains of operation. As we shall soon see, endomorphic agents refer to agents that include abstractions of their own structure and behavior within their ontologies of the world [30].

The M&S Framework

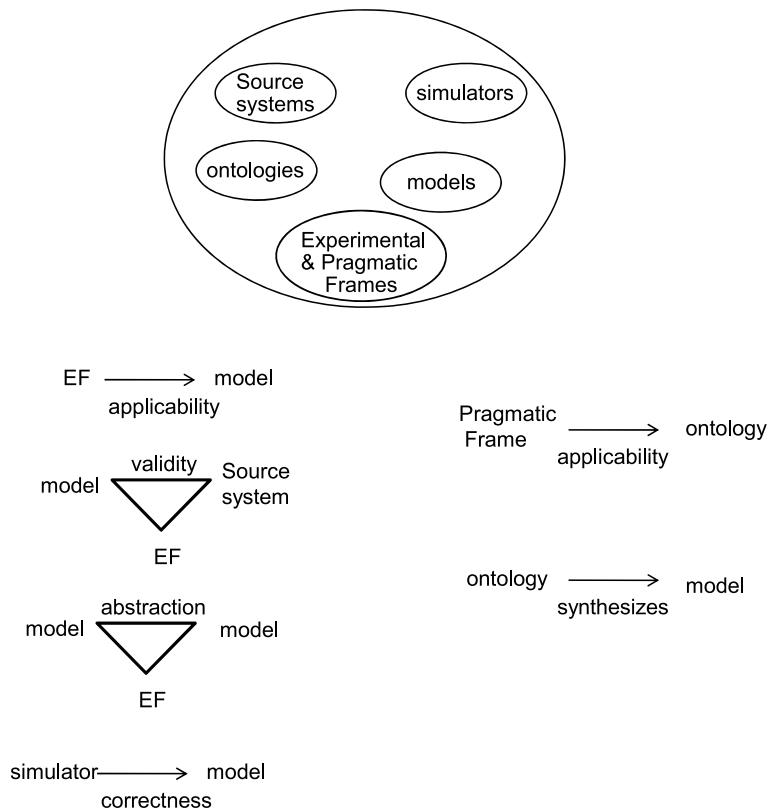
Within Unified Modeling Language (UML)

With object-orientation as unified by UML and some background in agent-based systems, we are in a position to discuss the computational realization of the M&S framework discussed earlier. The computational framework is based on the Discrete Event System Specification (DEVS) formalism and implemented in various object-oriented environments. Using UML we can represent the framework as a set of classes and relations as illustrated in Figs. 7 and 8. Various software implementations of DEVS support different subsets of the classes and relations. In particular, we mention a recent implementation of DEVS within a Service-Oriented Architecture (SOA) environment called DEVS/SOA [31,32]. This implementation exploits some of the benefits afforded by the web environment mentioned earlier and provides a context for consideration of the primary target of our discussion, comprehensive automation of the M&S enterprise.

We use one of the UML constructs, the use case diagram, to depict the various capabilities that would be involved in automating all or parts of the M&S enterprise. Use cases are represented by ovals that connect to at least one actor (stick figure) and to other use cases through “includes” relations, shown as dotted arrows. For example, a sensor (actor) collects data (use case) which includes storage of data (use case). A memory actor stores and retrieves models which include storage and retrieval (respectively) of data. Constructing models includes retrieval



Artificial Intelligence in Modeling and Simulation, Figure 7
M&S framework formulated within UML



Artificial Intelligence in Modeling and Simulation, Figure 8
M&S framework classes and relations in a UML representation

ing stored data within an experimental frame. Validating models includes retrieving models from memory as components and simulating the composite model to generate data within an experimental frame. The emulator-simulator actor does the simulating to execute the model so that its generated behavior can be matched against the stored data in the experimental frame. The objectives of the human modeler drive the model evaluator and hence the choice of experimental frames to consider as well as models to validate. Models can be used in (at least) two time frames [33]. In the long term, they support planning of actions to be taken in the future. In the short term, models support execution and control in real-time of previously planned actions.

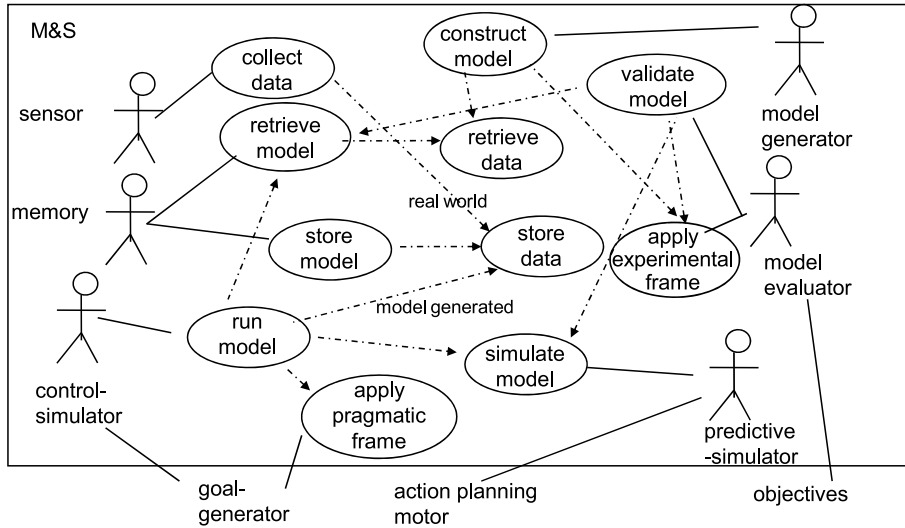
AI Methods in Fundamental Problems of M&S

The enterprise of modeling and simulation is characterized by activities such as model, simulator and experimental frame creation, construction, reuse, composition, verification and validation. We have seen that valid

model construction requires significant expertise in all the components of the M&S enterprise, e.g., modeling formalisms, simulation methods, and domain understanding and knowledge. Needless to say, few people can bring all such elements to the table, and this situation creates a significant bottleneck to progress in such projects. Among the contributing factors are lack of trained personnel that must be brought in, expense of such high capability experts, and the time needed to construct models to the resolution required for most objectives. This section introduces some AI-related technologies that can ameliorate this situation: Service-Oriented Architecture and Semantic Web, ontologies, constrained natural language capabilities, and genetic algorithms. Subsequently we will consider these as components in unified, comprehensive, and autonomous automation of M&S.

Service-Oriented Architecture and Semantic Web

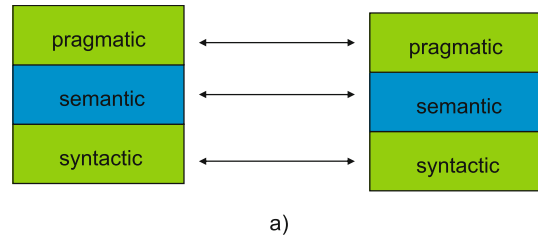
On the World Wide Web, a Service-Oriented Architecture is a market place of open and discoverable web-services



Artificial Intelligence in Modeling and Simulation, Figure 9
UML use case formulation of the overall M&S enterprise

incorporating, as they mature, Semantic Web technologies [34]. The eXtensible Markup Language (XML) is the standard format for encoding data sets and there are standards for sending and receiving XML [35]. Unfortunately, the problem just starts at this level. There are myriad ways, or Schemata, to encode data into XML and a good number of such Schemata have already been developed. More often than not, they are different in detail when applied to the same domains. What explains this incompatibility?

In a Service-Oriented Architecture, the producer sends messages containing XML documents generated in accordance with a schema. The consumer receives and interprets these messages using the same schema in which they were sent. Such a message encodes a world state description (or changes in it) that is a member of a set delineated by an ontology. The ontology takes into account the pragmatic frame, i.e., a description of how the information will be used in downstream processing. In a SOA environment, data dissemination may be dominated by “user pull of data,” incremental transmission, discovery using metadata, and automated retrieval of data to meet user pragmatic frame specifications. This is the SOA concept of data-centered, interface-driven, loose coupling between producers and consumers. The SOA concept requires the development of platform-independent, community-accepted, standards that allow raw data to be syntactically packaged into XML and accompanied by metadata that describes the semantic and pragmatic information needed to effectively process the data into increasingly higher-value products downstream.



Artificial Intelligence in Modeling and Simulation, Figure 10
Interoperability levels in distributed simulation

Ontologies

Semantic Web researchers typically seek to develop intelligent agents that can draw logical inferences from diverse, possibly contradictory, ontologies such as a web search might discover. Semantic Web research has led to a focus on ontologies [34]. These are logical languages that provide a common vocabulary of terms and axiomatic relations among them for a subject area. In contrast, the newly emerging area of ontology integration assumes that human understanding and collaboration will not be replaced by intelligent agents. Therefore, the goal is to create concepts and tools to help people develop practical solutions to incompatibility problems that impede “effective” exchange of data and ways of testing that such solutions have been correctly implemented.

As illustrated in Fig. 10, interoperability of systems can be considered at three linguistically-inspired levels: *syntactic*, *semantic*, and *pragmatic*. The levels are summarized in Table 2. More detail is provided in [36].

Artificial Intelligence in Modeling and Simulation, Table 2
Linguistic levels

Linguistic Level	A collaboration of systems or services interoperates at this level if:	Examples
Pragmatic – how information in messages is used	The receiver reacts to the message in a manner that the sender intends	An order from a commander is obeyed by the troops in the field as the commander intended. A necessary condition is that the information arrives in a timely manner and that its meaning has been preserved (semantic interoperability)
Semantic – shared understanding of meaning of messages	The receiver assigns the same meaning as the sender did to the message.	An order from a commander to multi-national participants in a coalition operation is understood in a common manner despite translation into different languages. A necessary condition is that the information can be unequivocally extracted from the data (syntactic interoperability)
Syntactic – common rules governing composition and transmitting of messages	The consumer is able to receive and parse the sender's message	A common network protocol (e.g. IPv4) is employed ensuring that all nodes on the network can send and receive data bit arrays adhering to a prescribed format.

Constrained Natural Language

Model development can be substantially aided by enabling users to specify modeling constructs using some form of constrained natural language [37]. The goal is to overcome modeling complexity by letting users with limited or nonexistent formal modeling or programming background convey essential information using natural language, a form of expression that is natural and intuitive. Practicality demands constraining the actual expressions that can be used so that the linguistic processing is tractable and the input can be interpreted unambiguously. Some techniques allow the user to narrow down essential components for model construction. Their goal is to reduce ambiguity between the user's requirements and essential model construction components. A natural language interface allows model specification in terms of a verb phrase consisting of a verb, noun, and modifier, for example "build car quickly." Conceptual realization of a model from a verb phrase ties in closely with Checkland's [38] insight that an appropriate verb should be used to express the root definition, or core purpose, of a system. The main barrier between many people and existing modeling software is their lack of computer literacy and this provides an incentive to develop natural language interfaces as a means of bridging this gap. Natural language expression could create modelers out of people who think semantically, but do not have the requisite computer skills to express these ideas. A semantic representation frees the user to explore the system on the familiar grounds of natural language and opens the way for brain storming, innovation and testing of models before they leave the drawing board.

Genetic Algorithms

The genetic algorithm is a subset of evolutionary algorithms that model biological processes to search in highly complex spaces. A genetic algorithm (GA) allows a population composed of many individuals to evolve under specified selection rules to a state that maximizes the "fitness." The theory was developed by John Holland [39] and popularized by Goldberg who was able to solve a difficult problem involving the control of gas pipeline transmission for his dissertation [40]. Numerous applications of GAs have since been chronicled [41,42]. Recently, GAs have been applied to cutting-edge problems in automated construction of simulation models, as discussed below [43].

Automation of M&S

We are now ready to suggest a unifying theme for the problems in M&S and possible AI-based solutions, by raising the question of whether all of M&S can be automated into an integrated autonomous artificial modeler/simulationist. First, we provide some background needed to explore an approach to developing such an intelligent agent based on the System Entity Structure/Model Base framework, a hybrid methodology that combines elements of AI and M&S.

System Entity Structure

The System Entity Structure (SES) concepts were first presented in [44]. They were subsequently extended and implemented in a knowledge-based design environment [45]. Application to model base management originated with [46]. Subsequent formalizations and imple-

mentations were developed in [47,48,49,50,51]. Applications to various domains are given in [52].

A System Entity Structure is a knowledge representation formalism which focuses on certain elements and relationships that relate to M&S. Entities represent things that exist in the real world or sometimes in an imagined world. Aspects represent ways of decomposing things into more fine grained ones. Multi-aspects are aspects for which the components are all of one kind. Specializations represent categories or families of specific forms that a thing can assume provides the means to represent a family of models as a labeled tree. Two of its key features are support for decomposition and specialization. The former allows decomposing a large system into smaller systems. The latter supports representation of alternative choices. Specialization enables representing a generic model (e.g., a computer display model) as one of its specialized variations (e.g., a flat panel display or a CRT display). On the basis of SES axiomatic specifications, a family of models (design-space) can be represented and further automatically pruned to generate a simulation model. Such models can be systematically studied and experimented with based on alternative design choices. An important, salient feature of SES is its ability to represent models not only in terms of their decomposition and specialization, but also their aspects. The SES represents alternative decompositions via aspects. The system entity structure (SES) formalism provides an operational language for specifying such hierarchical structures. An SES is a structural knowledge representation scheme that systematically organizes a family of possible structures of a system. Such a family characterizes decomposition, coupling, and taxonomic relationships among entities. An *entity* represents a real world object. The decomposition of an entity concerns how it may be broken down into sub-entities. In addition, coupling specifications tell how sub-entities may be coupled together to reconstitute the entity and be associated with an aspect. The *taxonomic* relationship concerns admissible variants of an entity. The SES/Model-Base framework [52] is a powerful means to support the plan-generate-evaluate paradigm in systems design. Within the framework, entity structures organize models in a model base. Thus, modeling activity within the framework consists of three sub-activities: specification of model composition structure, specification of model behavior, and synthesis of a simulation model.

The SES is governed by an axiomatic framework in which entities alternate with the other items. For example, a thing is made up of parts; therefore, its entity representation has a corresponding aspect which, in turn, has entities representing the parts. A System Entity Structure specifies

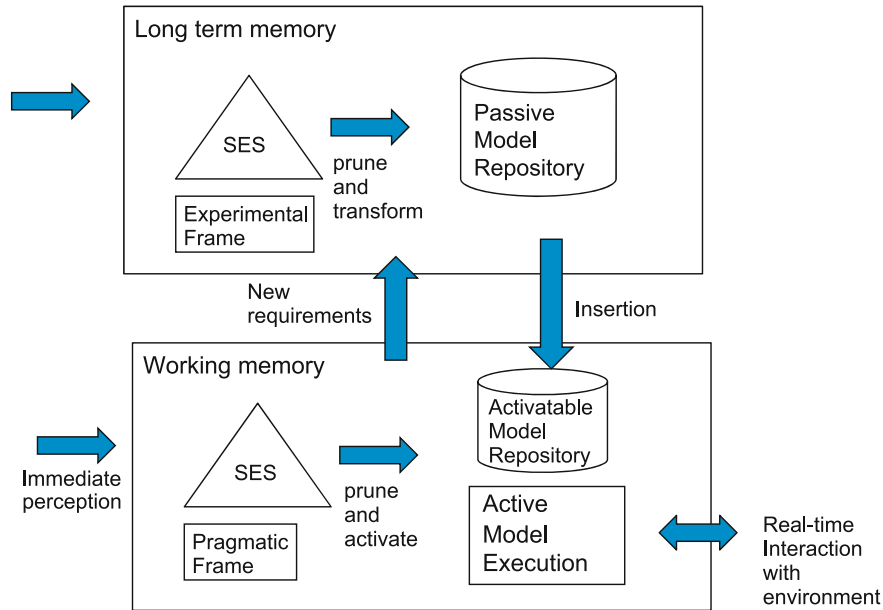
a family of hierarchical, modular simulation models, each of which corresponds to a complete pruning of the SES. Thus, the SES formalism can be viewed as an ontology with the set of all simulation models as its domain of discourse. The mapping from SES to the Systems formalism, particularly to the DEVS formalism, is discussed in [36]. We note that simulation models include both static and dynamic elements in any application domain, hence represent an advanced form of ontology framework.

SES/Model Base Architecture for an Automated Modeler/Simulationist

In this section, we raise the challenge of creating a fully automated modeler/simulationist that can autonomously carry out all the separate functions identified in the M&S framework as well as the high level management of these functions that is currently under exclusively human control. Recall the use case diagrams in Fig. 9 that depict the various capabilities that would need to be involved in realizing a completely automated modeler/simulationist. To link up with the primary modules of mind, we assign model construction to the belief generator—interpreting beliefs as models [54]. Motivations outside the M&S component drive the belief evaluator and hence the choice of experimental frames to consider as well as models to validate. External desire generators stimulate the imaginer/envisioner to run models to make predictions within pragmatic frames and assist in action planning.

The use case diagram of Fig. 9, is itself a model of how modeling and simulation activities may be carried out within human minds. We need not be committed to particular details at this early stage, but will assume that such a model can be refined to provide a useful representation of human mental activity from the perspective of M&S. This provides the basis for examining how such an artificially intelligent modeler/simulationist might work and considering the requirements for comprehensive automation of M&S.

The SES/MB methodology, introduced earlier, provides a basis for formulating a conceptual architecture for automating all of the M&S activities depicted earlier in Fig. 9 into an integrated system. As illustrated in Fig. 11, the dichotomy into real-time use of models for acting in the real world and the longer-term development of models that can be employed in such real-time activities is manifested in the distinction between passive and active models. Passive models are stored in a repository that can be likened to long-term memory. Such models under go the life-cycle mentioned earlier in which they are validated and employed within experimental frames of interest for



Artificial Intelligence in Modeling and Simulation, Figure 11
SES/Model base architecture for automated M&S

long-term forecasting or decision-making. However, in addition to this quite standard concept of operation, there is an input pathway from the short term, or working, memory in which models are executed in real-time. Such execution, in real-world environments, often results in deficiencies, which provide impetus and requirements for instigating new model construction. Whereas long-term model development and application objectives are characterized by experimental frames, the short-term execution objectives are characterized by pragmatic frames. As discussed in [36], a pragmatic frame provides a means of characterizing the use for which an executable model is being sought. Such models must be simple enough to execute within, usually, stringent real-time deadlines.

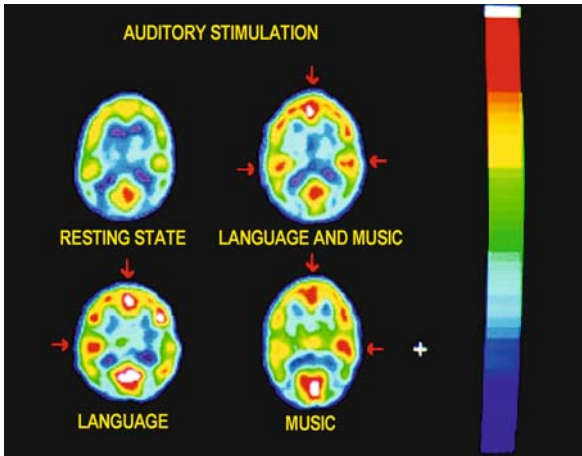
The M&S Framework Within Mind Architecture

An influential formulation of recent work relating to mind and brain [53] views mind as the behavior of the brain, where mind is characterized by a massively modular architecture. This means that mind is, composed of a large number of modules, each responsible for different functions, and each largely independent and sparsely connected with others. Evolution is assumed to favor such differentiation and specialization since under suitably weakly interactive environments they are less redundant and more efficient in consuming space and time resources. Indeed, this formulation is reminiscent of the class of prob-

lems characterized by Systems of Systems (SoS) in which the attempt is made to integrate existing systems, originally built to perform specific functions, into a more comprehensive and multifunctional system. As discussed in [15], the components of each system can be viewed as communicating with each other within a common ontology, or model of the world that is tuned to the smooth functioning of the organization. However, such ontologies may well be mismatched to support integration at the systems level. Despite working on the results of a long history of pre-human evolution, the fact that consciousness seems to provide a unified and undifferentiated picture of mind, suggests that human evolution has to a large extent solved the SoS integration problem.

The Activity Paradigm for Automated M&S

At least for the initial part of its life, a modeling agent needs to work on a "first order" assumption about its environment, namely, that it can exploit only semantics-free properties [39]. Regularities, such as periodic behaviors, and stimulus-response associations, are one source of such semantics-free properties. In this section, we will focus on a fundamental property of systems, such as the brain's neural network, that have a large number of components. This distribution of activity of such a system over space and time provides a rich and semantics-free substrate from which models can be generated. Proposing structures and



Artificial Intelligence in Modeling and Simulation, Figure 12
Brain module activities

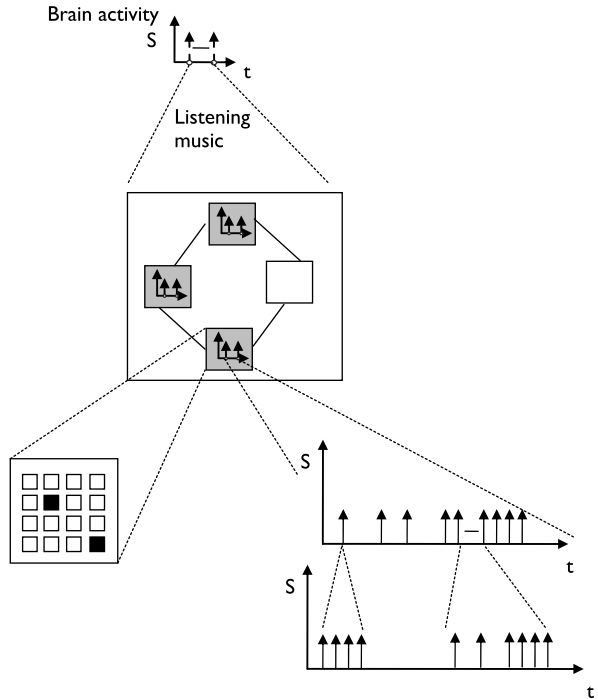
algorithms to track and replicate this activity should support automated modeling and simulation of patterns of reality. The goal of the activity paradigm is to extract mechanisms from natural phenomena and behaviors to automate and guide the M&S process.

The brain offers a quintessential illustration of activity and its potential use to construct models. Figure 12 describes brain electrical activities [54]. Positron emission tomography (PET) is used to record electrical activity inside the brain. The PET method scans show what happens inside the brain when resting and when stimulated by words and music. The red areas indicate high brain activities. Language and music produce responses in opposite sides of the brain (showing the sub-system specializations). There are many levels of activity (ranging from low to high.)

There is a strong link between modularity and the applicability of activity measurement as a useful concept. Indeed, modules represent loci for activity—a distribution of activity would not be discernable over a network were there no modules that could be observed to be in different states of activity. As just illustrated, neuroscientists are exploiting this activity paradigm to associate brain areas with functions and to gain insight into areas that are active or inactive over different, but related, functions, such as language and music processing. We can generalize this approach as a paradigm for an automated modeling agent.

Component, Activity and Discrete-Event Abstractions

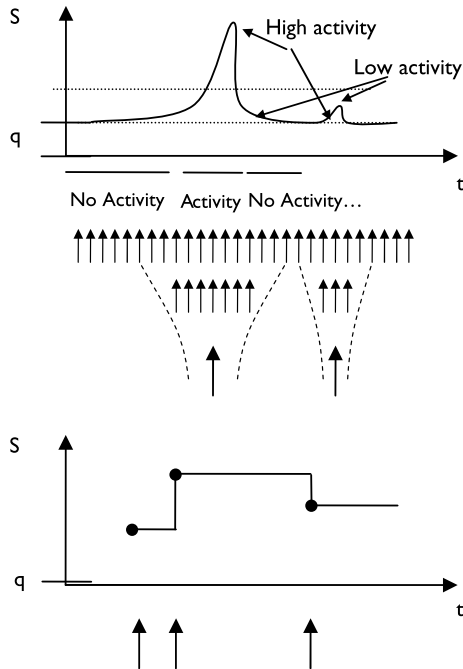
Figure 13 depicts a brain description through components and activities. First, the modeler considers one brain ac-



Artificial Intelligence in Modeling and Simulation, Figure 13
Hierarchy of components, activity and discrete-event abstractions

tivity (e.g., listening music.) This first level activity corresponds to simulation components at a lower level. At this level, activity of the components can be considered to be only on (grey boxes) or off (white boxes.) At lower levels, structure and behaviors can be decomposed. The structure of the higher component level is detailed at lower levels (e.g., to the neuronal network). The behavior is also detailed through activity and discrete-event abstraction [55]. At the finest levels, activity of components can be detailed.

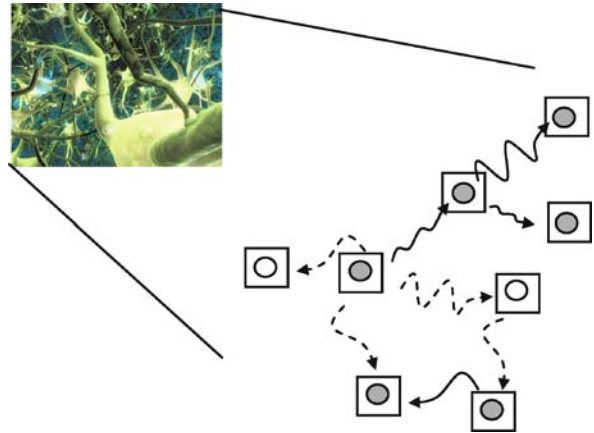
Using the pattern detection and quantization methods continuously changing variables can also be treated within the activity paradigm. As illustrated in Fig. 14, small slopes and small peaks can signal low activity whereas high slopes and peaks can signal high activity levels. To provide for scale, discrete event abstraction can be achieved using quantization [56]. To determine the activity level, a *quantum* or measure of significant change has to be chosen. The quantum size acts as a filter on the continuous flow. For example, one can notice that in the figure, using the displayed quantum, the smallest peaks will not be significant. Thus, different levels of resolution can be achieved by employing different quantum sizes. A genetic algorithm can be used to find the optimum such level of resolution given for a given modeling objective [43].



Artificial Intelligence in Modeling and Simulation, Figure 14
Activity sensitivity and discrete-events

Activity Tracking

Within the activity paradigm, M&S consists of capturing activity paths through component processing and transformations. To determine the basic structure of the whole system, an automated modeler has to answer questions of the form: *where and how is activity produced, received, and transmitted?* Figure 15 represents a component-based view of activity flow in a neuronal network. Activity paths through components are represented by full arrows. Activity is represented by full circles. Components are represented by squares. The modeler must generate such a graph based on observed data—but how does it obtain such data? One approach is characteristic of current use of PET scans by neuroscientists. This approach exploits a relationship between activity and energy—activity requires consumption of energy, therefore, observing areas of high energy consumption signals areas of high activity. Notice that this correlation requires that energy consumption be localizable to modules in the same way that activity is so localized. So for example, current computer architectures that provide a single power source to all components do not lend themselves to such observation. How activity is passed on from component to component can be related to the modularity styles (none, weak, strong) of the components. Concepts relating such modularity styles to activity transfer need to be developed to support an activity



Artificial Intelligence in Modeling and Simulation, Figure 15
Activity paths in neurons

tracking methodology that goes beyond reliance on energy correlation.

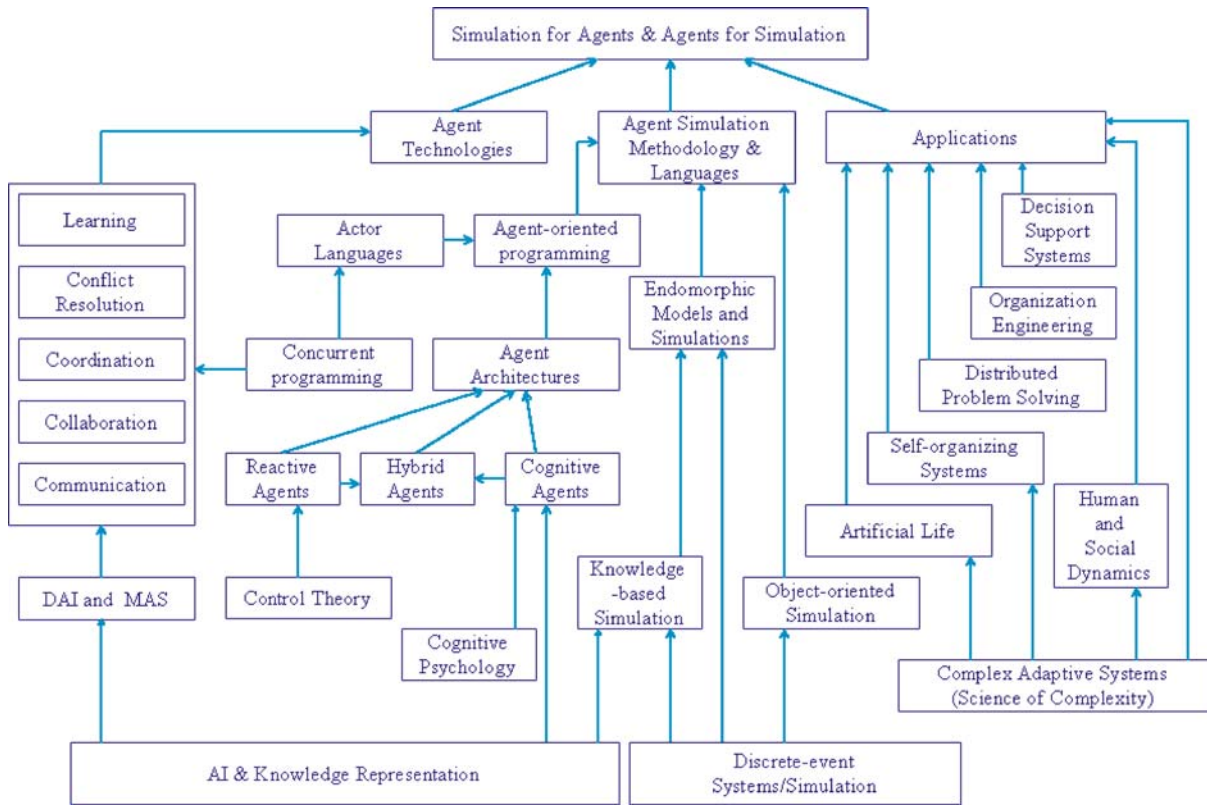
Activity Model Validation

Recall that having generated a model of an observed system (whether through activity tracking or by other means), the next step is validation. In order to perform such validation, the modeler needs an approach to generating activity profiles from simulation experiments on the model and to comparing these profiles with those observed in the real system. Muzy and Nutaro [57] have developed algorithms that exploit activity tracking to achieve efficient simulation of DEVS models. These algorithms can be adopted to provide an activity tracking pattern applicable to a given simulation model to extract its activity profiles for comparison with those of the modeled system.

A forthcoming monograph will develop the activity paradigm for M&S in greater detail [58].

Intelligent Agents in Simulation

Recent trends in technology as well as the use of simulation in exploring complex artificial and natural information processes [62,63] have made it clear that simulation model fidelity and complexity will continue to increase dramatically in the coming decades. The dynamic and distributed nature of simulation applications, the significance of exploratory analysis of complex phenomena [64], and the need for modeling the micro-level interactions, collaboration, and cooperation among real-world entities is bringing a shift in the way systems are being conceptualized. Using intelligent agents in simulation models is based on the idea that it is possible to represent the behavior of active entities in the world in terms of the interactions of



Artificial Intelligence in Modeling and Simulation, Figure 16
Evolution of the use of intelligent agents in simulation

an assembly of agents with their own operational autonomy.

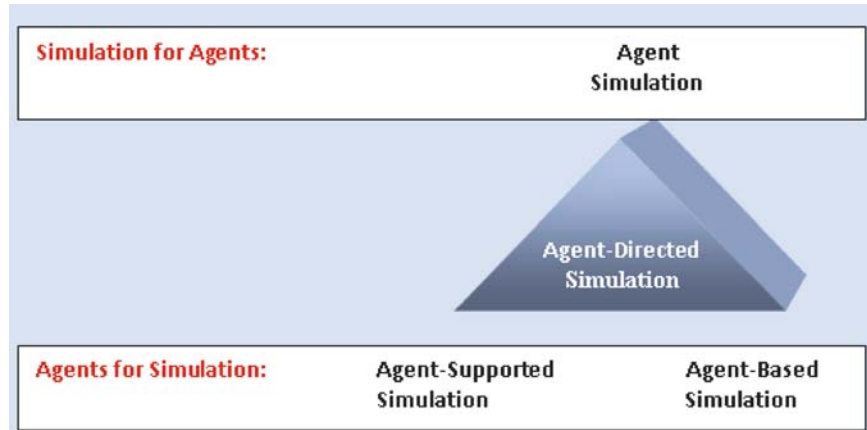
The early pervading view on the use of agents in simulation stems from the developments in Distributed Artificial Intelligence (DAI), as well as advances in agent architectures and agent-oriented programming. The DAI perspective to modeling systems in terms of entities that are capable of solving problems by means of reasoning through symbol manipulation resulted in various technologies that constitute the basic elements of agent systems. The early work on design of agent simulators within the DAI community focused on answering the question of how goals and intentions of agents emerge and how they lead to execution of actions that change the state of their environment. The agent-directed approach to simulating agent systems lies at the intersection of several disciplines: DAI, Control Theory, Complex Adaptive Systems (CAS), and Discrete-event Systems/Simulation. As shown in Fig. 16, these core disciplines gave direction to technology, languages, and possible applications, which then influenced the evolution of the synergy between simulation and agent systems.

Distributed Artificial Intelligence and Simulation

While progress in agent simulators and interpreters resulted in various agent architectures and their computational engines, the ability to coordinate agent ensembles was recognized early as a key challenge [65]. The MACE system [66] is considered as one of the major milestones in DAI. Specifically, the proposed DAI system integrated concepts from concurrent programming (e.g., actor formalism [67]) and knowledge representation to symbolically reason about skills and beliefs pertaining to modeling the environment. Task allocation and coordination were considered as fundamental challenges in early DAI systems. The contract net protocol developed by [68] provided the basis for modeling collaboration in simulation of distributed problem solving.

Agent Simulation Architectures

One of the first agent-oriented simulation languages, AGENT-0 [69], provided a framework that enabled the representation of beliefs and intentions of agents. Unlike object-oriented simulation languages such as SIM-



Artificial Intelligence in Modeling and Simulation, Figure 17
Agent-directed simulation framework

ULA 67 [70], the first object-oriented language for specifying discrete-event systems, AGENT-O and McCarthy's Elephant2000 language incorporated speech act theory to provide flexible communication mechanisms for agents. DAI and cognitive psychology influenced the development of cognitive agents such as those found in AGENT-0, e.g., the Belief-Desires-Intentions (BDI) framework [71]. However, procedural reasoning and control theory provided a basis for the design and implementation of reactive agents. Classical control theory enables the specification of a mathematical model that describes the interaction of a control system and its environment. The analogy between an agent and control system facilitated the formalization of agent interactions in terms of a formal specification of dynamic systems. The shortcomings of reactive agents (i.e., lack of mechanisms of goal-directed behavior) and cognitive agents (i.e., issues pertaining to computational tractability in deliberative reasoning) led to the development of hybrid architectures such as the RAP system [72].

Agents are often viewed as design metaphors in the development of models for simulation and gaming. Yet, this narrow view limits the potential of agents in improving various other dimensions of simulation. To this end, Fig. 17 presents a unified paradigm of Agent-Directed Simulation that consists of two categories as follows: (1) Simulation for Agents (agent simulation), i.e., simulation of systems that can be modeled by agents (in engineering, human and social dynamics, military applications etc.) and (2) Agents for Simulation that can be grouped under two groups: agent-supported simulation and agent-based simulation.

Agent Simulation

Agent simulation involves the use of agents as design metaphors in developing simulation models. Agent simulation involves the use of simulation conceptual frameworks (e.g., discrete-event, activity scanning) to simulate the behavioral dynamics of agent systems and incorporate autonomous agents that function in parallel to achieve their goals and objectives. Agents possess high-level interaction mechanisms independent of the problem being solved. Communication protocols and mechanisms for interaction via task allocation, coordination of actions, and conflict resolution at varying levels of sophistication are primary elements of agent simulations. Simulating agent systems requires understanding the basic principles, organizational mechanisms, and technologies underlying such systems.

Agent-Based Simulation

Agent-based simulation is the use of agent technology to monitor and generate model behavior. This is similar to the use of AI techniques for the generation of model behavior (e.g., qualitative simulation and knowledge-based simulation). Development of novel and advanced simulation methodologies such as multisimulation suggests the use of intelligent agents as simulator coordinators, where run-time decisions for model staging and updating takes place to facilitate dynamic composability. The perception feature of agents makes them pertinent for monitoring tasks. Also, agent-based simulation is useful for having complex experiments and deliberative knowledge processing such as planning, deciding, and reasoning. Agents are

also critical enablers to improve composability and interoperability of simulation models [73].

Agent-Supported Simulation

Agent-supported simulation deals with the use of agents as a support facility to enable computer assistance by enhancing cognitive capabilities in problem specification and solving. Hence, agent-supported simulation involves the use of intelligent agents to improve simulation and gaming infrastructures or environments. Agent-supported simulation is used for the following purposes:

- to provide computer assistance for front-end and/or back-end interface functions;
- to process elements of a simulation study symbolically (for example, for consistency checks and built-in reliability); and
- to provide cognitive abilities to the elements of a simulation study, such as learning or understanding abilities.

For instance, in simulations with defense applications, agents are often used as support facilities to

- see the battlefield,
- fuse, integrate and de-conflict the information presented by the decision-maker,
- generate alarms based on the recognition of specific patterns,
- Filter, sort, track, and prioritize the disseminated information, and
- generate contingency plans and courses of actions.

A significant requirement for the design and simulation of agent systems is the distributed knowledge that represents the mental model that characterizes each agent's beliefs about the environment, itself, and other agents. Endomorphic agent concepts provide a framework for addressing the difficult conceptual issues that arise in this domain.

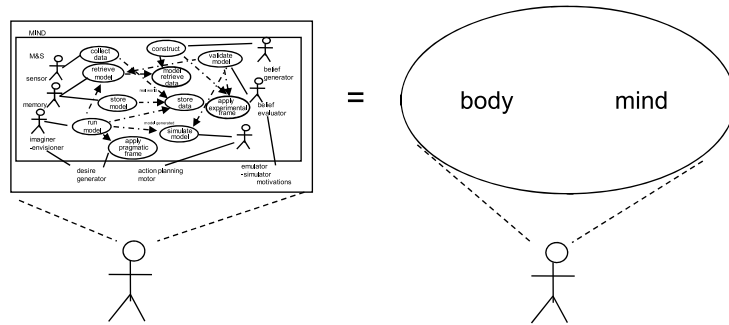
Endomorphic Agents

We now consider an advanced feature that an autonomous, integrated and comprehensive modeler/simulationist agent must have if it is fully emulate human capability. This is the ability, to a limited, but significant extent, to construct and employ models of its own mind as well of the minds of other agents. We use the term “mind” in the sense just discussed.

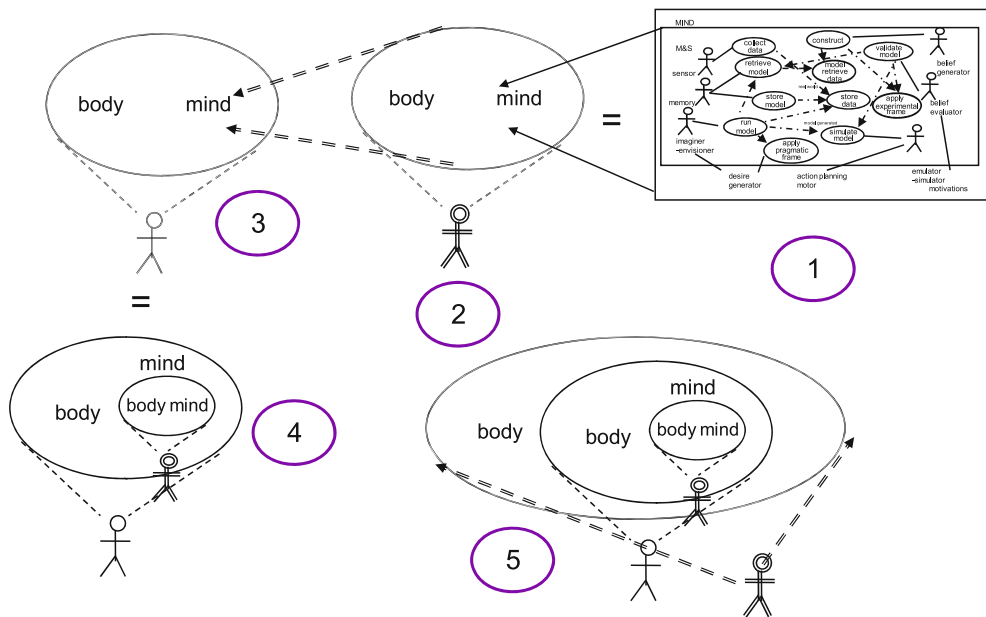
The concept of an endomorphic agent is illustrated in Fig. 18 and 19 in a sequence of related diagrams. The diagram labeled with an oval with embedded number 1 is that of Fig. 9 with the modifications mentioned earlier to

match up with human motivation and desire generation modules. In diagram 2, the label “mind” refers to the set of M&S capabilities depicted in Fig. 9. As in [30], an agent, human or technological, is considered to be composed of a mind and a body. Here, “body” represents the external manifestation of the agent, which is observable by other agents. Whereas, in contrast, mind is hidden from view and must be a construct, or model, of other agents. In other words, to use the language of evolutionary psychology, agents must develop a “theory of mind” about other agents from observation of their external behavior. An endomorphic agent is represented in diagram 3 with a mental model of the body and mind of the agent in diagram 2. This second agent is shown more explicitly in diagram 4, with a mental representation of the first agent's body and mind. Diagram 5 depicts the recursive aspect of endomorphism, where the (original) agent of diagram 2 has developed a model of the second agent's body and mind. But the latter model contains the just-mentioned model of the first agent's body and mind. This leads to a potentially infinite regress in which—apparently—each agent can have a representation of the other agent, and by reflection, of himself, that increases in depth of nesting without end. Hofstadter [59] represents a similar concept in the diagram on page 144 of his book, in which the comic character Sluggo is “dreaming of himself dreaming of himself dreaming of himself, without end.” He then uses the label on the Morton Salt box on page 145 to show how not all self reference involves infinite recursion. On the label, the girl carrying the salt box obscures its label with her arm, thereby shutting down the regress. Thus, the salt box has a representation of itself on its label but this representation is only partial.

Reference [30] related the termination in self-reference to the agent's objectives and requirements in constructing models of himself, other agents, and the environment. Briefly, the agent need only to go as deep as needed to get a reliable model of the other agents. The agent can stop at level 1 with a representation of the other's bodies. However, this might not allow predicting another's movements, particularly if the latter has a mind in control of these movements. This would force the first agent to include at least a crude model of the other agent's mind. In a competitive situation, having such a model might give the first agent an advantage and this might lead the second agent to likewise develop a predictive model of the first agent. With the second agent now seeming to become less predictable, the first agent might develop a model of the second agent's mind that restores lost predictive power. This would likely have to include a reflected representation of himself, although the impending regression could be



Artificial Intelligence in Modeling and Simulation, Figure 18
M&S within mind



Artificial Intelligence in Modeling and Simulation, Figure 19
Emergence of endomorphic agents

halted if this representation did not, itself, contain a model of the other agent. Thus, the depth to which competitive endomorphic agents have models of themselves and others might be the product of a co-evolutionary “mental arms race” in which an improvement in one side triggers a contingent improvement in the other—the improvement being an incremental refinement of the internal models by successively adding more levels of nesting.

Minsky [60] conjectured that termination of the potentially infinite regress in agent’s models of each other within a society of mind might be constrained by shear limitations on the ability to marshal the resources required to support the necessary computation. We can go further

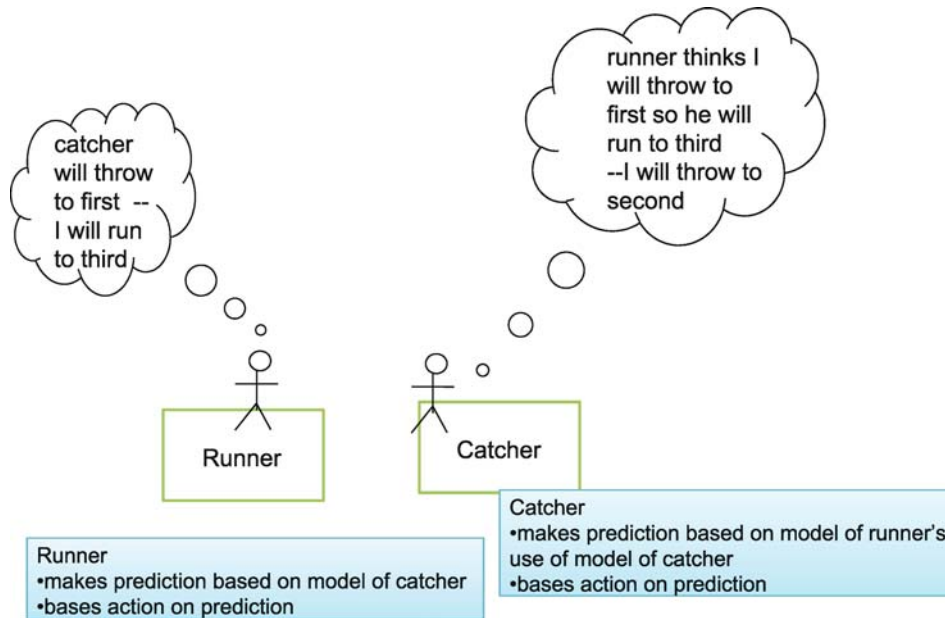
by assuming that agents have differing mental capacities to support such computational nesting. Therefore, an agent with greater capacity might be able to “out think” one of lesser capability. This is illustrated by the following real-life story drawn from a recent newspaper account of a critical play in a baseball game.

Interacting Models of Others in Competitive Sport

The following account is illustrated in Fig. 20.

A Ninth Inning to Forget

Cordero Can’t Close, Then Base-Running Gaffe Ends Nats’ Rally



Artificial Intelligence in Modeling and Simulation, Figure 20
Interacting models of others in baseball

Steve Yanda – Washington Post Staff Writer
Jun 24, 2007

Copyright The Washington Post Company Jun 24, 2007
Indians 4, Nationals 3

Nook Logan played out the ending of last night's game in his head as he stood on second base in the bottom of the ninth inning. The bases were loaded with one out and the Washington Nationals trailed by one run. Even if Felipe Lopez, the batter at the plate, grounded the ball, say, right back to Cleveland Indians closer Joe Borowski, the pitcher merely would throw home. Awaiting the toss would be catcher Kelly Shoppach, who would tag the plate and attempt to nail Lopez at first. By the time Shoppach's throw reached first baseman Victor Martinez, Logan figured he would be gliding across the plate with the tying run. Lopez did ground to Borowski, and the closer did fire the ball home. However, Shoppach elected to throw to third instead of first, catching Logan drifting too far off the bag for the final out in the Nationals' 4-3 loss at RFK Stadium. "I thought [Shoppach] was going to throw to first," Logan said. And if the catcher had, would Logan have scored all the way from second? "Easy."

We'll analyze this account to show how it throws light on the advantage rendered by having an endomorphic ca-

pability to process to a nesting depth exceeding that of an opponent.

The situation starts the bottom of the ninth inning with the Washington Nationals at bats having the bases loaded with one out and trailing by one run. The runner on second base, Nook Logan plays out the ending of the game in his head. This can be interpreted in terms of endomorphic models as follows. Logan makes a prediction using his models of the opposing pitcher and catcher, namely that the pitcher would throw home and the catcher would tag the plate and attempt to nail the batter at first. Logan then makes a prediction using a model of himself, namely, that he would be able to reach home plate while the pitcher's thrown ball was traveling to first base.

In actual play, the catcher threw the ball to third and caught Logan out. This is evidence that the catcher was able to play out the simulation to a greater depth than was Logan. The catcher's model of the situation agreed with that of Logan as it related to the other actors. The difference was that the catcher used a model of Logan that predicted that the latter (Logan) would predict the he (the catcher) would throw to first. Having this prediction, the catcher decided instead to throw the ball to the second baseman which resulted in putting Logan out. We note that the catcher's model was based on his model of Logan's model so it was at one level greater in depth of nesting than the latter. To have succeeded, Logan would have had to be

able to support one more level, namely, to have a model of the catcher that would predict that the catcher would use the model of himself (Logan) to out-think him and then make the counter move, not to start towards third base.

The enigma of such endomorphic agents provides extreme challenges to further research in AI and M&S. The formal and computational framework that the M&S framework discussed here provides may be of particular advantage to cognitive psychologists and philosophers interested in an active area of investigation in which the terms “theory of mind,” “simulation,” and “mind reading” are employed without much in the way of definition [61].

Future Directions

M&S presents some fundamental and very difficult problems whose solution may benefit from the concepts and techniques of AI. We have discussed some key problem areas including verification and validation, reuse and composability, and distributed simulation and systems of systems interoperability. We have also considered some areas of AI that have direct applicability to problems in M&S, such as Service-Oriented Architecture and Semantic Web, ontologies, constrained natural language, and genetic algorithms. In order to provide a unifying theme for the problem and solutions, we raised the question of whether all of M&S can be automated into an integrated autonomous artificial modeler/simulationist. We explored an approach to developing such an intelligent agent based on the System Entity Structure/Model Base framework, a hybrid methodology that combines elements of AI and M&S. We proposed a concrete methodology by which such an agent could engage in M&S based on activity tracking. There are numerous challenges to AI that implementing such a methodology in automated form presents. We closed with consideration of endomorphic modeling capability, an advanced feature that such an agent must have if it is to fully emulate human M&S capability. Since this capacity implies an infinite regress in which models contain models of themselves without end, it can only be had to a limited degree. However, it may offer critical insights into competitive co-evolutionary human or higher order primate behavior to launch more intensive research into model nesting depth. This is the degree to which an endomorphic agent can marshal mental resources needed to construct and employ models of its own “mind” as well as the “minds” of other agents. The enigma of such endomorphic agents provides extreme challenges to further research in AI and M&S, as well as related disciplines such as cognitive science and philosophy.

Bibliography

Primary Literature

1. Wymore AW (1993) Model-based systems engineering: An introduction to the mathematical theory of discrete systems and to the tricotyledon theory of system design. CRC, Boca Raton
2. Zeigler BP, Kim TG, Praehofer H (2000) Theory of modeling and simulation. Academic Press, New York
3. Ören TI, Zeigler BP (1979) Concepts for advanced simulation methodologies. *Simulation* 32(3):69–82
4. <http://en.wikipedia.org/wiki/DEVS> Accessed Aug 2008
5. Kneppell PL, Aragno DC (1993) Simulation validation: a confidence assessment methodology. IEEE Computer Society Press, Los Alamitos
6. Law AM, Kelton WD (1999) Simulation modeling and analysis, 3rd edn. McGraw-Hill, Columbus
7. Sargent RG (1994) Verification and validation of simulation models. In: Winter simulation conference. pp 77–84
8. Balci O (1998) Verification, validation, and testing. In: Winter simulation conference.
9. Davis KP, Anderson AR (2003) Improving the composability of department of defense models and simulations, RAND technical report. <http://www.rand.org/pubs/monographs/MG101/>. Accessed Nov 2007; J Def Model Simul Appl Methodol Technol 1(1):5–17
10. Yilmaz L, Ören TI (2004) A conceptual model for reusable simulations within a model-simulator-context framework. Conference on conceptual modeling and simulation. Conceptual Models Conference, Italy, 28–31 October, pp 28–31
11. Traore M, Muxy A (2004) Capturing the dual relationship between simulation models and their context. Simulation practice and theory. Elsevier
12. Page E, Oppen J (1999) Observations on the complexity of composable simulation. In: Proceedings of winter simulation conference, Orlando, pp 553–560
13. Kasputis S, Ng H (2000) Composable simulations. In: Proceedings of winter simulation conference, Orlando, pp 1577–1584
14. Sarjoughain HS (2006) Model composability. In: Perrone LF, Wieland FP, Liu J, Lawson BG, Nicol DM, Fujimoto RM (eds) Proceedings of the winter simulation conference, pp 104–158
15. DiMario MJ (2006) System of systems interoperability types and characteristics in joint command and control. In: Proceedings of the 2006 IEEE/SMC international conference on system of systems engineering, Los Angeles, April 2006
16. Sage AP, Cuppan CD (2001) On the systems engineering and management of systems of systems and federation of systems. Information knowledge systems management, vol 2, pp 325–345
17. Dahmann JS, Kuhl F, Weatherly R (1998) Standards for simulation: as simple as possible but not simpler the high level architecture for simulation. *Simulation* 71(6):378
18. Sarjoughian HS, Zeigler BP (2000) DEVS and HLA: Complementary paradigms for M&S? *Trans SCS* 4(17):187–197
19. Yilmaz L (2004) On the need for contextualized introspective simulation models to improve reuse and composability of defense simulations. *J Def Model Simul* 1(3):135–145
20. Tolk A, Muguira JA (2003) The levels of conceptual interoperability model (LCIM). In: Proceedings fall simulation interoperability workshop, <http://www.sisostds.org> Accessed Aug 2008
21. <http://en.wikipedia.org/wiki/Simula> Accessed Aug 2008

22. <http://en.wikipedia.org/wiki/Java> Accessed Aug 2008
23. http://en.wikipedia.org/wiki/Expert_system Accessed Aug 2008
24. http://en.wikipedia.org/wiki/Frame_language Accessed Aug 2008
25. http://en.wikipedia.org/wiki/Agent_based_model Accessed Aug 2008
26. http://www.swarm.org/wiki/Main_Page Accessed Aug 2008
27. Unified Modeling Language (UML) <http://www.omg.org/technology/documents/formal/uml.htm>
28. Object Modeling Group (OMG) <http://www.omg.org>
29. http://en.wikipedia.org/wiki/Semantic_web Accessed Aug 2008
30. Zeigler BP (1990) Object Oriented Simulation with Hierarchical, Modular Models: Intelligent Agents and Endomorphic Systems. Academic Press, Orlando
31. http://en.wikipedia.org/wiki/Service_oriented_architecture Accessed Aug 2008
32. Mittal S, Mak E, Nutaro JJ (2006) DEVS-Based dynamic modeling & simulation reconfiguration using enhanced DoDAF design process. Special issue on DoDAF. J Def Model Simul, Dec (3)4:239–267
33. Ziegler BP (1988) Simulation methodology/model manipulation. In: Encyclopedia of systems and controls. Pergamon Press, England
34. Alexiev V, Breu M, de Bruijn J, Fensel D, Lara R, Lausen H (2005) Information integration with ontologies. Wiley, New York
35. Kim L (2003) Official XMLSPY handbook. Wiley, Indianapolis
36. Zeigler BP, Hammonds P (2007) Modeling & simulation-based data engineering: introducing pragmatics into ontologies for net-centric information exchange. Academic Press, New York
37. Simard RJ, Zeigler BP, Couretas JN (1994) Verb phrase model specification via system entity structures. AI and Planning in high autonomy systems, 1994. Distributed interactive simulation environments. Proceedings of the Fifth Annual Conference, 7–9 Dec 1994, pp 192–1989
38. Checkland P (1999) Soft systems methodology in action. Wiley, London
39. Holland JH (1992) Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence. MIT Press, Cambridge
40. Goldberg DE (1989) Genetic algorithms in search, optimization and machine learning. Addison-Wesley Professional, Princeton
41. Davis L (1987) Genetic algorithms and simulated annealing. Morgan Kaufmann, San Francisco
42. Zbigniew M (1996) Genetic algorithms + data structures = evolution programs. Springer, Heidelberg
43. Cheon S (2007) Experimental frame structuring for automated model construction: application to simulated weather generation. Doct Diss, Dept of ECE, University of Arizona, Tucson
44. Zeigler BP (1984) Multifaceted modelling and discrete event simulation. Academic Press, London
45. Rozenblit JW, Hu J, Zeigler BP, Kim TG (1990) Knowledge-based design and simulation environment (KBDSE): foundational concepts and implementation. J Oper Res Soc 41(6):475–489
46. Kim TG, Lee C, Zeigler BP, Christensen ER (1990) System entity structuring and model base management. IEEE Trans Syst Man Cyber 20(5):1013–1024
47. Zeigler BP, Zhang G (1989) The system entity structure: knowledge representation for simulation modeling and design. In: Widman LA, Loparo KA, Nielsen N (eds) Artificial intelligence, simulation and modeling. Wiley, New York, pp 47–73
48. Luh C, Zeigler BP (1991) Model base management for multifaceted systems. ACM Trans Model Comp Sim 1(3):195–218
49. Couretas J (1998) System entity structure alternatives enumeration environment (SEAS). Doctoral Dissertation Dept of ECE, University of Arizona
50. Hyu C Park, Tag G Kim (1998) A relational algebraic framework for VHDL models management. Trans SCS 15(2):43–55
51. Chi SD, Lee J, Kim Y (1997) Using the SES/MB framework to analyze traffic flow. Trans SCS 14(4):211–221
52. Cho TH, Zeigler BP, Rozenblit JW (1996) A knowledge based simulation environment for hierarchical flexible manufacturing. IEEE Trans Syst Man Cyber- Part A: Syst Hum 26(1):81–91
53. Carruthers P (2006) Massively modular mind architecture the architecture of the mind. Oxford University Press, USA, pp 480
54. Wolpert L (2004) Six impossible things before breakfast: The evolutionary origin of belief, W.W. Norton London
55. Zeigler BP (2005) Discrete event abstraction: an emerging paradigm for modeling complex adaptive systems perspectives on adaptation, In: Booker L (ed) Natural and artificial systems, essays in honor of John Holland. Oxford University Press, Oxford
56. Nutaro J, Zeigler BP (2007) On the stability and performance of discrete event methods for simulating continuous systems. J Comput Phys 227(1):797–819
57. Muzy A, Nutaro JJ (2005) Algorithms for efficient implementation of the DEVS & DSDEVs abstract simulators. In: 1st Open International Conference on Modeling and Simulation (OICMS). Clermont-Ferrand, France, pp 273–279
58. Muzy A The activity paradigm for modeling and simulation of complex systems. (in process)
59. Hofstadter D (2007) I am a strange loop. Basic Books
60. Minsky M (1988) Society of mind. Simon & Schuster, Goldman
61. Alvin I (2006) Goldman simulating minds: the philosophy, psychology, and neuroscience of mindreading. Oxford University Press, USA
62. Denning PJ (2007) Computing is a natural science. Commun ACM 50(7):13–18
63. Luck M, McBurney P, Preist C (2003) Agent technology: enabling next generation computing a roadmap for agent based computing. Agentlink, Liverpool
64. Miller JH, Page SE (2007) Complex adaptive systems: an introduction to computational models of social life. Princeton University Press, Princeton
65. Ferber J (1999) Multi-Agent systems: an introduction to distributed artificial intelligence. Addison-Wesley, Princeton
66. Gasser L, Braganza C, Herman N (1987) Mace: a extensible testbed for distributed AI research. Distributed artificial intelligence – research notes in artificial intelligence, pp 119–152
67. Agha G, Hewitt C (1985) Concurrent programming using actors: exploiting large-scale parallelism. In: Proceedings of the foundations of software technology and theoretical computer science, Fifth Conference, pp 19–41
68. Smith RG (1980) The contract net protocol: high-level communication and control in a distributed problem solver. IEEE Trans Comput 29(12):1104–1113
69. Shoham Y (1993) Agent-oriented programming. Artif Intell 60(1):51–92
70. Dahl OJ, Nygaard K (1967) SIMULA67 Common base definition. Norwegian Computing Center, Norway

71. Rao AS, George MP (1995) BDI-agents: from theory to practice. In: Proceedings of the first intl. conference on multiagent systems, San Francisco
72. Firby RJ (1992) Building symbolic primitives with continuous control routines. In: Proceedings of the First Int Conf on AI Planning Systems. College Park, MD pp 62–29
73. Yilmaz L, Paspuetti S (2005) Toward a meta-level framework for agent-supported interoperation of defense simulations. *J Def Model Simul* 2(3):161–175

Books and Reviews

- Alexiev V, Breu M, de Bruijn J, Fensel D, Lara R, Lausen H (2005) Information integration with ontologies. Wiley, New York
- Alvin I (2006) Goldman Simulating Minds: The philosophy, psychology, and neuroscience of mindreading. Oxford University Press, USA
- Carruthers P (2006) Massively modular mind architecture The architecture of the mind. <http://www.amazon.com/exec/obidos/search-handle-url/102-1253221-6360121>
- John H (1992) Holland adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. The MIT Press Cambridge
- Zeigler BP (1990) Object oriented simulation with hierarchical, modular models: intelligent agents and endomorphic systems. Academic Press, Orlando
- Zeigler BP, Hammonds P (2007) Modeling & simulation-based data engineering: introducing pragmatics into ontologies for net-centric information exchange. Academic Press, New York
- Zeigler BP, Kim TG, Praehofer H (2000) Theory of modeling and simulation. Academic Press, New York

Astronomical Time Series, Complexity in

JEFFREY D. SCARGLE

Space Science Division, NASA Ames Research Center,
Moffett Field, USA

Article Outline

Glossary

Prolog: The Story of Goldilocks and the Three Time Series

Definition of the Subject

Introduction

Complex Time Series in Astronomy

Chaotic Dynamics in Celestial Mechanics

Chaotic Time Series Analysis in Astronomy

Future Directions

Bibliography

Glossary

Dynamical system A mathematical or physical system of which the evolution forward in time is determined by

certain laws of motion – deterministic, random, or a combination.

Deterministic system A dynamical system the future of which, except for noise, can be exactly predicted based on past data.

Chaotic system A dynamical system the future of which can be predicted for short times, but due to the exponential growth of noise (“sensitive dependence on initial conditions”) long term prediction is not possible.

Random system A dynamical system the future of which cannot be predicted, based on past data.

Linear system Consider a dynamical system described in part by two variables (representing forces, displacements, or the like), one of which is viewed as *input* and the other as *output*. The part of the system so described is said to be *linear* if the output is a linear function of the input. Examples: (a) a spring is linear in a small force applied to the spring, if the displacement is small; (b) the Fourier transform (output) of a time series (input) is linear. Example (b) is connected with the myth that a linear transform is not useful for “linear data.” (To describe given time series data as linear is an abuse of the term. The concept of linearity cannot apply to data, but rather to a model or model class describing the data. In general given data can be equally well described by linear and nonlinear models.)

Phase portrait A picture of the evolution of a dynamical system, in a geometrical space related to the phase space of the system. This picture consists of a sampling of points tracing out the system trajectory over time.

Stationary system If the statistical descriptors of a dynamical system do not change with time, the system is said to be stationary. In practice, with finite data streams, the mathematical definitions of the various kinds of stationarity cannot be implemented, and one is forced to consider local or approximate stationarity.

Prolog: The Story of Goldilocks and the Three Time Series

Once upon a time there was a little girl named Goldilocks. She went for a walk in the forest. Pretty soon she came upon an observatory. She knocked and, when no one answered, she walked right in.

On the server in the computer room, there were three data sets. Goldilocks was hungry to find magical, simple dynamics underlying an apparently complicated time series. She tested the time series from the first data set.

“This time series is too noisy!” she exclaimed.

So, she tested the time series from the second data set.

“This time series is too short,” she said

So, she analyzed the last time series.

“Ahhh, this time series is just right,” she said happily and she analyzed it. “Wow, look, I found a ‘chaotic attractor’ and a small, non-integer embedding dimension, so this must be a chaotic astronomical system!” Goldilocks began to speculate about the underlying physics, but there were many detailed models possible. Goldilocks fell asleep. When she awoke later, she realized that this had all been a fairy-tale dream.

THE END

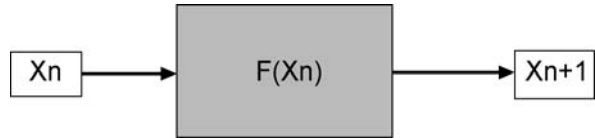
Definition of the Subject

The subject of this article is *time series analysis as a tool for analyzing complex astrophysical systems*. A time series is a set of sequential measurements where information about the measurement times is contained in the data record. It is implicit that the goal is to study the variation of the measured quantity, ruling out measurements repeated simply to more accurately determine a value, say be averaging.

A number of developments are combining to markedly increase the importance of this topic in modern astrophysics. The first is the growing realization that many astronomical objects are dynamically changing in complicated ways, yielding intensity changes that have an unpredictable, stochastic nature. In short, many of the objects previously thought to be evolving only slowly and serenely, and on very long time scales, are in actuality complex systems – if not chaotic in a rigorous sense, at least behaving stochastically. This development has obviously been fueled by the recent rapid progress in complex systems theory. Less obvious is the role of improved observational techniques. It is now much easier to distinguish randomness in time series data due to observational errors from intrinsic – regular or stochastic – variability in the source itself. And in this Age of Digital Astronomy analysis of large sets of high-quality time series is more and more crucial to progress.

Introduction

Let’s begin with a brief description of *dynamical systems*, in both the mathematical and physical sense. The former is an algorithm put forth as a model of the latter. *Dynamics* refers to the fact that the algorithm advances an initial state



Astronomical Time Series, Complexity in, Figure 1
Black box representation of a discrete, one-dimensional dynamical system

in time, modeling physical forces that drive the temporal behavior of an astrophysical system.

Now consider some simple dynamical systems that exhibit complex behavior of the sort that has been called *chaos*. Figure 1 depicts the essence of a simple, one dimensional dynamical system, namely a procedure for generating the value of the variable X at the next future time $n + 1$ (the output), given the value of X at the current time n (the input). The procedure starts at time zero, with initial condition X_0 as input, and is repeated indefinitely, generating the time series $\{X_0, X_1, X_2, X_3, \dots\}$. Algebraically, the above picture is represented as simply

$$X_{n+1} = f(X_n), \quad (1)$$

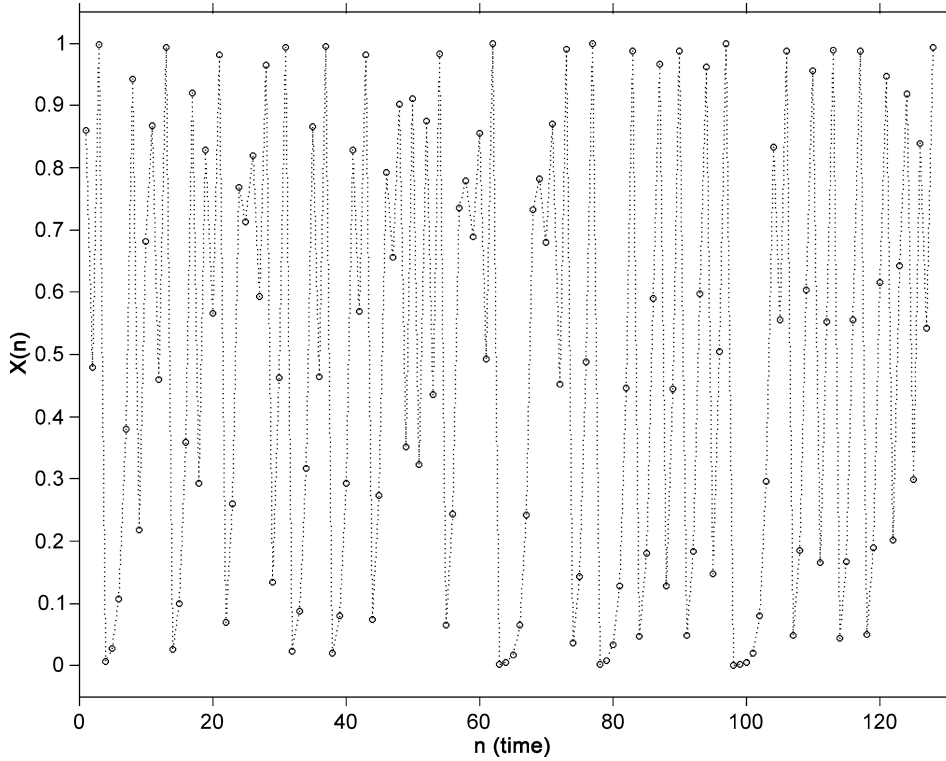
where f is the function representing the dynamics of the process. The following figure plots an example of such a dynamical system for a specific function f . The values of X look rather random, although the behavior for small values of X are a hint that some non-randomness is lurking here.

A commonly used test of whiteness is to compute the autocorrelation function and/or the power spectrum of X . Here are the results for the same process as above:

Note that the (symmetric) auto-correlation function is essentially a delta function, significantly different from zero only at zero lag. The power spectrum, of course, is irregular, but it is consistent with being constant with frequency. So our process appears to be white noise. Indeed, the autocorrelation and power spectrum of this process are a delta function and constant, respectively. The process X is indeed white noise!

But you have just been led down a garden path, because the interpretation of this as being a random process is wrong. Equation (1) precisely defines deterministic dynamics. The next future value is a definite, single-valued function of the past value. So our white process can hardly be truly random noise.

It is time to reveal X , as well as the fact that there is a stronger form of randomness than described by the term *uncorrelated*. The above figures all refer to the famous *lo-*



Astronomical Time Series, Complexity in, Figure 2

Time series generated by a simple nonlinear dynamical system. There are $N = 128$ time samples plotted

gistic map, namely

$$X_{n+1} = f(X_n) = \lambda X_n(1 - X_n); \quad (2)$$

in what follows $\lambda = 4$, a special value that makes X maximally chaotic. The interested reader should consult one of the many books discussing the extraordinarily complex behavior of this nonlinear system as a function of λ , including proof that it is a “white” process, i. e. has a constant power spectrum and a delta-function autocorrelation, for $\lambda = 4$.

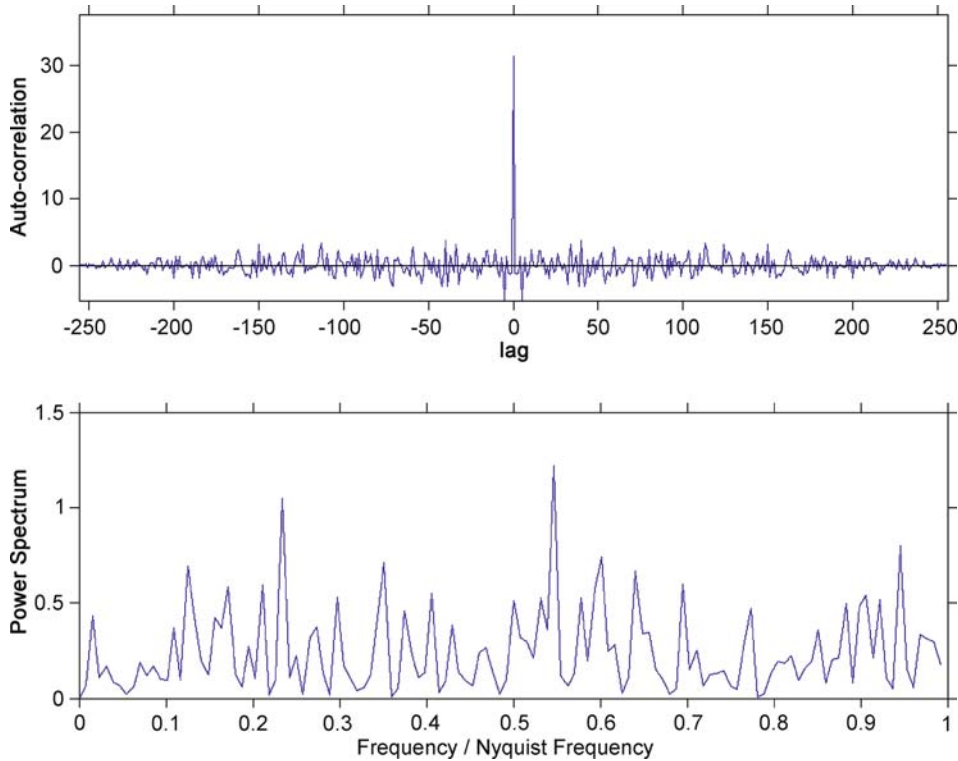
The point here is that the simple first-order diagnostics – autocorrelations and power spectra – as useful as they are, are not helpful at identifying the nature of our process. Let’s turn to basics. The essence of a random process is in its joint probability distributions, not in its correlations. The joint distribution of X_n and X_m is defined as the probability that the two variables take arbitrary values, thusly:

$$P(a, b) = \text{Prob}(X_n = a \text{ and } X_m = b), \quad (3)$$

where the notation here emphasizes that the arguments of P are dummy variables. The more common, if slightly misleading, notation is $P(X_n, X_m)$. One can estimate this quantity with 2D histograms – i. e., simply count pairs of values in 2D bins. Figure 4 is such an estimate of $P(X_n, X_{n+1})$. If the process X were truly random, all of the variables X_n would be statistically independent of each other. Independence is defined by the relation

$$\begin{aligned} \text{Prob}(X_n = a \text{ and } X_m = b) \\ = \text{Prob}(X_n = a) \text{ Prob}(X_m = b). \end{aligned} \quad (4)$$

In words, two variables are independent if their joint probability distribution is the product of their individual distributions. It should be clear from the figure that the logistic process is very far from independent; on the contrary, the value of X_{n+1} is determined completely by the value of X_n . Unfortunately, the concepts of uncorrelatedness and independence are frequently confused. In addition, scientists often derive results using the correlation function when they could derive much more powerful results using properties of joint probability distributions.



Astronomical Time Series, Complexity in, Figure 3

Autocorrelation and power spectrum from the same process as in Fig. 2 (but with twice as many time points, $N = 256$)

Complex Time Series in Astronomy

The first time series data in recorded history were probably astronomical in origin. It was important for the ancients to keep track of the motions of celestial objects, if only for predicting the arrival of each year's planting season. This agricultural service involved noting the passage of time and the location of the sun relative to other objects in the sky. Well-known examples of this connection between astronomical observations over time and practical aspects of agriculture punctuate the history of the ancient Egyptian the Mayan culture. There are presumably many other examples that have not survived the ravages of time – or the ravages of “all the wrong explorers and discoverers – thieves planting flags, murderers carrying crosses whom we have been raised to honor” (Peter S. Beagle, preface to J. R. R. Tolkien's “The Hobbit”).

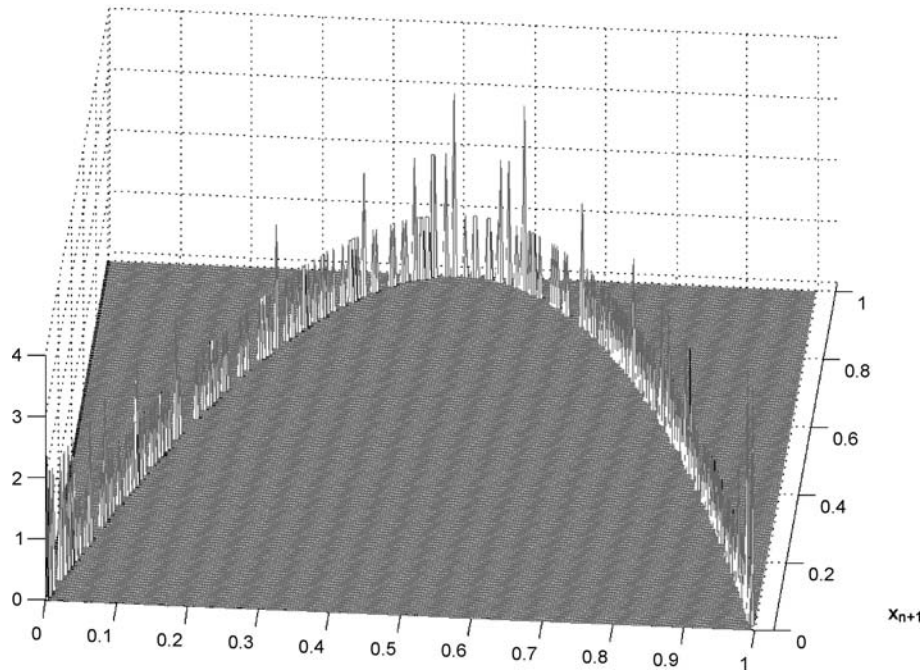
Perhaps the most famous historical incident of relevance is Gauss's successful determination in 1801 of the orbit of Ceres from a mere 41 days of data, over which the asteroid moved only 9 degrees on the sky. Gauss had invented the method of least squares some years earlier, and it was relatively easy (only 100 hours of computation

time!) to apply this method to the problem of determining an orbit of an object moving around the sun, from a time series of its positions on the sky.

“... for it is now clearly shown that the orbit of a heavenly body may be determined quite nearly from good observations embracing only a few days; and this without any hypothetical assumption.”

Carl Friedrich Gauss

From the dawn of recorded astronomical observations, as perhaps implemented in prehistoric sites such as Stonehenge, to the Space Age, astronomical time series have been central to scientific exploration. The underlying goal is to be able to understand the dynamical forces driving the underlying physical processes and evidenced by variability in the observed time series. Starting with Poincaré in the later part of the 19th century, whose work was largely in the context of planetary motions, astronomers faced the peculiar possibility that simple, deterministic astrophysical systems could have complex, apparently random behavior. For example, celestial mechanics realized that some of the planetary motions they had been dealing with



Astronomical Time Series, Complexity in, Figure 4

Histogram estimate of the joint probability distribution of X_n and X_{n+1} using data from the logistic process described in the previous figures

over the ages are in reality examples of chaos – the evolution of physical systems in time in a deterministic yet disordered way, often summarized as “simple systems with complex behavior.”

How can a simple and perfectly deterministic physical system produce complex and unpredictable behavior? The answer lies in a kind of instability – sensitivity to initial conditions. A nonlinear, chaotic system has this property in the sense that evolutionary paths starting from two arbitrarily close initial conditions separate exponentially with time. This property leads to an amplification of the inevitable initial uncertainties that forecloses prediction of the future. In the hope that complex, seemingly random behaviors observed in some systems could have simple explanations, there was in previous decades a spasm of activity in astrophysics, primarily in two complementary areas: (a) theoretical computations of the nonlinear dynamical models of astrophysical systems, and (b) analysis of time series and other data from astrophysical systems, in an effort to uncover the simplicity underlying the apparent complexity. The first of these is treated in the article Chaos and Complexity in Astrophysics, by Oded Regev, elsewhere in this Encyclopedia and rather briefly in the next section, while the second – being the current focus – will be treated in more detail in the rest of this article.

Chaotic Dynamics in Celestial Mechanics

The period around 1980–1990 saw intensive development in dynamical systems theory, particularly in celestial mechanics – i.e. the theoretical study of the dynamics and kinematics of astronomical bodies, in our solar system, in our galaxy, and elsewhere in the Universe. Lecar, Franklin, Holman and Murray [40] review the topic of chaotic dynamics in the solar system.

As in the history of earlier developments in dynamics (Galileo, Newton, Einstein, to name a few of the great minds illuminating this grand passage in human knowledge), astronomy played an important role in the origin of complex systems theory. Jack Lissauer [41] provides a comprehensive review of nonlinear dynamics in the solar system, covering the extensive work of Wisdom, Laskar, Duncan, and many other workers in this arena. The thread here is that in many important situations the equations of motion defining the dynamical system are known exactly, and the driving forces are known with great precision, so that highly accurate numerical computations of the dynamical evolution of the systems are achievable. These computations, and accompanying analytic studies, have detailed a number of key examples of sensitivity to initial conditions and quasi-random wandering of state space

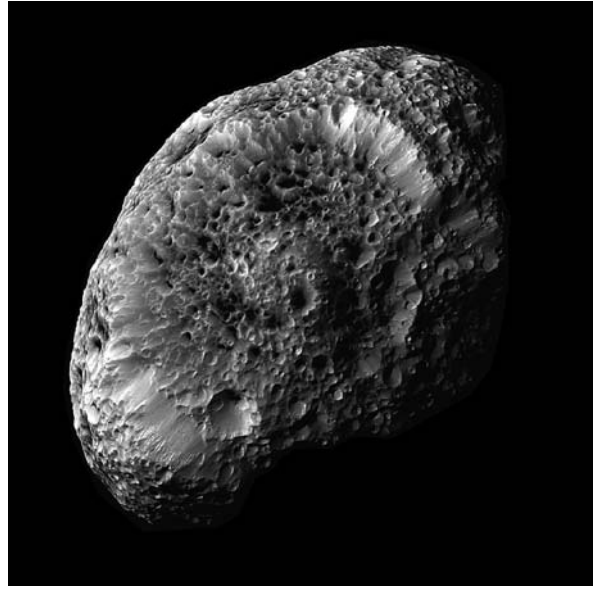
that are the hallmarks of deterministic chaos. One of the earliest of these success stories was the seminal Ph. D. thesis of Jack Wisdom [76], which elucidated the structure of gaps known in the spatial distribution of asteroids by showing that the missing regions correspond to chaotic orbital behavior.

For the most part, these computations relate to the very long timescales of many astronomical phenomena. One exception – the astronomical system showing chaotic behavior on the shortest time scale – is the fascinating story of the rotation of Hyperion, an irregularly shaped satellite of the planet Saturn. This satellite is unusual in several ways: its shape is very irregular, as can be seen in the dramatic false-color image in Fig. 5, its orbit is eccentric (eccentricity $e = 0.1$), and it is relatively close to its mother planet (the semi-major axis of its orbit is less than 25 times the radius of Saturn). These features of Hyperion and its orbit conspire to produce strong and regularly varying tidal forces on the satellite. Specifically, the differential equation describing the time evolution of the angle θ between the long axis of the satellite and a reference direction is

$$d^2\theta/dt^2 + (\omega_0^2/2r^3) \sin 2(\theta - f) = 0, \quad (5)$$

where f is an angle (the true anomaly) specifying the position of the satellite in its orbit, r is the radial distance of the satellite from the planet, and ω_0 is a function of the principle moments of inertia of the satellite [42]. This nonlinear equation has very interesting properties, and several authors (e. g. Wisdom, Peale and Mignard [77]) have shown that this system undergoes chaotic motion. The surface of section, or state space comprised by θ and $d\theta/dt$, consists of large regular regions, within which are chaotic islands roughly associated with the spin orbit resonances 1:2, 1:1, 3:2, 2:1, 5:2, etc. This configuration has marked similarities with the three-body system, although it is of course much simpler.

Detailed numerical computations of the Hyperion-Saturn system predicted chaotic rotation of the satellite. That is to say, the satellite's rotation axis should tumble chaotically and not remain even roughly pointed in a fixed direction. In a real sense, the magnitude and direction of rotation should not even have well defined values in the short term; further, the meaning of the long-term averages would be problematic. This peculiar state of motion, unprecedented among the other satellites of the Solar System, was consistent with a number of close-up images of Hyperion obtained by the Voyager spacecraft, and more recently by the Cassini spacecraft (Fig. 5). In addition, it was noted that a consequence of the irregular rotation is that the brightness of Hyperion also be chaotic, be-



Astronomical Time Series, Complexity in, Figure 5

This image of Hyperion shows strange details across the strange, tumbling moon's surface. Differences in color could indicate differences in the surface composition. This image was taken by the NASA Cassini spacecraft during a close flyby on Sept. 26, 2005

cause the shape and reflectivity of the surface is irregular. Klavetter [36] made time series measurements of Hyperion's brightness to study this question. The observations were difficult, due to the faintness of the satellite and the background of light scattered in the telescope by the very bright planet Saturn. In addition, he attempted to elucidate the state of chaotic motion by finding parameter values for a model of the rotational dynamics, by fitting the model to the time series data [36].

The chaotic rotation of Hyperion is the best case of the connection between dynamical systems theory and time series analysis, and the only case where the time scales are short enough for detailed observational studies. The reader is referred to the excellent article, *Chaos and Complexity in Astrophysics*, by Oded Regev, in this Encyclopedia, for more details on the relevant theoretical background of both this and other complex astronomical systems.

Chaotic Time Series Analysis in Astronomy

We now turn to our main topic, the extraction of information about complex astronomical systems by analyzing observational time series data. Astronomical time series have much in common with sequential data in other sciences, but also have features of special concern. Astronomy is fundamentally an observational science. Direct ex-

periments are for the most part not feasible, and in some cases one cannot even obtain repeated observations. Consider the distribution of galaxies, cosmic background radiation, etc. on the sky. Short of sending an astronomical observatory to another part of the Universe, there is only one view of the sky. As a result, it can be very difficult to judge whether certain peculiar structures are accidental (e. g. statistical fluctuations connected with observational errors) or physically significant. Indeed, concepts such as *randomness*, *determinism*, and *probability* need to be interpreted in somewhat special ways in cosmology and the study of large-scale structure of the Universe. (For an example, see the discussion of the term *cosmic variance*, e. g., on Wikipedia.)

More germane to this discussion, many observational time series cannot be obtained at evenly spaced time intervals, as is assumed in the application of most standard analysis methods. Gaps or other sampling irregularities can interfere with the ability to characterize complex variations. This is especially true of low signal-to-noise observations with random sample times – for then, the complexities of observational errors, sampling, and true system behavior are confusingly tangled together. Even worse, the sampling can be partially random, partially semi-regular, dependent on previous values or estimated signal components, or even dependent on the real-time perceived importance of the observations to the public or to a funding agency.

Another consideration, in many but by no means all cases, is simply the quantity and quality of the data. If each data point requires dedicating a large amount observing time with an expensive telescope, long time series will not be common. Too, the object of interest may be faint and affected by instrumental and cosmic backgrounds. The analyst may thus have to be content with short, noisy time series. Early in the history of this subject most of the data analysis methods were developed in the context of laboratory or computer studies, for which large amounts of data were readily available and could be repeated (with different initial conditions or parameter values) at will. As will be seen, diagnosing complex or chaotic behavior fundamentally requires access to data on the system's return to states very close to each other, but at wide range of times. Hence the difficulties described here are particularly limiting. Accordingly many standard methods were ineffective on most astronomical time series data.

Now consider the setting in which most time series analysis of complex systems is carried out, the *embedding* procedure, with which it is possible to derive dynamical information from observations of a subset of the variables needed to specify the full dynamical behavior of the sys-

tem, or even from variables that are not even dynamical in nature. This remarkable mathematical result means that observations of subsidiary quantities that are not directly involved in the dynamics can provide clues to underlying chaotic dynamical processes. For example, observations of brightness variations in stars can yield understanding of the dynamics of the stellar interior. An unfortunate feature of this mathematics is that the lack of direct connection between observable and the dynamics leaves room for data analytic results of questionable physical significance. The interested reader should supplement the intuitive, non-mathematical discussion here, with reference to one of the many excellent books on this topic, such as Sprott [67].

A primary question facing the analyst is whether the noisy data at hand were generated by a (simple) deterministic, chaotic, or random process – or some combination. The deterministic nature of chaos means that knowledge of the dynamical laws of the system and the current state, as long as they are both known accurately enough, allow accurate prediction of future states. Of course sensitive dependence on initial data prevents this kind of prediction from extending very far into the future. In contrast, if the physical process is random, there is an intrinsic uncertainty of future values even a short time into the future – perhaps due to quantum mechanical effects or the presence of complex, unmeasured hidden variables. One of the methodologies of time series analysis is to use features relating to these kinds of predictability (or lack thereof) to diagnose the nature of the underlying physical system. For practical purposes, then, short-term predictability is the key: chaotic processes may be predicted accurately over short periods of time, and random ones cannot.

It is important to note that even random noise can be predicted to some extent. A trivial example is that even white noise can be predicted by guessing that the next value will be the mean value, either as determined externally or from past observations. More germane is the fact that nonwhite noise can be predicted to an even greater degree. Correlations present in such a process can yield a prediction that is even better than the mean value. This partial predictability needs to be distinguished from the more accurate, but short term, predictability of a chaotic process.

At this point the reader will appreciate the difficulties of detecting or characterizing the various kinds of complex behavior from time series analysis. The remainder of this section will describe various properties of time series data that are useful in diagnosing chaotic dynamics and other complex behavior. Some are merely consistent with chaotic dynamics and do not really prove that chaos is present. Others are properties that of necessity must be present if the underlying process is chaotic.

An obvious first question is: To what extent the visual appearance of a plot of the time series indicate the presence of chaos? Rather easily one can visually distinguish stochastic processes from linearly deterministic ones, such as periodic oscillations or polynomial trends. (Here the term *stochastic* refers to the quality of disorder that is common to random and chaotic processes.) But the distinction between chaos and randomness is more subtle. As is evident in Fig. 1, plots of chaotic and random data with the same degree of correlation (e. g. white noise against the output of the logistic map for $\lambda = 4$) are nearly indistinguishable. The reader is encouraged to make such plots using Eq. (2) for white chaos and a random number generator for white noise.

The eye seems good at distinguishing order from disorder. Why do we have so much difficulty visualizing the difference between chaos and randomness? The problem lies not with the abilities of the human eye but largely with the way the data are presented. The distinction between various kinds of disorder, or stochastic behavior, reside in a different space than the standard time series plot, as will soon be evident.

Part of the problem with visual recognition is that the dependence of successive values of the observable X is not revealed very well by plotting them, next to each other on a time axis. This standard time series plot places too much emphasis on the time at which the measurements were made. What matters are the relationships between observations at successive times, not the absolute times of the observations. This suggests plotting X_n against X_{n+1} , as opposed to X_n against n . Compare Figs. 2 and 4.

This simple idea has been around in many guises for a long time, and probably cannot be credited to any one person. A charming introduction to the resulting plot, sometimes called a *return map* was given by Shaw [66] – with the slight twist that he plotted successive time intervals between discrete events (the falling of a water drop from a drippy faucet) against each other, rather than the values of an observable. This simple idea has grown into the most powerful tool of chaotic time series analysis. To see why this is so, we make what seems like a diversion – but we'll return soon.

This diversion is *state space*. It is generally useful to study a physical system in this abstract space, the coordinates of which are a complete set of independent dynamical variables. The use of this space lies in the fact that any configuration of the system corresponds to a unique point in this space, and vice versa. The system's complete evolution similarly corresponds to a trajectory or *orbit* in state space. This surprisingly useful geometrical view is old (cf. Poincare, whose studies of the three-body problem of

celestial mechanics, in the late 19th century, are considered by some to be the origin of the idea of chaos, and certainly of the geometric view of dynamics) and well described in the modern literature of dynamical systems, e. g. Wiggins [75].

Unfortunately, it is rare that an experimentalist can measure all the physical coordinates necessary to specify the state of the system. Practical considerations often permit measurement of only a few state variables or, even worse, only peripheral quantities that depend indirectly on the state. An example of this relation is the case of the chaotic tumbling of Saturn's satellite Hyperion [36], discussed above, where the overall brightness of the satellite must serve as a surrogate for the set of variables needed to specify the rotational state (rotation direction and magnitude, plus orbital parameters). The passive nature of astronomical observations exacerbates this disconnect between observables and dynamically relevant variables. A chemist or physicist can insert a probe and measure things directly, whereas the astronomer must accept those photons, particles, or perturbations of space-time that Nature chooses to send our way.

Remarkably, extensive measurements of just one variable can reveal the structure of the orbits in the full multivariate state space. Packard, Crutchfield, Farmer, and Shaw [51] and Takens [70] show that under certain conditions on the dynamical system there is a multidimensional *embedding space*, the coordinates of which can be derived from a single observed variable. The time series data generate trajectories in this space that are simply related to the system's state-space orbits. This relation is in the form of a smooth map, from the state space to the embedding space, that preserves the topology of the evolutionary paths. Thus the essential features of the unobservable state-space orbits can be understood by studying the accessible trajectories in the embedding space. In this analysis, one does not necessarily need to know the dimension of the state space – or even the identity of its coordinates.

Of the many choices of embedding space coordinates the most common is the observed variable evaluated at a set of lagged times:

$$X = (X_n, X_{n+k}, X_{n+2k}, X_{n+3k}, \dots, X_{n+(M-1)k}) \quad (6)$$

The lag k and the embedding dimension M are positive integers. In principle the lags need not be equal, but in practice they are almost always chosen to be so. As time goes by, points defined by Eq. (6) fall on embedding-space trajectories topologically equivalent to the system's state-space orbits. The trajectories can be traced out using time series data, albeit crudely if the noise level is large. When the system satisfies the requisite mathematical conditions

(and M is high enough) one says that one has constructed a *suitable embedding space*.

We thus arrive at a most important tool for the analysis of chaos, the *phase portrait*, also called the state-space portrait. The phase portrait plays a key role in the search for order in chaos. It can be constructed directly from the time series data. In simplest form it is just a plot of X_{n+1} against X_n . In other cases the process must be described in a space of higher dimensions, and we add coordinates X_{n+2}, X_{n+3}, \dots . Also, the lag k need not be one unit. A caution: the example described by Harding, Shimbrot and Cordes [32] shows that one can be fooled by the appearance of the phase portrait. They exhibit purely random, non-chaotic models for their pulsar data which produce phase portraits that show the interesting structure that is sometimes thought to uniquely characterize chaos.

In practice, with real data, there are further complications. Observational errors always produce noise that somewhat blurs the phase portrait. Yet with a sufficiently large signal-to-noise ratio this plot will reveal the nature of the dynamics in spite of noise. One does not know a priori the value of the lag. In the theorems justifying the embedding procedure – namely infinite, noise-free data streams – the value of k does not matter. In practice k does matter, and one must figure out a good value from the data. One does not know a priori the value of the dimension M of the embedding space. In theory M must be larger than $1 + 2d$ (d is the dimension of the physical state space), but in practice considerably smaller values provide suitable embeddings. However, it may well be that the correct dimension for viewing the underlying process is higher than 3, in which case there are problems displaying and visualizing the data.

The *attractor* of a physical system is, simply put, the part of state space toward which the system tends to evolve. The attractor ranges from simple closed curves in the case of periodic motion to highly complex, high-dimensional fractal sets for chaotic motion [2].

Another view of the embedding procedure is that it is the experimentalist's view of the attractor of the system. That is, the smooth transformation that maps state space into the embedding space maps the physical attractor into a set contained in the embedding space. The experimentalist cannot deal with the attractor itself, and must learn about it by studying its image in the embedding space. An important point about the phase portrait is sometimes overlooked: **the time sequence with which the sample points traverse the embedding space is important**. In simple terms, if the points smoothly trace out a figure, no matter how complex it is, the process is indicated to be

regular. On the other hand, if the points jump randomly, from one part of the embedding space to another, and back again, the process is indicated to be chaotic. While this behavior can be depicted with an animated display on a computer screen, and captured in a video file, it is much harder to display on a piece of paper or in a publication.

The dimension of the attractor (which clearly cannot be larger than the embedding dimension) is an important quantity. The relation between the attractor dimension, the embedding dimension, the dimension of the embedding-space image of the attractor, and a host of other dimension quantities that have been introduced is a complex subject. See [23,70] for the pioneering work, and textbooks such as [67] for up-to-date discussions.

In turn, estimation of the dimension of the attractor from the time series data is a complex topic. There are many different kinds of dimensions that can be defined, most of which can be estimated from data. There are many different techniques for estimating a given dimension. And a single number does not always characterize fully the dimension of an attractor. Generalized dimensions and a function which represents the distribution of dimensions have been introduced (e.g., Mandelbrot [43,44], Theiler [72]).

The literature on this subject is too large to review here. An overview [72] contains 143 references. The classic paper is by the Santa Cruz group [28]; see also [23] for a very nice physical introduction. The book [45] contains many articles on dimension and dimension estimation. The statistics of dimension estimates (e.g. Holzfuss and Meyer-Kress [34], Theiler [72], Smith 1990, Ramsey and Yuan [54]) is of interest to experimentalists, for they need to assess the reliability of their results. Unfortunately the uncertainties due to other effects [14], some of which we will soon meet, may often overwhelm purely statistical uncertainties.

The technique most commonly used by experimentalists provides an estimate of the correlation dimension [30,31]. This method implements the notion that the rate at which the volume contained in a hyper sphere embedded in M dimensional space grows with radius as R^M .

Unfortunately there are severe problems with the use of dimension estimates to diagnose chaotic dynamics. It is fair to say that the technique is essentially useless for detecting the presence of chaos in most astronomical applications. To start with, large amounts of data are necessary to determine dimensions with any reliability. Next, even modest noise can affect the results [49]. Smoothing of the data, due to either experimental effects (e.g., finite-time-constant instrumentation) or physical effects (e.g., radiative transfer effects in the case of astronom-

ical objects of variable luminosity) distorts the dimension [6,7,15,47,52,59,60,61,62].

Furthermore, there are more fundamental difficulties with extracting useful dimensional information from time series. Contrary to assumptions frequently made by early workers in this field, determination of a low dimension, fractional or otherwise, does not prove the existence of a chaotic process underlying the time series. In a seminal paper, Osborne and Provenzale [50] showed that purely non-chaotic colored noise has a low dimension as determined with the Grassberger–Procaccia and similar algorithms. They showed that the dimension is a simple function of the steepness (quantified by a power-law index) of the power spectrum. Accordingly, essentially any dimension can be produced by a purely random (no chaos need apply!) process, with the appropriate power spectrum. But see [53] for some methods to address the problem of distinguishing chaos from randomness, and [74] for a broad overview of the problems encountered with astronomical time series.

A second fundamental issue was raised by Ruelle and Eckmann [58], who showed that there is a simple upper limit to the value that can be obtained using the Grassberger–Procaccia or any other approach to dimension estimation where one regresses counts of points in boxes (hyper cubes) in the embedding space against the box volume. By noting the limiting cases of one point per box and all N points in the box, one sees that the maximum logarithmic slope (which is the dimension estimate) of the regression curve is $\log(N)$, approximately. This is an order of magnitude estimate, and it hardly matters whether the base of the logarithm is e or 10.

Hence if the true dimension is any value greater than the logarithm of the number of data points, one will estimate a dimension $D \sim \log(N)$. Unfortunately the astronomical literature is littered with dimension estimates in the range $D \sim 2$ –3 or so, based on 100 to 1000 or so data points, accompanied by breathless claims of detection of chaos. See also [57] for other comments aimed at dampening some of the enthusiasm in this arena.

In summary, for a number of reasons, estimates of the “fractal dimension” of time series data is nearly useless as a litmus test for nonlinear deterministic dynamics, or chaos.

Period doubling and bifurcations are sometimes associated with chaotic dynamics. The idea is as follows: suppose a dynamical system has periodic behavior, and the value of the period changes as some parameter of the system changes. An example is the logistic process of Eq. (2), for some values of λ less than 4. It is well known that this exceedingly simple system has exceedingly complex behav-

ior, including critical parameter values when one period becomes two. Although some variable stars and other periodic astronomical systems exhibit this kind of behavior, as a function of time and possibly related to underlying changes in physical parameters, these phenomena are not very common in astronomy, and therefore will not be discussed here. In any case, much of the theoretical research in this area focuses on control – that is, where the parameter or parameters can be adjusted more or less arbitrarily. Once again the passive, observational nature of astronomy comes in to play. Nevertheless, there is at least one feature of time series data that has been taken as indicative of possible period doubling, namely systematic amplitude changes of an otherwise periodic process. For example, in the light curve of a variable star, “the alternation of deep and shallow minima” is some indication of period doubling [13]. Clearly such a conclusion is speculative, and would need to be supported by further observations, e.g., of the light curve as conditions in the stellar interior or atmosphere undergo secular changes.

It was remarked earlier that sensitivity to initial conditions (SIC) is a key feature of chaos. There is a measure of the exponential divergence of trajectories, namely Lyapunov exponents. There is a huge literature on this concept (e.g. early papers such as, Eckmann, Kamphorst, Ruelle and Ciliberto [21], Wolf, Swift, Swinney and Vastano [78], and Sprott [67]) that measure this sensitivity. To assess divergence of nearby trajectories from time series, it is necessary to have data describing more than one trajectory, or a trajectory that returns rather close to earlier points of itself. Measuring Lyapunov exponents, either averaged over the trajectory or as a function of position in the embedding space is obviously a very difficult problem. As the Goldilocks tale at the beginning of this article hinted, the short, noisy time series available to many fields of observational astronomy are simply not sufficient to determine the exponential divergences contained in the Lyapunov exponent. There are no cases in astronomy where a reliable Lyapunov exponent has been determined from time series data.

Related analytic topics are prediction of time series and determination of the equations of motion of the observed system, to assess possible nonlinearity, randomness, nonstationarity, *etc.* Prediction of a process from observations of its past is as important in the analysis of chaotic time series as it is for random ones. Accordingly a large literature has developed (e.g. Abarbanel, Brown and Kadtke [1], Crutchfield [16], Crutchfield and McNamara [18], Crutchfield and Packard [19], Crutchfield and Young [20], Farmer and Sidorowich [24]). Various authors have pursued the related topic of *noise reduc-*

tion [25,38,69]. Relevant scaling properties are discussed by Atmanspacher, Scheingraber and Voges [4]. Hempelmann and Kurths [33] discuss analysis of the luminosity of cataclysmic variables (SS Cygni in particular) using symbolic dynamics, an approach that has considerable promise in this area [20]. But again, data limitations have prevented much success in the application of these methods in astronomy.

Let's end with a brief discussion of variability in brightness of astronomical objects, in various wavelength ranges. Regev [55] and Buchler and Regev [13] reviewed theoretical work on possible chaotic dynamics of stellar pulsations and analysis of variable star data. Numerical work that suggests chaotic behavior in stellar models includes [3,11,12,39,48]. Gillet [29] on the other hand does not confirm the period-doubling bifurcation scenario suggested by Kovács and Buchler [39] and intermittency and period-doubling bifurcations suggested in the work of Aikawa [4]. Serre, Buchler, and Goupil [65] study forecasting of variable star light curves using nonlinear methods. See also Kollath [37].

A final study demonstrates a data-driven approach. Scargle, Steiman-Cameron, Young, Donoho, Crutchfield and Imamura [64] started with a beautiful time series of X-ray observations of the bright x-ray source in Scorpius (Sco X-1) from the EXOSAT satellite. The observations covered nearly 10 hours, with a time resolution of 2 ms – thus covering 7 decades of time-scales. The only significant source of noise in these observations was the fluctuations associated with photon counting statistics, often called Poisson noise. Above such noise background this source evidences random variations of large amplitude and over a wide range of time scales.

In carrying out simple exploratory time series plots, it became evident that the general appearance of this variability is independent of time scale. This *fractal* or *scaling behavior* is common astrophysical systems (see Mandelbrot [43,44], and Flandrin [26] for the mathematics). This led to consideration of nonlinear physical models with outputs having such behavior. The classical dripping faucet system, the first archetypical chaotic system [66], had been generalized from a point system to a spatially extended one [17], the *dripping handrail* – so-called because the physical metaphor involved condensation of dew on a horizontal rail. Substituting accreting hot plasma for water, the inner edge of the accretion disk for the handrail, and a density driven instability for the drip mechanism, we hypothesized that the stochastic behavior in Sco X-1 can be described in terms of this kind of nonlinear dynamical system (see Young and Scargle [79] for a more detailed exposition of the model).

We studied the x-ray luminosity time series using the wavelet spectrum (called the *scalegram*), an analog of the ordinary Fourier power spectrum better suited to displaying power-law power spectra (“ $1/f$ noise”) associated with self-scaling in time series data. The basic idea was to compare scalegrams from the data against ones computed from the output of numerical simulations of the dripping handrail model. It was natural that the model produced power-law wavelet spectra, but astonishingly it also produced a local peak in the scalegram – a spectral feature indicating what astronomers call a quasi-periodic oscillation, or QPO. Astonishing because we were not even trying to fit this feature, and because the conventional wisdom was that the $1/f$ noise and the QPO are produced by completely different phenomena. Here a single, incredibly simple, physical model yielded both spectral features, fitting the scalegram estimated from the time series data quite well. Note that wavelets [22,73] are useful tools for diagnosing stochastic (random or chaotic) time series. Since the corresponding transforms are linear, this is a counterexample to the myth that linear tools are not useful for “nonlinear data.”

A similar model to the dripping handrail was proposed by Mineshige and co-workers (Mineshige, Takeuchi, and Nishimori [46] and earlier references therein), based on another non-linear dynamical paradigm called *self-organized criticality*. The physical metaphor for their work was the instabilities when sand is gradually dropped onto a flat surface, ultimately generating avalanches with interesting self-scaling properties, similar to those of some accretion systems [8]. Hence this is sometime called the *sand pile model*. Accretion systems in astronomy are complex and still not well understood. While there is not general acceptance of nonlinear dynamics as explaining the scaling and QPO features in the power spectra, this approach is competitive with the others under study.

There reader may wish to consult several general collections of papers dealing with chaotic phenomena in astrophysics [9,10,56].

Future Directions

Admittedly this article has a somewhat gloomy tone, since it has emphasized the difficulties of applying many of the standard data analysis methods to astronomical time series, including problems with the data as well as limitations on existing analysis methods. But there will be great future progress in this topic, due to both increased data quantity and quality and the development of more sophisticated data analysis technology. As we enter the Age of Digital Astronomy, the improved detectors, telescopes, and space

platforms are already producing large-scale surveys of the sky with good time coverage and resolution, coupled with photometric accuracy. These platforms will yield cornucopias of data which – coupled with advanced machine learning and data mining techniques, adapted and applied to the study of nonlinear dynamical systems from time series data – will greatly improve our knowledge of the astonishingly violent, irregular processes occurring throughout the Universe. [As a specific aside, it may be noted that “projection pursuit” [27,35] provides ways to deal with multidimensional data of any kind. While most workers have used it only to search for clustering in multidimensional data, projection pursuit may prove to be useful for state space structures.] In addition, multivariate visualization techniques, especially those aimed at elucidating the time evolution of states of a complex system, will be crucial to future progress.

Bibliography

Primary Literature

1. Abarbanel H, Brown R, Kadtke J (1989) Prediction and system identification in chaotic nonlinear systems: Time series with broadband spectra. *Phys Lett A* 138:401–408
2. Abraham R, Shaw C (1992) *Dynamics, the Geometry of Behavior*. Addison Wesley, Reading
3. Aikawa T (1987) The Pomeau–Manneville Intermittent Transition to Chaos in Hydrodynamic Pulsation Models. *Astrophys Space Sci* 139:218–293
4. Aikawa T (1990) Intermittent chaos in a subharmonic bifurcation sequence of stellar pulsation models. *Astrophys Space Sci* 164:295
5. Atmanspacher H, Scheingraber H, Voges W (1988) Global scaling properties of a chaotic attractor reconstructed from experimental data. *Phys Rev A* 37:1314
6. Badii R, Broggi G, Derighetti B, Ravani M, Cilberto S, Politi A, Rubio MA (1988) Dimension Increase in Filtered Chaotic Signals. *Phys Rev Lett* 60:979–982
7. Badii R, Politi A (1986) On the Fractal Dimension of Filtered Chaotic Signals. In: Mayer-Kress G (ed) *Dimensions and Entropies in Chaotic Systems, Quantification of Complex Behavior*. Springer Series in Synergetics, vol 32. Springer, New York
8. Bak P, Tang C, Wiesenfeld K (1987) Self-organized criticality: An explanation of $1/f$ noise. *Phys Rev Lett* 59:381–384
9. Buchler J, Eichhorn H (1987) Chaotic Phenomena in Astrophysics. *Proceedings of the Second Florida Workshop in Nonlinear Astronomy*. Annals New York Academy of Sciences, vol 497. New York Academy of Science, New York
10. Buchler J, Perdang P, Spiegel E (1985) Chaos in Astrophysics. Reidel, Dordrecht
11. Buchler JR, Goupil MJ, Kovács G (1987) Tangent Bifurcations and Intermittency in the Pulsations of Population II Cepheid Models. *Phys Lett A* 126:177–180
12. Buchler JR, Kovács G (1987) Period-Doubling Bifurcations and Chaos in W Virginis Models. *Ap J Lett* 320:L57–62
13. Buchler JR, Regev O (1990) Chaos in Stellar Variability. In: Krasner S (ed) *The Ubiquity of Chaos*. American Association for the Advancement of Science, Washington DC, pp 218–222
14. Caswell WE, York JA (1986) Invisible Errors in Dimension Calculation: geometric and systematic effects. In: Mayer-Kress G (ed) *Dimensions and Entropies in Chaotic Systems, Quantification of Complex Behavior*. Springer Series in Synergetics, vol 32. Springer, New York
15. Chennaoui A, Pawelzik K, Liebert W, Schuster H, Pfister G (1988) Attractor reconstruction from filtered chaotic time series. *Phys Rev A* 41:4151–4159
16. Crutchfield J (1989) Inferring the dynamic, quantifying physical Complexity. In: Abraham N et al (eds) *Measures of Complexity and Chaos*. Plenum Press, New York
17. Crutchfield J, Kaneko K (1988) Are Attractors Relevant to Fluid Turbulence? *Phys Rev Lett* 60:2715–2718
18. Crutchfield J, McNamara B (1987) Equations of motion from a data series, *Complex Syst* 1:417–452
19. Crutchfield J, Packard NH (1983) Symbolic Dynamics of Noisy Chaos. *Physica D* 7:201–223
20. Crutchfield J, Young K (1989) Inferring Statistical Complexity. *Phys Rev Lett* 63:105–108
21. Eckmann J-P, Kamphorst SO, Ruelle D, Ciliberto S (1986) Liapunov Exponents from Time Series. *Phys Rev A* 34:4971–4979
22. Fang L-Z, Thews RL (1998) *Wavelets in Physics*. World Scientific, Singapore. A physics-oriented treatment of wavelets, with several astronomical applications, mostly for spatial or spectral data
23. Farmer JD, Ott E, Yorke JA (1983) The dimension of chaotic attractors. *Physica D* 7:153–180
24. Farmer JD, Sidorowich J (1987) Predicting chaotic time series. *Phys Rev Lett* 59:845–848
25. Farmer JD, Sidorowich J (1988) Exploiting chaos to predict the future and reduce noise. In: Lee Y (ed) *Evolution, Learning and Cognition*. World Scientific Pub Co, Singapore
26. Flandrin P (1999) Time-Frequency/Time-Scale Analysis. Academic Press, San Diego. Volume 10 in an excellent series, *Wavelet Analysis and Its Applications*. The approach adopted by Flandrin is well suited to astronomical and physical applications
27. Friedman JH, Tukey JW (1974) A Projection Pursuit Algorithm for Exploratory Data Analysis. *IEEE Trans Comput* 23:881–890
28. Froehling H, Crutchfield J, Farmer JD, Packard NH, Shaw R (1981) On Determining the Dimension of Chaotic Flows. *Physica D* 3:605–617. *C Hercules. Astron Astrophys* 259:215–226
29. Gillet D (1992) On the origin of the alternating deep and shallow light minima in RV Tauri stars: R Scuti and A. *Astron Astrophys* 259:215
30. Grassberger P, Procaccia I (1983) Characterization of strange attractors. *Phys Rev Lett* 50:346–349
31. Grassberger P, Procaccia I (1983) Estimation of the Kolmogorov entropy from a chaotic signal. *Phys Rev A* 28:2591–2593
32. Harding A, Shinbrot T, Cordes J (1990) A chaotic attractor in timing noise from the VELA pulsar? *Astrophys J* 353:588–596
33. Hempelmann A, Kurths J (1990) Dynamics of the Outburst Series of SS Cygni. *Astron Astrophys* 232:356–366
34. Holzfuss J, Mayer-Kress G (1986) An Approach to Error-Estimation in the Application of Dimension Algorithms. In: Mayer-Kress G (ed) *Dimensions and Entropies in Chaotic Systems, Quantification of Complex Behavior*. Springer Series in Synergetics, vol 32. Springer, New York

35. Jones MC, Sibson R (1987) What is Projection Pursuit? *J Roy Statist Soc A* 150:1–36
36. Klavetter JJ (1989) Rotation of Hyperion. I – Observations. *Astron J* 97:570. II – Dynamics. *Astron J* 98:1855
37. Kollath Z (1993) On the observed complexity of chaotic stellar pulsation. *Astrophys Space Sci* 210:141–143
38. Kostelich E, Yorke J (1988) Noise reduction in dynamical systems. *Phys Rev A* 38:1649–1652
39. Kovács G and Buchler J R (1988) Regular and Irregular Pulsations in Population II Cepheids. *Ap J* 334:971–994
40. Lecar M, Franklin F, Holman M, Murray N (2001) Chaos in the Solar System. *Ann Rev Astron Astrophys* 39:581–631
41. Lissauer JJ (1999) Chaotic motion in the Solar System. *Rev Mod Phys* 71:835–845
42. Lissauer JJ, Murray CD (2007) *Solar System Dynamics: Regular and Chaotic Motion*. Encyclopedia of the Solar System, Academic Press, San Diego
43. Mandelbrot B (1989) Multifractal Measures, Especially for the Geophysicist. *Pure Appl Geophys* 131:5–42
44. Mandelbrot B (1990) Negative Fractal Dimensions and Multifractals. *Physica A* 163:306–315
45. Mayer-Kress G (ed) (1986) *Dimensions and Entropies in Chaotic Systems, Quantification of Complex Behavior*. Springer Series in Synergetics, vol 32. Springer, New York
46. Mineshige S, Takeuchi M, Nishimori H (1994) Is a Black Hole Accretion Disk in a Self-Organized Critical State. *ApJ* 435:L12
47. Mitschke F, Moeller M, Lange W (1988), Measuring Filtered Chaotic Signals. *Phys Rev A* 37:4518–4521
48. Moskalik P, Buchler JR (1990) Resonances and Period Doubling in the Pulsations of Stellar Models. *Ap J* 355:590–601
49. Norris JP, Matilsky TA (1989) Is Hercules X-1 a Strange Attractor? *Ap J* 346:912–918
50. Osborne AR, Provenzale A (1989) Finite correlation dimension for stochastic systems with power-law spectra. *Physica D* 35:357–381
51. Packard NH, Crutchfield JP, Farmer JD, Shaw RS (1980) Geometry from a Time Series. *Phys Rev Lett* 45:712–716
52. Paoli P, Politi A, Broggi G, Ravani M, Badii R (1988) Phase Transitions in Filtered Chaotic Signals. *Phys Rev Lett* 62:2429–2432
53. Provenzale A, Smith LA, Vio R, Murante G (1992) Distinguishing between low-dimensional dynamics and randomness in measured time series. *Physica D* 58:31
54. Ramsey JB, Yuan H-J (1990) The Statistical Properties of Dimension Calculations Using Small Data Sets. *Nonlinearity* 3:155–176
55. Regev O (1990) Complexity from Thermal Instability. In: Krasner S (ed) *The Ubiquity of Chaos*. American Association for the Advancement of Science, Washington DC, pp 223–232. Related work in press, *MNRAS*
56. Regev O (2006) *Chaos and complexity in astrophysics*, Cambridge University Press, Cambridge
57. Ruelle D (1990) Deterministic Chaos: The Science and the Fiction (the Claude Bernard Lecture for 1989). *Proc Royal Soc London Ser A Math Phys Sci* 427(1873):241–248
58. Ruelle D, Eckmann J-P (1992) Fundamental limitations for estimating dimensions and Lyapunov exponents in dynamical systems. *Physica B* 56:185–187
59. Scargle JD (1989) An introduction to chaotic and random time series analysis. *Int J Imag Syst Tech* 1:243–253
60. Scargle JD (1989) Random and Chaotic Time Series Analysis: Minimum Phase-Volume Deconvolution. In: Lam L, Morris H (eds) *Nonlinear Structures in Physical Systems*. Proceedings of the Second Woodward Conference. Springer, New York, pp 131–134
61. Scargle JD (1990) *Astronomical Time Series Analysis: Modeling of Chaotic and Random Processes*. In: Jaschek C, Murtagh F (eds) *Errors, Bias and Uncertainties in Astronomy*. Cambridge U Press, Cambridge, pp 1–23
62. Scargle JD (1990) Studies in astronomical time series analysis. IV: Modeling chaotic and random processes with linear filters. *Ap J* 343:469–482
63. Scargle JD (1992) Predictive deconvolution of chaotic and random processes. In: Brillinger D, Parzen E, Rosenblatt M (eds) *New Directions in Time Series Analysis*. Springer, New York
64. Scargle JD, Steiman-Cameron T, Young K, Donoho D, Crutchfield J, Imamura J (1993) The quasi-periodic oscillations and very low frequency noise of Scorpius X-1 as transient chaos – A dripping handrail? *Ap J Lett* 411(part 2):L91–94
65. Serre T, Buchler JR, Goupil M-J (1991) Predicting White Dwarf Light Curves. In: Vauclair G, Sion E (eds) *Proceedings of the 7th European Workshop*. NATO Advanced Science Institutes (ASI) Series C, vol 336. Kluwer, Dordrecht, p 175
66. Shaw R (1984) *The Dripping Faucet as a Model Chaotic System*. Aerial Press, Santa Cruz
67. Sprott JC (2003) *Chaos and Time-Series Analysis*, Oxford University Press, Oxford. This is a comprehensive and excellent treatment, with some astronomical examples
68. Steiman-Cameron T, Young K, Scargle J, Crutchfield J, Imamura J, Wolff M, Wood K (1994) Dripping handrails and the quasi-periodic oscillations of the AM Herculis. *Ap J* 435:775–783
69. Sugihara G, May R (1990) Nonlinear forecasting as a way of distinguishing chaos from measurement error in time series. *Nature* 344:734
70. Takens F (1981) Detecting strange attractors in turbulence. In: Rand DA, Young L-S (eds) *Dynamical Systems and Turbulence*, Lecture Notes in Mathematics, vol 898. Springer, pp 366–381
71. Theiler J (1986) Spurious Dimension from Correlation Algorithms Applied to Limited Time-series data. *Phys Rev A* 34:2427–2432
72. Theiler J (1990) Statistical Precision of Dimension Estimators. *Phys Rev A* 41:3038–3051
73. van den Berg JC (1998) *Wavelets in Physics*, Cambridge University Press, Cambridge. A physics-oriented treatment of wavelets, with a chapter by A Bijaoui on astrophysical applications, some relating to complex systems analysis
74. Vio R, Cristiani S, Lessi O, Provenzale A (1992) Time Series Analysis in Astronomy: An Application to Quasar Variability Studies. *Ap J* 391:518–530
75. Wiggins S (2003) *Introduction to Applied Nonlinear Dynamical Systems and Chaos*, 2nd edn. Springer, New York
76. Wisdom J (1981) The origin of the Kirkwood gaps: A mapping for asteroidal motion near the 3/1 commensurability. The resonance overlap criterion and the onset of stochastic behavior in the restricted three-body problem. Ph D Thesis, California Inst of Technology
77. Wisdom J, Peale SJ, Mignard F (1984) *Icarus* 58:137
78. Wolf A, Swift JB, Swinney HL, Vastano JA (1985) Determining Lyapunov exponents from a time series, *Physica D* 16: 285–317
79. Young K, Scargle J (1996) The Dripping Handrail Model: Transient Chaos in Accretion Systems. *Ap J* 468:617

Books and Reviews

- Heck A, Perdan JM (eds) (1991) *Applying Fractals in Astronomy*. Springer, New York. A collection of 10 papers
- Jaschek C, Murtagh F (eds) (1990) *Errors, Bias and Uncertainties in Astronomy*. Cambridge U Press, Cambridge
- Lowen SB, Teich MC (2005) *Fractal-Based Point Processes*, Wiley-Interscience, Hoboken. An excellent treatment of the unusual subject of fractal characteristics of event data generated by complex processes
- Maoz D, Sternberg A, Leibowitz EM (1997) *Astronomical Time Series*, Kluwer, Dordrecht. Proceedings of the Florence and George Wise Observatory 25th Anniversary Symposium
- Ruelle D (1989) *Chaotic Evolution and Strange Attractors: The Statistical Analysis of Time Series for Deterministic Nonlinear Systems*. Cambridge U Press, Cambridge
- Statistical Challenges in Modern Astronomy, a series of books derived from the excellent Penn State conference series, with many papers on time series or related topics. <http://astrostatistics.psu.edu/scma4/index.html>
- Subba Rao T, Priestly MB, Lessi O (eds) (1997) *Applications of Time Series Analysis in Astronomy and Meteorology*. Chapman & Hall, London. Conference proceedings; sample data were available to the participants

Astrophysics, Chaos and Complexity in

ODED REGEV^{1,2}

¹ Department of Physics, Technion–Israel – Institute of Technology, Haifa, Israel

² Department of Astronomy, Columbia University, New York, USA

Article Outline

Glossary

Definition of the Subject

Introduction

Hamiltonian Chaos in Planetary, Stellar and Galactic Dynamics

Chaotic Time Variability of Astronomical Sources

Spatio-Temporal Patterns and Complexity in Extended Systems

Future Directions

Bibliography

Glossary

Accretion disk A flat gaseous structure into which matter endowed with angular momentum settles when it is being gravitationally attracted to a relatively compact object. Dissipative processes, which are needed to allow for the extraction of angular momentum from

the orbiting gas so as to allow for it to be accreted by the star, heat the disk and make it observable by the outgoing radiation.

Attractor A geometrical object in the state space of a dynamical system to which the system's trajectories tend for large times (or iteration numbers). Attractors of dissipative systems have a dimension that is lower than that of the state space and when this dimension is fractal, the attractor is called *strange*.

Bifurcation A qualitative change in the essential behavior of a mathematical system (e.g., algebraic equations, iterated maps, differential equations), as a parameter is varied. Usually involves an abrupt appearance and/or disappearance of solutions.

Close binary stars Two stars in a Keplerian orbit around each other, having a small enough separation, so that they may affect each other during their evolution (for example by mass transfer). When the pair contains, in addition to a normal unevolved star, a compact object (in *cataclysmic binaries* it is a white dwarf and in *X-ray binaries* a neutron star or black hole) the system may be observationally prominent. Mass transfer from the normal star via an accretion disk onto the compact object may result in violent phenomena, such as nova explosions or X-ray bursts.

Dynamical system (DS) A set of rules by application of which the state of a physical (or some other, well defined) system can be found, if an initial state is known. Iterated maps (IM) in which the evolution proceeds in discrete steps, i.e., one defined by appropriate iterative formula, and differential equations, both ordinary (ODE) and partial (PDE) are typical DS.

DS flow All *trajectories* (or *orbits*) of a DS in its *state space*, which is spanned by the relevant state variables. The flow is fully determined by the DS and may serve as its geometrical representation.

Hubble time Estimate for the age of the universe obtained from taking the inverse of the *Hubble constant*, the constant ratio between the recession velocity of galaxies and the distance to them (Hubble law). Current estimates give the Hubble time as ~ 14 Gyr.

Interstellar medium (ISM) Diffuse matter of low density, filling the space between stars in a galaxy. The average number density of the ISM is typically $n \sim 1 \text{ cm}^{-3}$ but its distribution is highly non-uniform, with the densest clouds reaching $n \sim 10^6 \text{ cm}^{-3}$.

Fractal A set whose (suitably defined) geometrical dimension is non-integral. Typically, the set appears self-similar on all scales. A number of geometrical objects associated with chaos (e.g. strange attractors) are fractals.

Integrable DS A DS whose exact solution may be calculated analytically in terms of elementary or special functions and possibly quadratures. In integrable Hamiltonian DS the motion proceeds on a set of invariant tori in phase space.

Integral of motion A function of a DS's state variables which remains constant during the evolution. The existence of integrals of motion allows for a dimensional reduction of the DS. A sufficient number of independent integrals of motion guarantees integrability.

KAM theorem An important theorem by Kolmogorov, Arnold and Moser (KAM), elucidating the loss of integrability and transition to chaos in Hamiltonian DS by the breakdown of invariant tori.

Liapunov exponents A quantitative measure of the local divergence of nearby trajectories in a DS. If the largest such exponent, Λ , is positive, trajectories diverge locally exponentially, giving rise to chaotic behavior. In that case a measure for the time that chaos is manifested, the *Liapunov time*, is $\propto 1/\Lambda$.

Poincaré section A geometrical construction with the help of which the behavior of a multi-dimensional DS can be examined by means of a two-dimensional IM. The IM is obtained by finding the system's trajectory successive crossings of a given surface cutting the state space of the DS (*surface of section*). Essential features of the dynamics can be captured if the surface is placed in an appropriate position.

Pulsating stars Stars that, due to an instability, exhibit radial or non-radial pulsation. Most have periodic variations, with periods ranging from a few days to years, but some are irregular. The classical Cepheids, which are the most common regular pulsators, were instrumental in determining cosmological distances by using their known period-luminosity relation.

Shilnikov scenario One of the known mathematical structures that guarantee chaos in dissipative DS. Let an unstable fixed point exist in the DS state space and have a fast unstable direction and a stable subspace of trajectories that slowly spiral in. If a homoclinic (connecting the point to itself) orbit exists then there are chaotic orbits around it.

Definition of the Subject

Astronomy is the science that deals with the origin, evolution, composition, distance to, and motion of all bodies and scattered matter in the universe. It includes astrophysics, which is usually considered to be the theoretical part of astronomy, and as such focuses on the physical properties and structure of cosmic bodies, scattered

matter and the universe as a whole. Astrophysics exploits the knowledge acquired in physics and employs the latter's methods, in an effort to model astronomical systems and understand the processes taking place in them.

In recent decades a new approach to nonlinear dynamical systems (DS) has been introduced and applied to a variety of disciplines in which DS are used as mathematical models. The theory of chaos and complexity, as this approach is often called, evolved from the study of diverse DS which behave unpredictably and exhibit complex characteristics, despite their seeming simplicity and deterministic nature. The complex behavior is attributed to the property of *sensitivity to initial conditions* (SIC), whereby despite their completely deterministic behavior, two identical systems in initial states differing only by a minute amount, relatively rapidly develop in very different ways. The theory has been most fruitful in physics, but it is important to note that its first paradigms were actually deeply related to astrophysical systems.

A large majority of astrophysical systems are theoretically modeled by nonlinear DS. The application of the ideas and methods of chaos and complexity theory to astrophysics seems thus to be natural. Indeed, these methods have already been exploited in the study of some systems on a vast variety of scales – from planetary satellites through pulsating stars and up to the large scale structure of the universe. The main importance of these approaches is in their ability to provide new analytical insights into the intricate physical processes taking place in cosmic matter, shaping the various astronomical objects and causing the prominent observable phenomena that occur in them.

Introduction

Newton's laws of mechanics and gravitation had their most prominent application in the significant work of Laplace, *Celestial Mechanics*. It appeared in five volumes between 1798 and 1827 and summarized his mathematical development and application of Newton's work. Laplace offered a complete mechanical interpretation of the Solar System – planets and their satellites, including the effects of perturbations and tidal interactions. This work had immediately been adopted as an unequivocal manifestation of Nature's determinism, that is, the possibility to precisely determine the future of any DS – a set of ordinary differential equations (ODE) in this case – if only appropriate initial conditions are known. Laplace himself was so confident in his results that, according to one story, when Napoleon asked him what is the role of the Creator in his theory, he replied that "this hypothesis" was redundant.

Remarkably, the same system in celestial mechanics (and actually its simplest form – containing just three bodies) which had been considered to be the primary paradigm of determinism gave rise, almost a century after Laplace’s work, to profound analytical difficulties. Another great French mathematician, Henri Poincaré, made the fundamental discovery that the gravitational n -body system is non-integrable, that is, its general solution can not be expressed analytically, already for $n > 2$. This work initiated the development of the theory of chaos in Hamiltonian systems, which culminated, about 50 years ago, in the Kolmogorov–Arnold–Moser (KAM) theorem. Section “[Hamiltonian Chaos in Planetary, Stellar and Galactic Dynamics](#)” of this article reviews the application of this theory to planetary, stellar and galactic dynamics.

In unrelated investigations of dissipative DS (Hamiltonian systems are conservative) in the 1960s, aperiodic behavior was observed in numerical studies of simplistic models of thermal convection in geophysical and astrophysical contexts by Lorentz [40] and Moore and Spiegel [44]. It was conjectured that this behavior is due to SIC, with which these systems were endowed. By the same time, the American mathematician Stephen Smale discovered a new class of “strange” attractors on which the dynamics is chaotic and which naturally arise in DS – sets of ODE or iterated maps (IM) – endowed with SIC. In the early 1980s, it was realized, following the work of Mitchell Feigenbaum and others on bifurcations in quadratic IM, that such DS possess a universal behavior as they approach chaos. Some applications of these developments to astrophysical systems are discussed in Sect. “[Chaotic Time Variability of Astronomical Sources](#)”.

A variety of important physical (and astrophysical) systems, notably fluids, are modeled by partial differential equations (PDE). This class of DS provides the *spatio-temporal* information, necessary for the description and understanding of the relevant phenomena. These DS, especially when they are nonlinear (as is often the case), are significantly more complicated than ODE and IM, but appropriate reduction techniques can sometimes help in applying to them the knowledge gained for ODE and IM. These and a variety of other techniques constitute what is called today *pattern theory*. Spatio-temporal patterns have been identified and categorized and they help to understand, using analytical and semi-analytical (usually perturbative) approaches, some essential properties of the solutions. It is of interest to properly quantify, describe and understand the processes that create spatio-temporal complexity, like the shapes of interstellar clouds, patterns in thermal convection, turbulent flows and other astrophysically relevant topics. Section “[Spatio-Temporal Patterns and Complex-](#)

[ity in Extended Systems](#)” is devoted to a review of these issues.

The quest for the universal and the generic, from which an understanding of complicated and seemingly erratic and heterogeneous natural phenomena can emerge, is central to the applications of the theory of chaos, patterns and complexity in DS to astrophysics. It may provide a powerful investigative tool supplementing fully fledged numerical simulations, which have traditionally been employed in the study of nonlinear DS in astrophysics.

Hamiltonian Chaos in Planetary, Stellar and Galactic Dynamics

Planetary systems, star clusters of various richness, galaxies, galaxy clusters and super-clusters (clusters of clusters) can be approached most directly by considering the gravitational n -body problem (GNBP). In this approach all other interactions, save the gravitational one, are neglected, and the bodies are considered to be point masses. This model is viable, at least approximately, when the distances between the bodies are much larger than their typical size and any scattered matter between the bodies has only a very small effect on the dynamics.

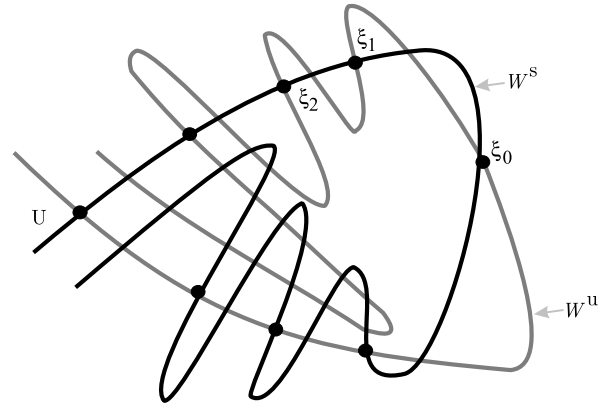
The GNBP is one of the paradigms of classical dynamics, whose development originated in Newton’s laws and through the implementation of techniques of mathematical analysis (mainly by Lagrange in his work *Analytical Mechanics*, published in 1788) finally acquired a powerful abstract formulation in the form of Hamiltonian canonical formalism. Hamiltonian systems, that is, those that obey the Hamilton equations are endowed with important conservation properties linked to symmetries of the Hamiltonian, a function that completely describes the system. This function is defined on the configuration space which consists of the *phase space* (spanned by the generalized coordinates and momenta) and the time coordinate. A Hamiltonian system conserves phase volume and sometimes also other quantities, notably the total energy (when the Hamiltonian does not depend explicitly on time, as is the case in the GNBP).

The Hamilton equations consist of two first order ODE for each degree of freedom, thus the GNBP in a three-dimensional physical space yields, in general, $6n$ equations and so for large n it is quite formidable. The $n = 2$ case is reducible to an equivalent one body problem, known as the Kepler problem. The complete solution of this problem was first given by Johann Bernoulli in 1710, quite long before the Lagrange–Hamilton formalism was introduced. The gravitational two-body problem has been successfully applied to various astrophysical systems, e. g. the

motion of planets and their satellites and the dynamics of binary stars. The quest for a similar reduction for systems with $n > 2$, was immediately undertaken by several great mathematicians of the time. The side benefit of these studies, conducted for almost two centuries, was a significant progress in mathematics (mainly in the theory of ODE), but definite answers were found in only some particular, rather limited cases. Attempts to treat even the simplest problem of this kind, (the restricted, planar, three-body, i. e., where one of the bodies is so light that its gravity has no effect on the dynamics and all the orbits are restricted to a plane) ended in a failure. All general perturbative approaches invariably led to diverging terms because of the appearance of “small divisors” in the perturbation series.

The first real breakthrough came only in the 1880s, when Henri Poincaré worked on the GNB, set for a prize by King Oscar II of Sweden. Poincaré did not provide a solution of the problem, but he managed to understand why it is so hard to solve. By ingenious geometrical arguments, he showed that the orbits in the restricted three-body problem are too complicated to be described by any explicit formula. In more technical terms, Poincaré showed that the restricted three-body problem, and therefore the general GNB, is *non-integrable*. He did so by introducing a novel idea, now called a Poincaré section, with the help of which he was able to visualize the essentials of the dynamics by means of a two-dimensional area preserving IM. Figure 1 shows schematically a typical mathematical structure, a *homoclinic tangle* in this case, that is behind chaotic behavior in systems like the restricted three-body problem (and, incidentally, also the forced pendulum). The existence of a hyperbolic fixed point in the appropriate equivalent IM and the transversal intersection of its stable and unstable manifolds gives rise to the complex behavior. Remarkably, Poincaré was able to visualize such a structure without the aid of computer graphics.

The non-integrability of the GNB naturally prompted the question of the Solar System stability. No definite answer to this problem could however be reasonably expected on the basis of analytical work alone. Before electronic computers became available, this work had largely been based on perturbation or mean field methods. Despite the difficulties, these efforts yielded new and deep insights on chaotic behavior in Hamiltonian systems after intensive work of close to 50 years. First and foremost among these is the KAM theorem, which elucidated the mathematical process by which an integrable Hamiltonian system transits to chaotic behavior by losing its integrability, when a suitably defined control parameter (e. g. the relative size of the non-integrable perturbation to an integrable Hamiltonian) is gradually increased from



Astrophysics, Chaos and Complexity in, Figure 1

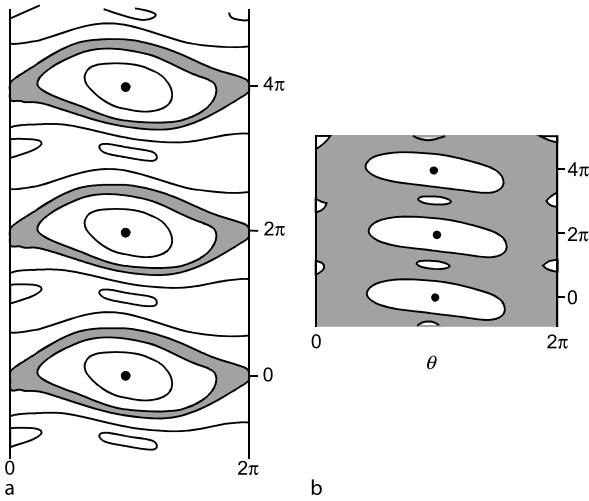
Schematic illustration of the homoclinic tangle of the unstable saddle fixed point U . W^s and W^u are, respectively, the stable and unstable invariant sets (manifolds) of the saddle point. A transversal intersection of these manifolds (ξ_0) necessarily leads to an infinite number of subsequent intersections, resulting in chaotic behavior of the DS

zero. In particular, the crucial role of resonant tori in this process has been recognized, the transition starting with the increasing distortion of these tori. The resonant tori become ultimately corrugated on all scales, acquiring a fractal shape and allowing orbits to break-up from them and diffuse in the regions between the surviving non-resonant tori. Over 20 years after the final formulation of the KAM theorem, Boris Chirikov suggested a diagnostic criterion for the onset of chaos in Hamiltonian systems. He studied numerically the *standard map*

$$\begin{aligned} I_{j+1} &= I_j + K \sin \Theta_j \\ \Theta_{j+1} &= \Theta_j + I_j, \end{aligned} \quad (1)$$

a particular area preserving two-dimensional IM, where I and Θ are typically action-angle variables of a Hamiltonian system and K is a constant. He showed that fully-spread chaos arises when $K > 1$ and found that this happens because *resonances overlap*. This is schematically illustrated in Fig. 2.

The basic defining property of deterministic chaos is SIC (due to the divergence of initially arbitrarily close phase space trajectories of the DS) and it is quantified by the positivity of the largest *Liapunov exponent*, which guarantees chaotic behavior in all DS. In Hamiltonian systems, additional subdivision of different degrees of chaotic motion is available. As the Hamiltonian DS is driven farther into the chaotic regime the motion becomes more strongly irregular, in the sense that the DS trajectories in phase space explore progressively larger volumes. All



Astrophysics, Chaos and Complexity in, Figure 2

Non-resonant KAM tori and chaotic regions (*shaded*) in the standard map. Schematic drawing for an almost integrable case ($K \lesssim 1$) (a) and the fully chaotic case ($K > 1$) (b). The broken up resonant tori, displayed in a which is expanded so as to show more details, overlap and merge, giving the situation shown in b – widespread chaos

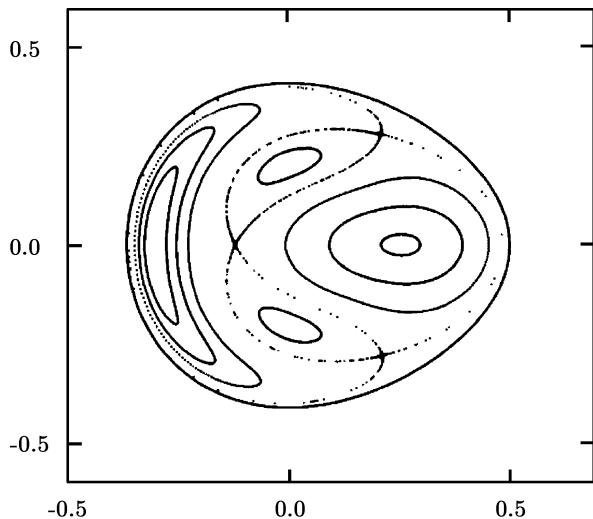
Hamiltonian DS are *recurrent*, i. e., their trajectories return infinitely many times arbitrarily close to the initial point (according to the Poincaré recurrence theorem this is true for almost all orbits). A DS is called *ergodic* when long-time averages of a variable is equivalent to its phase-space average. Hamiltonian system are ergodic on non-resonant tori. If any initial phase-space volume eventually spreads over the whole space, then the chaotic system is said to have the property of *mixing*. Deeper into the chaotic regime a Hamiltonian system becomes a *K-system*. This happens when trajectories in a connected neighborhood diverge exponentially on average and finally, when every trajectory has a positive Liapunov exponent, there is global instability and the system is called a *C-system*.

The first Hamiltonian system, directly relevant to astrophysics, which was explicitly shown to exhibit deterministic chaos, was that studied by Hénon and Heiles [29]. They numerically investigated the orbits of a point mass (a star) in a model potential (approximating that of an axially symmetric galaxy). Casting the gravitational influence of all the other masses in a system into a mean field and computing the orbits of a point mass in it, is significantly easier than a full n -body integration and has always been one of the primary techniques of galactic dynamics. The question addressed in this study involved the possible existence of a third isolating integral, which would guarantee integrability in this case. The fact that the motion in the az-

imuthal (with respect to the symmetry axis) direction can be separated out and that total energy and angular momentum are two independent integrals of motion leaves out a three-dimensional phase volume that is accessible to the system. A third isolating (i. e., independent of the other two) integral would thus guarantee motion on a two-dimensional phase surface only, that is, integrability. Using a model axi-symmetric potential, whose non-dimensional form

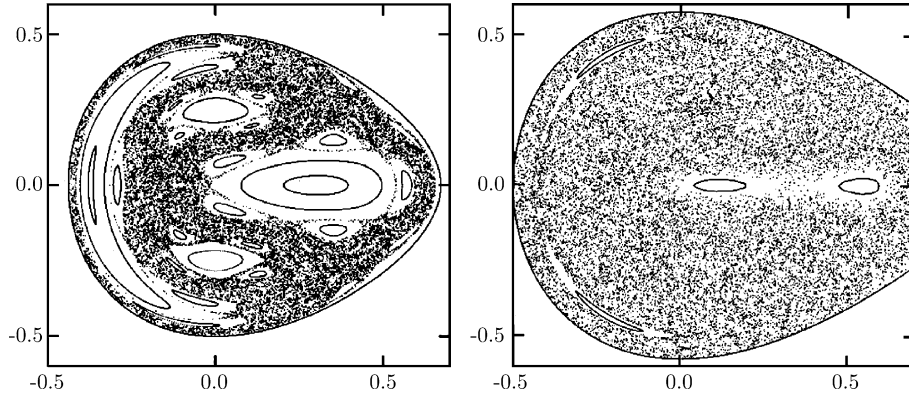
$$V(x, y) = \frac{1}{2} (x^2 + y^2) + \left(x^2 y - \frac{y^3}{3} \right), \quad (2)$$

is clearly seen to contain a simple nonlinear part in addition to the harmonic one, Hénon and Heiles integrated the Hamilton equations of motion for several values of E , the non-dimensional energy parameter. By examining the Poincaré section in each case, they found that the system is integrable and the trajectory proceeds on tori (Fig. 3) as long as E is small enough. For a certain, larger value of E some of the tori are destroyed, with a chaotic “sea” appearing in between the surviving ones, but finally above some other, still higher, value of E , almost all tori are destroyed and chaos becomes widespread (Fig. 4). These results largely follow the scenario set forth by the KAM theorem. They also settled the issue: A third integral of motion of galactic dynamics does *not* exist in general. The implications of these findings have since then been thoroughly investigated (see also below), primarily by George Con-



Astrophysics, Chaos and Complexity in, Figure 3

Surface of section in the x - y plane for the Hénon-Heiles system with a relatively low value of the energy parameter $E = 1/12$. The system is integrable and the motion proceeds on invariant tori



Astrophysics, Chaos and Complexity in, Figure 4

Same as Fig. 3, but for values $E = 1/8$ (left panel) and $E = 1/16$. Chaotic orbits that exist between the surviving tori eventually spread and fill almost all of the available space, with only limited islands of regular motions surviving

topoulos and his collaborators. Moreover, the pioneering work of Hénon and Heiles could perhaps not directly predict the properties of the stellar orbits and their distribution in some particular galaxy. It has, however, remained to this day a fundamental contribution to the theory of Hamiltonian DS.

The Hénon and Heiles system is actually a one-body problem (in a smooth external potential) and owes its chaotic behavior to the nonlinearity brought about by the anharmonic terms in the potential. Perhaps the simplest and most intriguing astrophysical realizations of the GNBPs involve the motion of bodies in the Solar System. There exists today ample theoretical evidence, based largely on series of gradually improving numerical studies, that chaos permeates the Solar System and can be found essentially everywhere one looks. The numerical calculations have recently been supplemented by analytical and semi-analytical work.

The earliest definite finding (theoretical *and* observational) of chaotic motion among the permanent bodies in the Solar System was in the rotational dynamics of the small and rather inconspicuous Saturn's moon, Hyperion. In 1981 the Voyager 2 mission to Saturn discovered that Hyperion's orbit, residing well outside the Saturnian rings, is quite eccentric and the moon itself is unusually flat and its spin behavior is quite strange. It was found, for example, that Hyperion is spinning along neither its longest axis nor the shortest one; thus its spinning motion must be unstable. On the theoretical side, even though this problem is not a realization of the GNBPs with $n > 2$, the inclusion of this moon's rotation around its axis may still allow for complicated dynamics in the system. The problem was analyzed by Wisdom, Peale and Mignard [63], who employed a known nonlinear equation for the time evolution

of the angle $\theta(t)$ between the moon's principal axis, having the smallest moment of inertia, and a fixed direction in space. The equation,

$$\frac{1}{n^2} \frac{d^2\theta}{dt^2} + \frac{\eta^2}{2} \left(\frac{a}{r}\right)^3 \sin[2(\theta - \phi)] = 0, \quad (3)$$

includes the orbital parameters of the moon $a(t)$, $\phi(t)$ and the orbital mean motion frequency $n \equiv \sqrt{GM_S/a^3}$, assumed to be given, and also the "asphericity" parameter of the moon's figure, $\eta \equiv \sqrt{3(I_B - I_A)/I_C}$, with $I_C \geq I_B > I_A$ being the principal moments of inertia of the moon.

Equation 3 can be perceived as describing a nonlinear oscillator subject to periodic parametric forcing, a non-integrable Hamiltonian system, and thus it could not be analytically solved, save for special cases that allow for perturbative treatment. Wisdom et al. integrated the equation numerically and examined the appropriate Poincaré section for various solutions. Re-scaling the time variable (formally $t \rightarrow nt$, so that the orbital period becomes 2π), one can easily see that the equation of motion results from a time-dependent Hamiltonian

$$H(x, p, t) = \frac{1}{2}p^2 + \frac{1}{2}\eta^2\alpha^2(t)\cos\{[2x - \phi(t)]\}, \quad (4)$$

where $x \equiv \theta$, $p \equiv \dot{\theta}$ and $\alpha(t) \equiv 2^{-1/2}[a/r(t)]^{3/2}$. Now, choosing orbital parameters appropriate for Hyperion's orbit Wisdom et al. found that even for the relatively small value of $\eta = 0.2$ (Hyperion actually has $\eta = 0.89$) narrow disconnected chaotic layers appear in the $p - x$ plane of section and they reside close to the resonances $p = 1/2, 1, 3/2$. For $\eta = 0.89$ there appears an extended chaotic zone. These characteristics of the surface of the section are reminiscent of the transition to chaos in the generic case depicted in Fig. 2, where the resonance over-

lap criterion is illustrated. The direction of Hyperion's axis of rotation is, thus, bound to vary chaotically and calculations show that significant changes can occur during just a few orbital times.

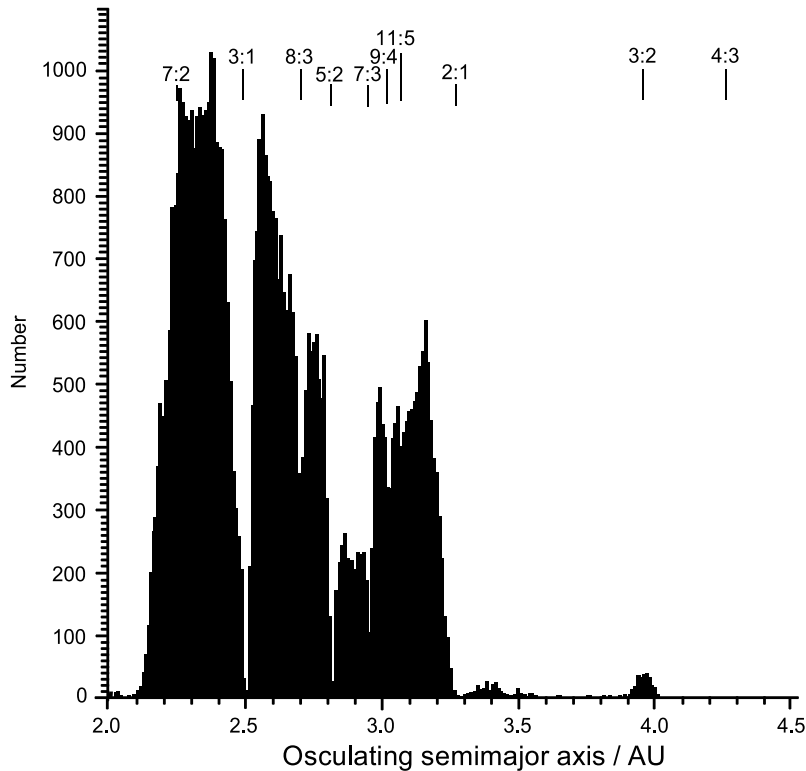
Chaotic tumbling of the kind found for Hyperion cannot be expected for planets orbiting the Sun, because for them $I_A \approx I_B$. Complex behavior in the rotational dynamics may arise, however, from the perturbations of the orbit caused by the presence of other planets. Laskar and Robutel [35] explored this possibility for all the inner Solar System planets and found that the obliquity of Mars is currently varying chaotically between 0° and $\sim 60^\circ$ on a time-scale of a few million years, and it has probably always been varying in this manner. The rotational dynamics of Mercury and Venus do not appear to be in the chaotic regime, but it is reasonable that these planets may have experienced, in their past, slow chaotic changes in the obliquity of the rotational axis. The Earth's case is quite intriguing because this analysis indicates that it should be now in the state of chaotic variation of its obliquity between 0° and $\sim 80^\circ$ with a time-scale of a few million years. Laskar, Joutel and Robutel [36] repeated the calculations for the Earth, including also the effects of the Moon, and obtained the comforting result that the Moon's torque appears to stabilize the Earth's rotational state, with the obliquity varying in a regular manner and by no more than $\pm 1.3^\circ$ about the mean value of 23.3° .

Just as the periodic orbital motion in the two-body problem may induce complex rotational dynamics in one of the bodies, tidal interaction due to the bodies' finite size may affect the orbit in a manner that is far from being simple. Mardling [41] investigated this possibility in the context of binary stars, formulating the problem by means of a Hamiltonian system including stellar, tidally induced, oscillations. She found that for some specific initial orbital parameters the radial separation and eccentricity may experience chaotic variations, giving rise to the possibility that an initially unbound orbit may change so as to give rise to a stable bound system (a binary). In subsequent studies the effect of such tidal captures on an evolution of a star cluster was investigated. In addition, Mardling noticed and pointed out the formal similarity of the tidal-capture problem to the three-body problem.

Since the early work of Daniel Kirkwood, it has been known that the space distribution of asteroids (also known as *minor planets*) is not uniform and, in particular, displays prominent gaps, which, as already Kirkwood himself had recognized, appear when the their period would be equal to a simple fraction of Jupiter's period (see Fig. 5). Early studies, based on the restricted three-body problem (an asteroid moving in the field of the Sun and Jupiter, or-

biting around each other in a Keplerian circle), had employed perturbative averaging techniques to the motion of the large bodies and failed to predict the Kirkwood gaps. Wisdom [62] noticed that these studies could not account for possible spreading of asteroid orbits through the entire chaotic regions accessible to them in phase space, because the existing KAM tori prevent it. This is a purely dimensional effect and such spreading is possible and may proceed (albeit on a very long time-scale) even if the system is close to integrability and resonance overlap does not occur, but the dimension of the accessible phase space has to be higher by at least two than that of the tori. To allow for the possibility of *Arnold* diffusion (so named after its discoverer), Wisdom suggested abandoning the previous approach (giving rise to only four-dimensional phase space) and use, instead, the three-dimensional restricted *elliptic* problem. He proposed studying an equivalent discrete map (because full numerical integration over a long enough time was not feasible then), whose structure near one particular commensurability (the 3:1 ratio with Jupiter's motion) is similar to that of the full Hamiltonian flow. Test particles, initially placed near the 3:1 commensurability in low eccentricity orbits, were found to typically hover around the initial orbit for almost a million years, and then suffered a sudden large increase of eccentricity, which would put them in a Mars-crossing orbit. Thus an asteroid could not survive for a long time and be observed in an orbit close to the 3:1 resonance. It would ultimately be removed from it by an encounter with Mars. Subsequent full numerical integration confirmed this idea and provided an explanation for the formation of this prominent Kirkwood gap and, in principle, the other gaps as well. Another important finding of these studies was that even though the Liapunov time for this system is relatively short ($\sim 10^3$ years), the chaotic increase of eccentricity occurred only after a significantly longer time. This is typical for systems close to integrability, when chaotic trajectories can explore the available phase space only very slowly, in this sense, by means of Arnold diffusion. This situation is sometimes referred to as *stable chaos* (see also below).

The outer Solar System is abundant in minor bodies as well – asteroids and comets. Comets are thought to originate from icy planetesimals that formed at the outskirts of the primitive Solar nebula. Some of them found stable orbits in the Kuiper belt, a relatively narrow and flat ring-like region starting just outside the orbit of Neptune and ending ~ 50 AU from the Sun. Other comets were probably scattered by encounters with nearby stars into a vast spherical shell-like reservoir, ranging between approximately 30,000 AU and 50,000 AU from the Sun – the



Astrophysics, Chaos and Complexity in, Figure 5

The number of asteroids as a function of the size (in AU). The location of the Kirkwood gaps, marked by the corresponding resonances with Jupiter motion, is clearly apparent. Reproduced, with permission, from [37], © Annual Reviews

Oort cloud. This topic is still quite controversial nowadays, but extensive numerical calculations suggest that, in any case, chaos seems to be playing a major role in cometary dynamics (cf. [19]). Most comets coming from the Oort cloud (long-period comets) are thought to have chaotic orbits, the origin of chaos being attributed in this case to repeated close encounters with massive planets. Sagdeev and Zaslavsky [52] showed that encounters between Jupiter and a comet on an almost parabolic orbit can account for a large chaotic zone, extending into the Oort cloud. More specifically, attempts to numerically calculate the orbit of the famous Halley comet, close to the time of its latest visit in 1985, failed to agree with each other and with observations. This problem may have been caused by the fact that Halley's orbit is in fact chaotic. Chirikov and Vecheslavov [15]) supported this suggestion and estimated that the timescale for practical unpredictability in this case could be as short as ~ 30 years. Regarding the short-period comets, numerical studies have indicated that there are practically no stable orbits among the outer planets. Such orbits have, however, been found above ~ 40 AU

from the Sun and it is thus reasonable that planetesimals indeed could survive for very long times in the Kuiper belt. Close to the stable regions there exist also regions of instability, from which comets could be injected into the inner Solar System by means of Arnold diffusion and ultimately (but probably only temporarily) be captured into resonant orbits, by interaction with other planets, and appear as a short-period comets.

The most difficult dynamical problem in the Solar System results from the inclusion of all (or, at least, a few massive) planets and the Sun in a full GNPB. Since the late 1980s a number of numerical integrations, employing several different approaches, have been performed on this problem. The results of these studies were generally in rather remarkable agreement, although different techniques had been used to allow for long integration times. Positive Liapunov exponents, indicating chaos, have been found for the orbits of most planets and the Liapunov times t_A have been estimated, as follows. Sussman and Wisdom [57] found $10 \lesssim t_A \lesssim 20$ Myr for Pluto, $t_A \sim 4$ Myr for the inner planets and $3 \lesssim t_A \lesssim 30$ Myr for

the outer ones. Murray and Holman [46] refined the result for the Jovian planets, and determined it more accurately, at $t_A \sim 5$ Myr. Laskar [33] found $t_A \sim 4$ Myr for the inner planets and, in addition, pointed out that despite this value, the effects of chaos on the orbits of these planets remain small for at least 5 Gyr (with the exception of Mercury, whose orbit was found to be able to change drastically its eccentricity within ~ 3.5 Gyr). Similar conclusions have also been reached for the outer planets. It thus appears that the disparity between the Liapunov time and the time after which a dramatic change in the orbit may actually appear is similar for the planetary orbits to that of the asteroid orbits and the term “stable chaos” is appropriate to planetary orbits as well.

Stellar dynamics in clusters and galaxies is, in principle, a realization of the GNBPs with large n . Direct numerical simulations of such problems are still, even nowadays, often requiring prohibitive computational resources. Historically, the approach to such problems had largely been based on statistical methods, inspired by the impressive success of statistical mechanics in handling systems composed of an enormous number of microscopic particles. Typical relaxation times, relevant to dynamical astronomy are, however, too long for stars to have settled into the invoked (and observed) statistical equilibria in clusters and galaxies. The understanding of the underlying mechanism of a *violent* relaxation process, that was proposed to solve this problem, calls for a better understanding of the n -body dynamics. To circumvent the full GNBPs, studies of orbits in a given mean field potential were conducted, but the question of self-consistency (of the possibly evolving background potential) posed a nontrivial challenge to this approach.

Already the pioneering study of Hénon and Heiles [29] showed that chaotic orbits are expected even in simplistic steady potentials. In a number of more recent works it has been found that chaotic orbits in realistic galactic potentials tend to diffuse over the entire phase space available to them, in times shorter than Hubble time. For example, Hasan, Pfenniger and Norman [27] showed this for barred galaxies and Merritt and Fridman [42] for those having triaxial, strongly cusped mass distributions. In addition, Merritt and Valluri [43] found that random time-dependent perturbations in the mean potential are likely to enhance the rate of diffusion of the orbits. Habib, Kandrup and Mahon [26] investigated these type of effects in more detail by explicitly including weak friction and noise in the calculation of orbits in non-integrable two-dimensional potentials. Habib et al. found that even in two-dimensional, but strongly chaotic, potentials, the statistical properties of orbit ensembles (and indeed the struc-

ture of phase space itself) were altered dramatically (on a timescale much shorter than t_R) by the presence of friction and noise.

All these studies have essentially been modifications of the mean field dynamics, but it is reasonable to conclude from them and some similar calculations that the full GNBPs can be expected to contain a significant amount of irregularities and exhibit evolution on a short (compared to t_R) timescale. Thus, quite a strong evolution of the phase space distribution is expected to occur already in time-independent (but non-integrable) potentials. The mechanism driving this evolution was identified to be associated with chaotic diffusion or *mixing*. The numerical experiments mentioned above suggest that mixing behavior indeed does occur in some realistic potentials. Moreover, it is significantly enhanced by cusps and irregularities (which have been modeled by noise and friction) in the potential.

What do these findings imply for galactic dynamics? It is quite clear that this question cannot be answered within the realm of the mean field approach. If the stochastic orbits are causing a slow (but not too slow) evolution of the halo, chaos probably plays a role in galactic evolution. Moreover, it is possible that the underlying potential is not only evolving in time, but also cannot be assumed to be smooth. One has to treat thus the full GNBPs, which is clearly the fully self-consistent approach to the problem of cluster or galactic dynamics. While it is clear that the system is chaotic, it is still not clear into which category of chaotic Hamiltonian system it should be classified. Already Gurzadyan and Savvidy [25] have shown that the GNBPs does not satisfy the requirements for it to be classified as a C-system. Subsequently, El-Zant [20] pointed out that conventional methods used for quantifying phase-space mixing are not adequate for the GNBPs because of singularities (e.g. collisions), and proposed to use local geometric properties rather than global ones. He also addressed, in a more recent study, the influence of potential “softening”, usually employed in n -body simulations to prevent singularities. The role of chaos in cluster and galactic dynamics and in dynamical astronomy is still not fully understood and some of the issues remain controversial. The subject is discussed in detail in the recent book by Contopoulos [16].

Chaotic Time Variability of Astronomical Sources

A variety of astronomical sources emit radiation whose intensity is intrinsically a time variable. This means that the variability is the object’s inherent property and does not result from geometrical (e.g. due to the object’s motion) or

other external (e. g. because of the Earth's atmosphere or other obscuring matter) effects. Such variability may provide a valuable addition to the usual fixed intensity and spectral information on the source.

Time-dependent signals that are measured in lab experiments and astronomical observations are usually recorded as digital *time series* data. Such data are also provided when a DS is numerically solved by a computer program. Time series analysis is, in general, a non-trivial process. In particular, this is so if the signal is contaminated by observational noise and the amount of data is limited, as is the case in a typical astronomical observation. Traditional techniques, like Fourier analysis, work well when the signal is periodic or multi-periodic, but they fail if it is irregular. Irregular variability may result from a random process, but it can also be due to deterministic chaos.

The questions then arise how to distinguish between a deterministically chaotic signal and a random one, and what is the relevant information that can be extracted from the former. Theoretical and practical issues related to this problem have indeed been addressed in parallel to the developments in chaos theory and have yielded significant understanding and a number of effective algorithms (see Abarbanel et al. [1]). The central idea, based on the pioneering work of David Ruelle, is based on a reconstruction of the essential dynamics of the DS which produces the signal, by embedding it in a multidimensional (pseudo) state space. The dimension of the set on which this dynamics takes place can then be found, using an algorithm of the kind that was originally proposed by Grassberger and Procaccia [24]. Low-dimensional fractal dimension is indicative of deterministic chaos.

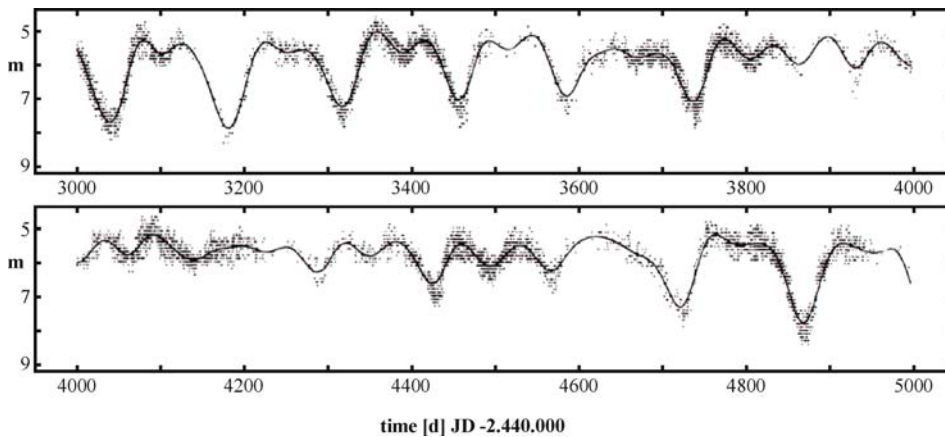
A number of attempts have been made to discover chaos in aperiodic time series arising from photometric observation of several types of astronomical objects. While quite reliable results have been achieved for some pulsating stars, the situation has been less satisfactory for accreting systems (X-ray binaries and active galactic nuclei). Norris and Matilsky [47] pointed out some of the pitfalls of a straightforward application of the Grassberger–Procaccia method on the reconstructed state space. They and Scargle [53], in a more general work, have reached the conclusion that an adequate predictive analysis (like a model or mapping) is most probably indispensable in the studies of irregular astronomical series which suffer from an insufficient amount of data and a relatively poor signal-to-noise ratio. Further general caveats and in particular those relevant to studies of active galactic nuclei variability were given by Vio et al. [60].

The most convincing evidence of low-dimensional dynamics and chaos in an astronomical signal was given by

Buchler et al. [11]. They proposed and used an appropriate discrete global mapping in the reconstructed pseudo state-space, which allowed them to use enough data points to faithfully determine the dimension of the attractor present in the equivalent dissipative DS, arising from the properly reduced data set. The aperiodic pulsating star R Scuti, whose light-curve had been followed for over 30 years, mainly by amateur observers (see Fig. 6), was analyzed first, giving a positive Liapunov exponent and a low fractal dimension ($D = 3.1$) of the attractor. In a subsequent work Buchler, Kolláth and Cadmus [12] presented analyzes of the light curves of several additional large-amplitude, irregularly pulsating stars (astronomically classified as *semi-regular*). They eliminated multi-periodicity and stochasticity as possible causes of the irregular behavior, leaving low-dimensional chaos as the only viable alternative. They then used their reconstruction technique in an attempt to extract quantitative information from the light curves and to uncover common physical features in this class of irregular variable stars. Similarly to the case of R Sct discussed above, it was found that the dynamics of at least three stars (out of the four examined) takes place in a four-dimensional dynamical state space, suggesting that two vibrational modes are involved in the pulsation. The physical nature of the irregular light curves was identified as resulting from a resonant mechanism with the first overtone mode, having a frequency close to twice that of the fundamental one. If the fundamental mode is self-excited while the resonant one is by itself stable, chaotic behavior can be expected on the basis of what is known as the Shilnikov scenario.

Convincing detection of chaotic behavior in signals from variable accreting systems has not, as yet, been definitely found. Contamination by observational, and possible also inherent, stochastic noise and an insufficient amount of data constitute the main challenges in this case.

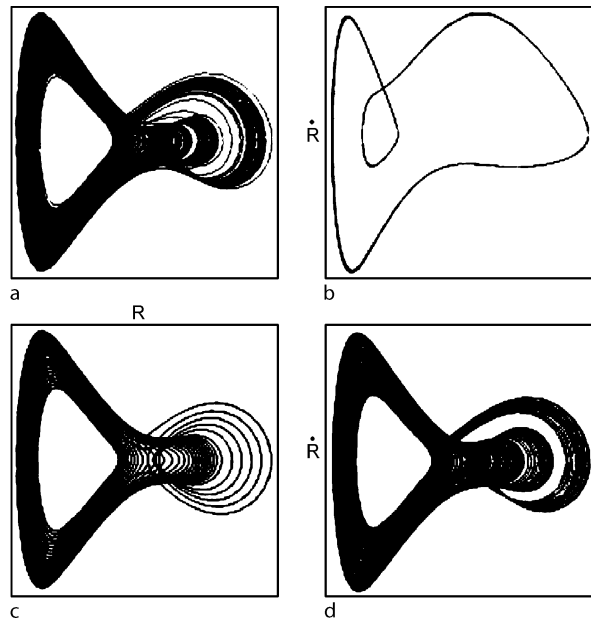
On the theoretical side, some simplistic and idiosyncratic models, as well as quite sophisticated methodical approaches, have been proposed to account for chaotic behavior in time-variable astrophysical systems. Buchler and Regev [10] devised a one-zone pulsator stellar model, in which ionization and radiative-transfer were incorporated. Such simplistic models had been quite effective before in elucidating the mechanisms responsible for pulsational instability in stars. The model was of an extended ionization zone atop an inert stellar core, the oscillation arising from possible dynamical instability due to partial ionization. At the same time the zone was driven thermally by the radiation diffusing through the envelope, from the core outward. When the dynamical time was significantly different from the thermal time, the model exhibited pe-



Astrophysics, Chaos and Complexity in, Figure 6

Typical observed light-curve segments of the irregularly pulsating star R Scuti. The raw data points are indicated by the *dots* and the smoothed filtered signal by the *continuous line*. Reproduced, with permission, from [11]

riodic oscillations. However if these two time scales were set to be comparable, Buchler and Regev found chaotic behavior. The mathematical structure of this model was quite similar to the simplistic model of thermal convection, in which Moore and Spiegel [44] had found aperiodic variability, proposing also (together with Baker, whose one-zone model was instrumental in understanding pulsational instability in general) that a similar model can be used to get irregular stellar pulsation. In Fig. 7 a projection of the system's trajectory in its three-dimensional state-space on the $R-\dot{R}$ (outer radius–outer velocity) plane is shown for four cases. In cases (b) and (c) there is chaotic behavior. The layered structure in those cases is reminiscent of chaotic attractors in typical DS. Using some standard criteria, Buchler and Regev have shown that this variability has indeed the properties of deterministic chaos. Other models of this kind have also discovered chaotic behavior, resulting from other mechanisms – e. g. a nonlinear non-adiabatic oscillator, but without dynamical instability and relaxation oscillations; a dynamically driven (by the inner core) Hamiltonian oscillator, etc. It is difficult to decide which of the above mechanisms, if any, is relevant to a realistic stellar model or indeed a pulsating star. Detailed numerical modeling of pulsating stars have, however, also shown transition to chaotic behavior via a sequence of period-doubling bifurcations, when a parameter (in this case the effective temperature) was varied. It has been suggested by Buchler and his collaborators that the transition to chaos is associated, in this case, with the excitation of vibrational overtones, brought about by resonances.



Astrophysics, Chaos and Complexity in, Figure 7

Projections of the attractor on the $R - \dot{R}$ plane in the model of Buchler and Regev [10]. In cases a and d the oscillations are periodic while in b and c there is chaos. Reproduced, with permission, from [10]

A systematic approach to stellar pulsation, based on DS theory has also been put forward. It employs the idea of dimensional reduction (using either perturbative analysis or a projection technique). The use of such approaches is quite widespread, in particular when the original DS is

formulated in terms of rather complicated PDE, as is the case in stellar pulsation theory. The idea is to capture the essentials of the dynamics in the form of a simpler mathematical system like a model PDE or a set of ODE – *amplitude equations*, or even IM. This is usually feasible close to the instability threshold, where the existence of a relatively low-dimensional *center manifold* to which the dynamics of the dissipative system is attracted, is guaranteed. Spiegel [56] and Buchler [8] discussed the possible techniques which can be applicable to stellar pulsation and Goupil and Buchler [23] summarized their work on amplitude equations, by giving their formal form for a rather general (non-radial, non-adiabatic) case. By studying the relevant amplitude equations in the various cases, it is possible to identify the nature of the possible mode couplings that are expected to occur in the nonlinear regime. In particular, the excitation of multi-mode pulsation and the effect of resonances can be investigated and then applied to phenomenological findings on pulsating stars (cf. Buchler [9]). For example, the amplitude equations, derived in this manner for a case in which two, almost resonant (having frequencies (Ω_1 and $\Omega_2 \approx 2\Omega_1$) radial modes are taken into account are

$$\begin{aligned} \frac{dA_1}{dt} &= \kappa_1 A_1 + i \frac{\Omega \Delta}{2} A_1 + P_1 A_1^* A_2 \\ \frac{dA_2}{dt} &= \kappa_2 A_2 - i \Omega \Delta A_2 + P_2 A_2^2, \end{aligned} \quad (5)$$

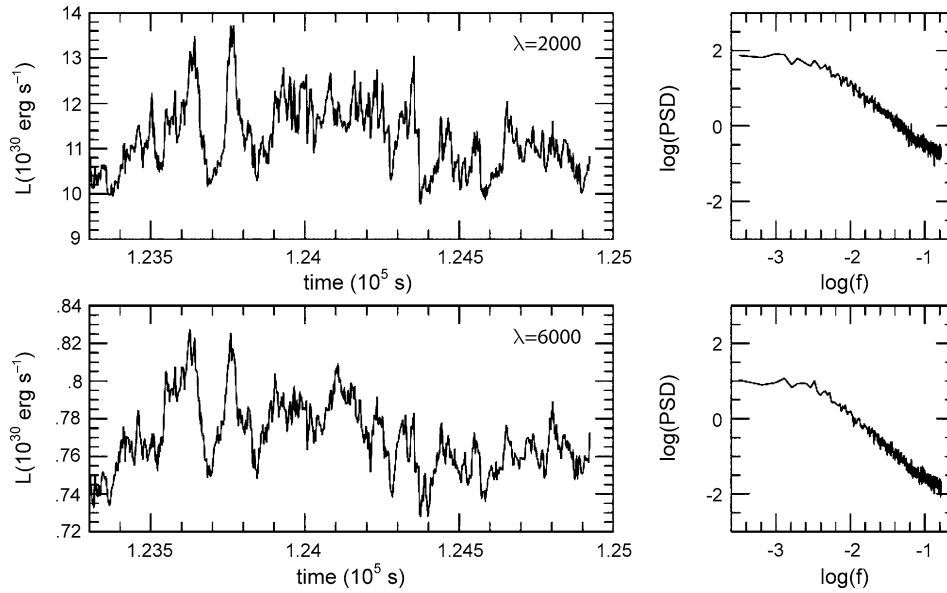
where A_i are the complex amplitudes of the modes, κ_i their linear growth-rates, $\Omega \equiv (2\Omega_1 + \Omega_2)/2$, $\Delta \equiv (2\Omega_1 + \Omega_2)/(2\Omega)$ and P_i are dependent on the specific model and can be derived from its structure. These equations capture the essential *slow* dynamics for the weakly nonlinear case.

Amplitude equations derived by perturbative techniques based on the separation of time-scales (as above) are not adequate for the case in which chaotic pulsation occurs with the dynamical and thermal timescales being comparable. However, dimensional reduction is still possible in this case, but more sophisticated approaches, e. g., the one suggested by Spiegel [56], are called for.

In the case of variable accreting sources several simplistic models, based on DS that can give rise to complex time-variability, were also proposed. Celnikier [13] pointed out that the essential features of the, so called, rapid X-ray burster, can be reproduced by a deterministic simple nonlinear equation, of the kind used in population dynamics studies. The IM employed was similar to the one used by Feigenbaum in his pioneering study, which demonstrated the universal period-doubling bifurcation route to chaos. The rapid burster belongs to a class

of variable X-ray sources, thought to originate in a close binary system in which a normal star is transferring mass onto its companion neutron star, giving rise to thermonuclear flashes on the latter. A simplistic two-zone model that captures some main characteristics of regular X-ray bursters has been proposed by Regev and Livio [51] and such models are still popular nowadays (e. g. [17]). For the rapid burster Livio and Regev [39] extended the model so as to include a third degree of freedom, thus providing the possibility of positive and negative feedback via mass accretion, and obtained a chaotic series of bursts, whose characteristics were very similar to the rapid burster.

Accreting close binary systems are known to exhibit irregularities in their light-curves in a number of spectral windows. Among them are quasi-periodic oscillations and short-time flickering. Young and Scargle [66] proposed a model, in this context, based on a coupled map lattice – an IM in which the value of a variable is dependent on the values of this variable in the previous iteration step, not only at the same location (lattice point), but also on this value at adjacent spatial locations. For some range of the parameters, quasi-periodic oscillations similar to the ones observed in several accreting binaries, were found. Young and Scargle found that persistent positive value of the largest Liapunov exponent cannot be guaranteed in this case, however chaotic behavior can be guaranteed during long transients. Such behavior was given the name *transient chaos*. Another type of a DS, a *cellular automaton* (essentially a coupled map lattice, but also with the dynamical variable allowed to take discrete values only) was proposed by Yonehara, Mineshige and Welsh [65] to model an accretion flow through a disk in close binary systems. The DS they invoked was in general found to evolve towards a characteristic state, whose qualitative behavior – mass accumulation during an extended time interval followed by rapid avalanches – resembled other well-studied DS. In most cases the avalanches are localized and quite small and terminate rapidly, but occasionally a local avalanche may trigger some very prominent events. This generic behavior has been found to occur in a variety of open dissipative systems and the state in which it occurs was called by Bak, Tang and Wiesenfeld [4] *self-organized criticality*. Such states generally do not have positive Liapunov exponents and are thus not truly chaotic, but they clearly exhibit scaling behavior. The most famous of these systems is the sand pile model, reminding some perhaps also of the behavior of stock markets. In Fig. 8 the flickering light-curves of a *cataclysmic variable* in two different wavelengths are shown along with the power spectral density (PSD) as calculated by Yonehara et al. for one particular choice of the parameters.



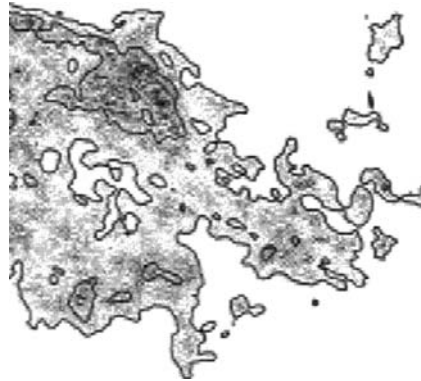
Astrophysics, Chaos and Complexity in, Figure 8

Light-curves and power spectral densities produced by the cellular automaton model of Yonehara et al. [65] for the flickering of Cataclysmic Variables. The *upper panels* correspond to 2000 Å emission, while the *lower ones* to emission at 6000 Å. Reproduced, with permission, from [65]

Spatio-Temporal Patterns and Complexity in Extended Systems

Extended astrophysical systems, that is, those that can be spatially resolved in observations often display very complex structures. The distribution of scattered matter in galaxies, the interstellar medium (ISM), is far from being uniform. Observations in a variety of wavelengths indicate that clouds of different densities are common and exhibit very complicated spatial forms (see Fig. 9 for one example). Observable phenomena in more compact objects than interstellar clouds, e.g. accretion disks of different types and stars (which are usually not spatially resolvable in direct observation), often require consideration of spatial processes as well. The spatial distribution of the stars themselves (seen as point sources) appears in the observations to be highly non-uniform. Hierarchical clustering, from star clusters of varying richness through galaxies and galactic clusters and up to the largest super-clusters of galaxies, is apparent on a variety of spatial scales, up to that of the entire visible universe. It is of interest to quantify the observational data and identify the physical processes responsible for shaping the various structures.

The time evolution of spatially extended astrophysical objects is naturally modeled by PDE. These DS are often very complex and include the equations of (magneto)fluid dynamics and also of thermal and radiative processes. At-



Astrophysics, Chaos and Complexity in, Figure 9

Contours of infra-red radiation intensity in some clouds observed by IRAS (detail). Adapted, by permission, from [6]

tempts to actually solve such nonlinear PDE (usually referred to as *microscopic equations*) can only be made by using numerical simulation. Despite the recent significant progress in computer power and visualization methods, such numerical calculations are often still inadequate, in particular because the Reynolds numbers of typical astrophysical flows are so high, such that a faithful resolution of the whole dynamical range is impossible. Modern approaches to spatio-temporal complexity utilize mathematical techniques to reduce the usually formidable mi-

croscopic PDE to significantly simpler and analytically (or semi-analytically) tractable DS (simple generic *model* PDE, ODE and IM). These approaches constitute what is now called *theory of patterns and complexity* and have so far been successfully employed in the theoretical study of laboratory and atmospheric fluid dynamical phenomena, chemical reactor systems, as well as in other diverse applications (see Cross and Hohenberg [18] for a review).

Physical processes that are instrumental in shaping the ISM complexity undoubtedly include fluid turbulence, which unfortunately still defies adequate understanding. Dynamical system approaches to turbulence have yielded some success, but only for rather limited simple issues, like transition (e. g. Bohr et al. [7]). In an effort to identify simpler physical mechanisms inducing ISM complexity, Elphick, Regev and Spiegel [21] began to examine the effects that thermal bi-stability (alone) may have on an extended system like the ISM. They used a one-dimensional simplified version for the relevant non-dimensional microscopic equation,

$$\partial_t Z = F(Z; p) + Z^\beta \partial_x^2 Z, \quad (6)$$

where the dependent variable $Z(x, t)$ is a positive power of the temperature, β is determined from the temperature dependence of the heat-diffusion coefficient and $F(Z; p)$ is the relevant cooling function with the pressure p considered to be a fixed parameter. Approximating $F(Z; p)$ by a suitable polynomial, they were able to reduce the problem to a set of ODE for the motion of fronts (between the two stable phases). In pattern theory this kind of approach is called *defect* dynamics. Elphick et al. found that an initially random distribution of front/anti-front pairs (which can be regarded as “clouds”) develops by front – anti-front annihilation and gives rise to an inverse cascade of larger and larger clouds, on the way to thermal phase separation. This process is similar to what is known in condensed matter physics as spinodal decomposition or roughening. When the simplest spatial forcing (as it should exist in the ISM, due to the non-uniform distribution of the heating sources) was introduced, possible complex steady states were found. Such states can be described by the following pattern map

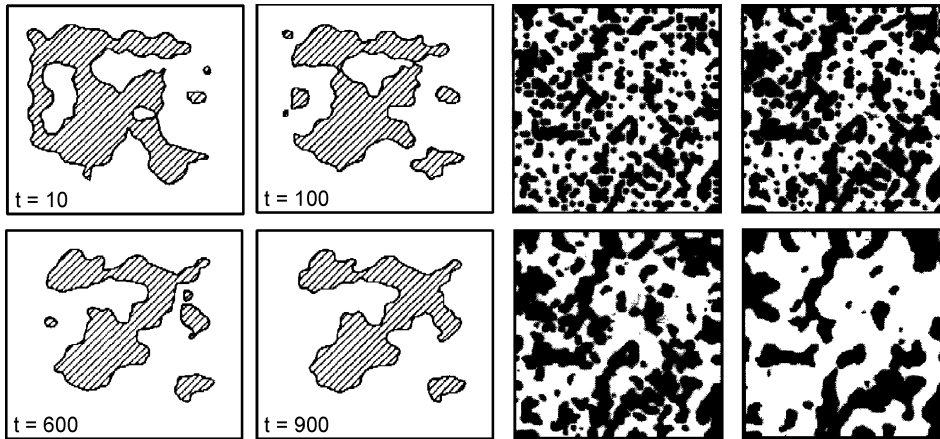
$$\begin{aligned} I_{j+1} &= I_j - K \sin \Theta_j \\ \Theta_{j+1} &= \Theta_j + L \log I_j, \end{aligned} \quad (7)$$

where Θ and I are simple functions of the front position and separation (cloud size), respectively. The constants K and L are dependent on the system specifics. This IM bears resemblance to the standard map (Eq. 1) and similarly to

the latter is chaotic. This means that thermal instability (and some forcing) alone can produce spatial complexity.

In subsequent studies Aharonson, Regev and Shaviv [3] used a multi-dimensional extension of Eq. 6 and solved it numerically in two dimensions, starting from a random distribution. It is well known in pattern theory that front curvature is driving evolution of structures in systems governed by reaction-diffusion equations, like Eq. 6, and indeed the system advanced relatively fast towards phase-separation. However, the inclusion of simple and quite slight spatio-temporal forcing atop the cooling function resulted in a final locked steady pattern with complex cloud boundaries (see the left panels of Fig. 10). Shaviv and Regev [54] included also fluid motion to account for mass conservation through the moving fronts, but this did not change the result qualitatively (the right panels of Fig. 10). Calculation of the cloud boundaries’ fractal dimension in the final state of the above numerical simulations gave a result that was similar to the one found in observations by Bazel and Désert [6]; Falgarone, Philips and Walker [22] and others. These studies analyzed observational data in a number of wavelengths and reported fractal dimensions $1.2 \lesssim D \lesssim 1.5$ for the various cloud boundaries. These observational claims have, however, remained quite controversial and several researchers have claimed that fractal structures can not be actually faithfully inferred from such observations. Still, the fact that thermal instability and diffusive spatial coupling alone can give rise to complicated cloud forms may be important in ongoing efforts to understand ISM complexity.

Ideas from pattern theory can also be used for astrophysical systems governed by microscopic equations that are considerably more complicated than those describing just a thermally bi-stable diffusive medium. Among the first paradigms of deterministic chaos were the dissipative ODE systems of Lorentz, Moore and Spiegel, derived by some drastic simplifications of the equations of fluid dynamics in order to model thermal convection. More judicious reduction procedures have been developed since then and successfully applied to different spatio-temporal systems. In addition to the mathematical techniques that allow one to extract the dynamics of coherent structures (defects) in spatio-temporal systems, when such structures do indeed exist, there are a number of other methods for the reduction of the original PDE system. An approach of this kind, which works well for a system close to its instability threshold, was recently applied, by Umurhan, Menou and Regev [58], to the magneto-rotational instability which is thought to be of primary importance in accretion disks. Employing a perturbative asymptotic analysis, valid close to the instability threshold



Astrophysics, Chaos and Complexity in, Figure 10

Evolution of cloud patterns in a reaction-diffusion system of the kind (6) but in 2D. The *left panels* show the purely thermal case and the *right one* display the effect of the inclusion of fluid motion. The *rightmost low picture* shows the ultimate state that persists due to forcing. Reproduced, with permission from [3] and [54]

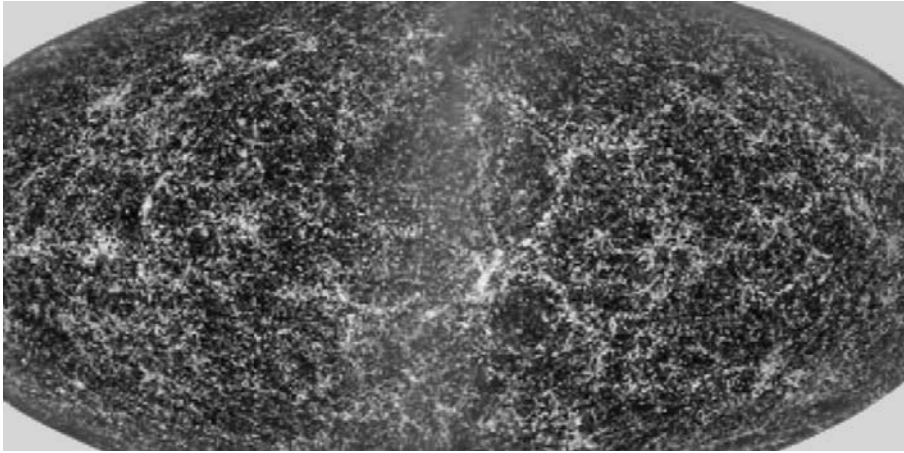
and for a weakly nonlinear case, Umurhan et al. reduced the full microscopic magneto-hydrodynamical equations to a well-known, generic PDE for the spatio-temporal amplitude of a disturbance – the real Ginzburg–Landau equation. Using then-known solutions of this equation, one could estimate various important flow properties as the instability develops and nonlinearly saturates.

The dynamo problem (i. e., the creation of magnetic fields in flows of ionized fluids) is of obvious astrophysical interest and it too can be approached, in addition to numerical calculations, by methods of nonlinear DS theory. The work of Balmforth et al. [5], in which the advection of vector fields by chaotic flows was investigated, is an example of such an approach. Childress and Gilbert [14] summarize a number of ideas along these lines, that can be fruitful for future research on the dynamo problem.

The nature of the distribution of stars in the universe, with its hierarchical clustering, has raised some important questions. Virtually all the viable cosmological models are derived from Einstein’s general theory of relativity and are based on the *cosmological principle*, which asserts that the universe is homogeneous and isotropic. However, direct observations of *luminous* matter have always revealed a substantial degree of clumpiness. Straightforward examination of the pictures reveals galaxy groups, clusters, super-clusters, voids, filaments and walls (see Fig. 11). Hierarchical clustering is typical of fractal sets and thus the question of whether the large scale distribution of matter in the universe is fractal or homogeneous has naturally been raised. Wu, Lahav and Rees [64] have recently compiled observational surveys of the galaxy dis-

tribution and concluded that relevant statistical measures, which enable the determination of dimension, point to the fact that the luminous matter distribution’s dimension is scale dependent. On relatively small scales, less than about $50 h^{-1}$ Mpc (h is the Hubble constant in units of 100 km/s/Mpc), the distribution is indeed undoubtedly fractal. It is the smoothness of the galaxy distribution (or lack thereof) on the largest scales that is still the key point in the debate between the proponents of a “fractal universe” and the more traditional cosmologists.

The cosmic microwave background (CMB) radiation, for which years of data (since the launch of the COBE satellite) are now available, can provide a measure of the spatial fluctuations in the universe on the grandest scales – of the order of $\sim 1000 h^{-1}$ Mpc. Fairly recent results indicate that the dimension of the mass distribution at these scales is not different from the value of three by more than a few tens of 10^{-6} , indicating that the universe is quite smooth on the largest scale. Various attempts to evaluate the mass distribution in the universe are based on the observation of different entities, from luminous matter (galaxies) and up to the CMB (the universe at a very early epoch, when galaxies were still absent). Some very recent optical surveys aim at an impressively deep coverage (up to $\sim 300\text{--}400 h^{-1}$ Mpc) but the grandest scales can be probed, at the present time, only by other means. Modern observations in radio, microwave, infra-red and X-ray bands also enabled a deeper look, well beyond the existing and upcoming deep optical surveys. Some recent radio surveys and X-ray background (XRB) data can probably probe scales of up to $\sim 500 h^{-1}$ Mpc and thus bridge the



Astrophysics, Chaos and Complexity in, Figure 11

An image of close to one million of the brightest galaxies detected by the Two Micron All Sky Survey -2MASS (detail). The non-uniform distribution with clusters, voids and filaments, is clearly visible. The blurred strip is due to the Galactic plane [59].

gap between the optical surveys and the largest CMB scale. Other probing tools, like the Lyman α forest, may also be valuable in this respect.

The question of how (and at what scale) the transition from a fractal distribution to a smooth one occur, is still quite open. There seems even to be no agreement on the question of what the relevant statistical measures are to find the scale this transition. The recent suggestion by Pietronero, Gabrielli and Sylos Labini [48] to define it as the crossover scale between power-law behavior and exponential decay of the correlation function appears to be the most meaningful.

Theoretical efforts, aimed at finding and understanding the physical mechanism behind the above results on the spatial cosmic distributions, have also been made quite recently. A model for nonlinear gravitational clustering was proposed by Provenzale et al. [49], who introduced a simple, dynamically motivated family of multi-fractals (sets whose fractal dimension changes with scale), which are well suited for a bottom-up scenario of gravitational evolution. Murante et al. [45] suggested that a galaxy distribution whose dimension varies with scale may naturally result from the presence of a number of very strong local density maxima. They randomly superimposed a few tens of singularities and let them computationally generate point sets, whose dimension could be estimated as a function of scale. A result resembling real galaxy data, on small enough scales, was obtained for sufficiently strong singularities. Provenzale, Spiegel and Thieberger [50] suggested that additional statistical measures of the galaxy distribution may be useful for a meaningful characterization and perhaps deeper understanding of the scaling behavior at

intermediate scales and found that lacunarity, a well-defined mathematical property found in multi-fractals and giving rise to oscillations around the mean value of their generalized dimensions, may have important astrophysical consequences.

Future Directions

Astronomy counted among the main disciplines in which the first studies on chaos and complexity arose, in both Hamiltonian and dissipative nonlinear DS. The most significant mathematical developments in the theory took place in the 20th century, mainly in its second half, and the efforts to extend it have continued until the present time. Detailed applications to particular astrophysical systems have, however, always been marred by the difficulty of obtaining a sufficient amount of high-quality data. Carefully controlled experiments, which are possible in the laboratory, are impossible when cosmic phenomena are merely passively observed.

The ongoing fast development of modern means for astronomical data acquisition – sophisticated telescopes, both earthbound and on dedicated satellites, increasingly powerful computers and new data reduction techniques – have recently opened new possibilities. On the theoretical side, intensive efforts are also being made to understand the behavior of the extremely complicated DS as those encountered in the study of astrophysical systems. Thus the prospects for fruitful and significant future developments appear to be very good.

Among the possible future directions, research on the following topics seems to be the most promising:

- Near the end of the 20th century, it was observationally confirmed that stars other than the Sun, some of them in binary systems, have orbiting objects around them that appear to be planets. In the last decade over 200 of such exo-planets have been discovered and there is growing evidence that chaos plays a role in both their formation as well as in the evolution of exo-planetary systems. Holman, Touma and Tremaine [30] studied such newly discovered exo-planets and concluded that their results imply that planetary orbits in binary stellar systems commonly experience periods of high eccentricity and dynamical chaos, and that such planets may occasionally collide with the primary star. Lissauer [38] pointed out the possible role of chaos in the formation of planets, in the Solar Systems and exo-planetary systems and Kiseleva-Eggleton et al. [32] found that within a few known exo-planetary systems at least one has chaotic orbits. The issue of planet migration, which acquired importance after the discovery of exo-planets, and the possible role of chaos in this process, has been addressed by Adams and Laughlin [2]. It appears, thus, that the inclusion of chaotic processes is indispensable in the important quest to understand the formation of planetary systems and their evolution. Such studies are expected to shed new light on the probability of finding exo-planets within “habitable” zones and to have implications for possible existence and survival of life in the universe.
- The most direct approach to the study of star cluster and galaxy dynamics is the one employing the GNPB with a large value of n . The direct problem can only be approached numerically and because chaos is bound to be present in this patently non-integrable Hamiltonian DS, it is important to assess its possible role in the simulations. This issue has not been fully resolved yet, but there are some recent findings which may elucidate outstanding questions and future directions in this context. For example, Kandrup and Sideris [31] addressed the difference between the behavior found in full n -body simulations and that of orbits in a smooth potential and Hayes [28] compared the results of n -body calculations with softened potential to the exact ones. Much remains to be done in this line of research, because the theoretical efforts towards understanding the formation of structure in the universe, on a large range of scales, heavily rely on n -body simulations. Recent work on galactic dynamics, employing self-consistent models, indicates that it is important to carefully examine the orbits, distinguishing between the regular and chaotic ones, and then try to understand what the role is that the chaotic orbits play in shaping the galaxy. The work of Voglis, Stavropoulos and Kalapotharakos [61], for example, yielded a surprising new result – that long living spiral arms are excited on the disk, composed almost completely of chaotic orbits. Other studies of this kind, led mainly by George Contopoulos and his collaborators and students, have already provided additional important results in this context. It is reasonable to expect that new understanding on the role of chaos in galactic dynamics as well as on ways to handle its influence on n -body simulations will emerge as a result of future studies. Such efforts may also contribute to new results in the theory of Hamiltonian DS of this sort.
- Astrophysical fluid dynamics is a broad subject, in which input from DS and pattern theory may provide significant progress in the future. Most existing studies of fluid systems in astrophysics have been based, in addition to linear stability analyzes, on very simplified models or brute-force numerical simulations. New knowledge and understanding of nonlinear processes that numerical simulations alone can not provide, can be expected, as it has been achieved in other fluid dynamical applications, if nonlinear DS and pattern theory is applied to astrophysical fluids as well. Ed Spiegel and his collaborators have advocated approaches of this kind for some time (e. g. Spiegel [55]) and interesting results have already been obtained. Selected examples are given in the book edited by Zahn and Zinn-Justin [67] and more recent such contributions have already found their way to the astrophysical and general fluid-dynamical literature. They mark a promising new direction in the theoretical studies of important, but not yet fully understood, astrophysical (magneto)fluid-dynamical processes as accretion, convection, explosions, jets and so forth.

Bibliography

Primary Literature

1. Abarbanel HDI, Brown R, Sidorovich JJ, Tsimring SS (1993) The analysis of observed chaotic data in physical systems. *Rev Mod Phys* 65:1331–1392
2. Adams FC, Laughlin G (2003) Migration and dynamical relaxation in crowded systems of giant planets. *Icarus* 163:290–306
3. Aharonson V, Regev O, Shaviv N (1994) Pattern evolution in thermally bistable media. *Astrophys J* 426:621–628
4. Bak P, Tang C, Wiesenfeld K (1987) Self-organized criticality – An explanation of $1/f$ noise. *Phys Rev Lett* 59:381–384
5. Balmforth NJ, Cvitanović P, Ierley GR, Spiegel EA, Vattay G (1993) *Ann NY Acad Sci* 706:148–154
6. Bazell D, Désert FX (1988) Fractal structure of interstellar cirrus. *Astrophys J* 333:312–319

7. Bohr T, Jensen MH, Paladin G, Vulpiani M (1998) Dynamical system approach to turbulence. Cambridge University Press, Cambridge
8. Buchler JR (1993) A dynamical systems approach to nonlinear stellar pulsations. *Astrophys Space Sci* 210:9–31
9. Buchler JR (1998) Nonlinear pulsations. *ASP Conference Series* 135:220–230
10. Buchler JR, Regev O (1982) Oscillations of an extended ionization region in a star. *Astrophys J* 263:312–319
11. Buchler JR, Serre T, Kolláth Z, Mattei J (1996) Nonlinear analysis of the light curve of the variable star R Scuti. *Astrophys J* 462:489–501
12. Buchler JR, Kolláth Z, Cadmus RR Jr (2004) Evidence for low-dimensional chaos in semiregular variable stars. *Astrophys J* 613:532–547
13. Celnikier LM (1977) A simple mathematical simulacrum of the X-ray burster phenomenon. *Astron Astrophys* 60:421–423
14. Childress S, Gilbert AD (1995) Stretch, twist and fold: The fast dynamo. Springer, New York
15. Chirikov RV, Vecheslavov VV (1989) Chaotic dynamics of Comet Halley. *Astron Astrophys* 221:146–154
16. Contopoulos G (2002) Order and chaos in dynamical astronomy. Springer, New York
17. Cooper RL, Narayan R (2006) A two-zone model for type I X-ray bursts on accreting neutron stars. *Astrophys J* 652:584–596
18. Cross MC, Hohenberg PC (1993) Pattern formation outside equilibrium. *Rev Mod Phys* 65:851–1112
19. Duncan MJ, Quinn T (1993) The long-term dynamical evolution of the Solar System. *Ann Rev Astron Astrophys* 31:265–295
20. El-Zant AA (1997) On the stability of motion of N-body systems: A geometric approach. *Astron Astrophys* 326:113–129
21. Elphick C, Regev O, Spiegel EA (1991) Complexity from thermal instability. *Mon Not Roy Astr Soc* 250:617–628
22. Falgarone E, Phillips TG, Walker CK (1991) The edges of molecular clouds – Fractal boundaries and density structure. *Astrophys J* 378:186–201
23. Goupil M-J, Buchler JR (1994) Amplitude equations for nonadiabatic nonradial pulsators. *Astron Astrophys* 291:481–499
24. Grassberger P, Procaccia I (1983) Measuring the strangeness of strange attractors. *Physica* 9D:189–208
25. Gurzadyan VG, Savvidy GK (1986) Collective relaxation of stellar systems. *Astron Astrophys* 160:203–210
26. Habib S, Kandrup HE, Mahon ME (1997) Chaos and noise in galactic potentials. *Astrophys J* 480:155–166
27. Hasan H, Pfeniger C, Norman C (1993) Galactic bars with central mass concentrations – Three-dimensional dynamics. *Astrophys J* 409:91–109
28. Hayes WB (2003) Shadowing high-dimensional Hamiltonian systems: The gravitational N-body problem. *Phys Rev Lett* 90(1–4):054104
29. Hénon M, Heiles C (1964) The applicability of the third integral of motion: Some numerical experiments. *Astronomical J* 69:73–79
30. Holman M, Touma J, Tremaine S (1995) Chaotic variations in the eccentricity of the planet orbiting 16 Cygni B. *Nature* 386:254–256
31. Kandrup HE, Sideris IV (2003) Smooth potential chaos and N-body simulations. *Astrophys J* 585:244–249
32. Kiseleva-Eggleton L, Bois L, Rambaux N, Dvorak R (2002) Global dynamics and stability limits for planetary systems around HD 12661, HD 38529, HD 37124, and HD 160691. *Astrophys J* 578:L145–L148
33. Laskar J (1994) Large-scale chaos in the Solar System. *Astron Astrophys* 64:115–162
34. Laskar J (1996) Large scale chaos and marginal stability in the Solar System. *Celestial Mech Dynamical Astron Astrophys* 287:L9–L12
35. Laskar J, Robutel P (1993) The chaotic obliquity of planets. *Nature* 361:608–611
36. Laskar J, Joutel F, Robutel P (1993) Stabilization of the earth's obliquity by the moon. *Nature* 361:615–617
37. Lecar M, Franklin FA, Holman MJ, Murray NJ (2001) Chaos in the Solar System. *Ann Rev Astron Astrophys* 39:581–631
38. Lissauer JJ (1999) Chaotic motion in the Solar System. *Rev Mod Phys* 71:835–845
39. Livio M, Regev O (1985) X-ray burst sources: A model for chaotic behavior. *Astron Astrophys J* 148:133–137
40. Lorentz EN (1963) Deterministic nonperiodic flow. *J Atmos Sci* 20:130–141
41. Mardling RA (1995) The role of chaos in the circularization of tidal capture binaries – I. The chaos boundary. II. Long-time evolution. *Astrophys J* 450:722–731, 732–747
42. Merritt D, Fridman T (1996) Triaxial galaxies with cusps. *Astrophys J* 460:136–162
43. Merritt D, Valluri M (1996) Chaos and mixing in triaxial stellar systems. *Astrophys J* 471:82–105
44. Moore DW, Spiegel EA (1966) A thermally excited non-linear oscillator. *Astrophys J* 143:871–887
45. Murante G, Provenzale A, Spiegel EA, Thieberger R (1997) Density singularities and cosmic structures. *Mon Not Roy Astr Soc* 291:585–592
46. Murray N, Holman M (1999) The origin of chaos in the outer Solar System. *Science* 283:1877–1881
47. Norris N, Matilsky T (1989) Is Hercules X-1 a strange attractor? *Astrophys J* 346:912–918
48. Pietronero L, Gabrielli A, Sylos Labini F (2002) Statistical physics for cosmic structures. *Physica A* 306:395–401
49. Provenzale A, Galeotti P, Murante G, Villone B (1992) A simple multifractal model of nonlinear gravitational clustering. *Astrophys J* 401:455–460
50. Provenzale A, Spiegel EA, Thieberger R (1997) Cosmic lacunarity. *Chaos* 7:82–88
51. Regev O, Livio M (1984) Thermal cycles from a two-zone accreting model: X-ray bursts and shell flashes. *Astron Astrophys* 134:123–128
52. Sagdeev RZ, Zaslavsky GM (1987) Stochasticity in the Kepler problem and a model of possible dynamics of comets in the Oort cloud. *Nuovo Cimento B* 97:119–130
53. Scargle JD (1990) Studies in astronomical time series analysis IV – Modelling chaotic and random processes with linear filters. *Astrophys J* 359:469–482
54. Shaviv NJ, Regev O (1994) Interface dynamics and domain growth in thermally bistable fluids. *Phys Rev E* 50:2048–2056
55. Spiegel EA (1980) Fluid dynamical form of the linear and nonlinear Schrödinger equations. *Physica D* 1:236–240
56. Spiegel EA (1993) Patterns of aperiodic pulsation. *Astrophys Space Sci* 210:33–49
57. Sussman GJ, Wisdom J (1992) Chaotic evolution of the Solar System. *Science* 257:56–62
58. Umurhan M, Menou K, Regev O (2007) Weakly nonlinear anal-

ysis of the magnetorotational instability in a model channel flow. *Phys Rev Lett* 98(1–4):034501

59. University of Massachusetts and Infrared Processing and Analysis Center/California Institute of Technology (2003) Joint project: Two Micron All Sky Survey. <http://www.ipac.caltech.edu/2mass/releases/allsky>. Accessed 2008
60. Vio R, Cristiani S, Lessi O, Provenzale A (1992) Time series analysis in astronomy – An application to quasar variability studies. *Astrophys J* 391:518–530
61. Voglis N, Stavropoulos I, Kalapotharakos C (2006) Chaotic motion and spiral structure in self-consistent models of rotating galaxies. *Mon Not Roy Astr Soc* 372:901–922
62. Wisdom J (1982) The origin of the Kirkwood gaps – A mapping for asteroidal motion near the 3/1 commensurability. *Astron J* 87:577–593
63. Wisdom J, Peale SJ, Mignard F (1984) The chaotic rotation of Hyperion. *Icarus* 58:137–152
64. Wu KKS, Lahav O, Rees MJ (1999) The large-scale smoothness of the universe. *Nature* 397:225–230
65. Yonehara A, Mineshige S, Welsh WF (1997) Cellular-automaton model for flickering of cataclysmic variables. *Astrophys J* 486:388–396
66. Young K, Scargle JD (1996) The dripping handrail model: Transient chaos in accretion systems. *Astrophys J* 468:617–632
67. Zahn J-P, Zinn-Justin J (eds) (1993) *Astrophysical fluid dynamics* (Les-Houches – Session XLVII). North-Holland, Amsterdam

Books and Reviews

- Barnsley M (1998) *Fractals everywhere*. Academic, San Diego
- Heggie D, Hut P (2003) *The gravitational million-body problem: A multidisciplinary approach to star cluster dynamics*. Cambridge University Press, Cambridge
- Manneville P (2004) *Instability, chaos and turbulence*. Imperial College Press, London
- Pismen LM (2006) *Patterns and interfaces in dissipative dynamics*. Springer, Berlin
- Regev O (2006) *Chaos and complexity in astrophysics*. Cambridge University Press, Cambridge
- Shore SN (1992) *An introduction to astrophysical fluid dynamics*. Academic, San Diego
- Sussman GJ, Wisdom J (2001) *Structure and interpretation of classical mechanics*. MIT Press, Cambridge

Astrophysics: Dynamical Systems

GEORGE CONTOPOULOS

Academy of Athens, Research Center for Astronomy,
Athens, Greece

Article Outline

[Definition of the Subject](#)

[Introduction](#)

[Classification](#)

[Integrable Systems](#)

[Formal Integrals. KAM and Nekhoroshev Theory](#)

[Periodic Orbits](#)

[Transition from Order to Chaos](#)

[Dynamical Spectra](#)

[Application: Order and Chaos in Galaxies](#)

[Future Directions](#)

[Bibliography](#)

Definition of the Subject

Dynamical systems is a broad subject, where research has been very active in recent years. It deals with systems that change in time according to particular laws. Examples of such systems appear in astronomy and astrophysics, in cosmology, in various branches of physics and chemistry, and in all kinds of other applications, like meteorology, geodynamics, electronics and biology. In the present article we deal with the mathematical theory of dynamical systems from the point of view of dynamical astronomy. This theory is generic in the sense that its mathematics can be applied to diverse problems of physics and related sciences, ranging from elementary particles to cosmology.

Introduction

The theory of dynamical systems has close relations with astronomy and astrophysics, in particular with dynamical astronomy. Dynamical astronomy followed two quite different traditions until the middle of the 20th century, namely celestial mechanics and statistical mechanics.

Celestial mechanics was the prototype of order. The use of perturbation theories in the solar system was quite effective for predicting the motion of the planets, satellites and comets. Despite the fact that the perturbation series were in most cases extremely long, their effectiveness was never doubted. One of the most painstaking developments in this field was the “Theory of the Motion of the Moon” of Delaunay [29] and his successors. Today the theory of the Moon has been extended to an unprecedented accuracy by computer algebra, giving the position of the Moon with an accuracy of a few centimeters [8,30].

Celestial mechanics had proved its effectiveness already by the 19th century with the discovery of the planet Neptune by Leverrier and Adams. Its subsequent development reached its culmination in the work of Poincaré: “Méthodes Nouvelles de la Mécanique Céleste” [66]. But at the same time it reached its limits. Poincaré proved that most of the series of celestial mechanics are divergent. Therefore their accuracy is limited. On the other hand, it was made clear that the most important problem of celestial mechanics, the N -body problem, cannot be solved analytically. The work of Poincaré contains the basic ideas of most of the recent theories of dynamical systems, in particular what we call today the “Theory of Chaos”.

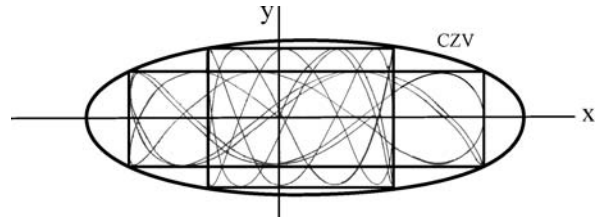
A completely different approach of the theory of dynamical systems was based on statistical mechanics. This approach dealt with the N -body problem, but from a quite different point of view. Instead of a dominant central body (the Sun) and many small bodies (planets), it dealt with N -bodies of equal masses (or of masses of the same order) with large N .

Statistical mechanics, developed by Boltzmann, Gibbs and others, dominated physics in general. In this approach the notion of individual orbits of particles became not only secondary, but even irrelevant. One might consider different types of orbits, but their statistical properties should be the same.

Nevertheless, a basic problem was present in the foundations of statistical mechanics. If the N -body systems are deterministic, how is it possible to derive the random properties of the ensembles of statistical mechanics? This question led to the development of a new theoretical approach that is called “ergodic” theory. If almost all orbits are ergodic, then the random behavior of the system can be proved, and not simply assumed. As an example, consider a room of gas of constant density. The constancy of the density may be considered as a probabilistic effect. But if the orbits of the molecules are ergodic, all particles stay on the average equal intervals of time in equal volumes inside this room. Thus, the constancy of density is due to the properties of the orbits of all particles.

In dynamical astronomy there was a complete dichotomy between celestial mechanics and stellar dynamics. Celestial mechanics continued its work with perturbation series, and no chaos at all, while the new branch of stellar dynamics (and galactic dynamics in particular) was completely influenced by statistical considerations, as in statistical mechanics. In particular, one dealt with distribution functions of stellar velocities that were well described by a velocity ellipsoid. But it was realized that the time required for the establishment of a statistical equilibrium was much longer than the age of the universe. Thus the stars had not enough time to settle into a statistical equilibrium. These problems led to a reconsideration of the foundations of galactic dynamics, namely to the study of the individual orbits of the stars. Such studies started only in the sixties. The first time when celestial mechanicians and galactic astronomers met at a common meeting was during the 1964 Thessaloniki Symposium on the “Theory of Orbits in the Solar System and in Stellar Systems” [16]. At that time a completely new tool was available in dynamical astronomy, namely the use of computers.

The first computer experiment with a N -body system was the so-called Fermi–Pasta–Ulam paradox, in 1955. These authors considered particles along a line attracted



Astrophysics: Dynamical Systems, Figure 1

The first published orbits (1958) in the meridian plane of an axisymmetric galaxy inside the curve of zero velocity (CZV = equipotential)

by linear forces, plus a small nonlinear force. The energy was initially given to a few modes. The energy of each mode cannot change if the forces are only linear. But because of the non linearities, the energies of all the modes of the system were expected to change continually, in a way consistent with a statistical mechanics prediction. Instead of that, the computer results have shown that the energy changes were almost periodic and no tendency to a final statistical situation ever appeared. This was a paradox, a great surprise for people grown up with statistical mechanics.

A little later (1956), a computer was used by Per-Olof Lindblad to calculate orbits of stars in a plane galaxy, in order to find the formation of spiral arms [51].

At about the same time, we calculated [13] two orbits of stars in three dimensions. Based on the prevalent assumptions of statistical mechanics, we expected these orbits to be ergodic and fill all the space inside the energy surface. Instead, we found that the orbits did not fill all the available space, but filled only curvilinear parallelograms, like deformed Lissajous figures (Fig. 1). Later it was shown that such orbits can be explained, qualitatively and quantitatively, by a formal third integral of motion [14].

The theory of the third integral goes back to the work of Birkhoff [5] and Whittaker [77,78]. In particular Whittaker found the so-called adelpic integral, i. e. an integral similar to the energy in simple dynamical systems. In the case of two harmonic oscillators this integral is reduced to the energy of one oscillator only. In more general cases higher order terms have to be added to find a constant of motion. However, in resonant cases the form of the third integral may be quite different. For example, in the case of two equal frequencies it is a generalization of the angular momentum.

The series giving the third integral are in general divergent [70]. But even so in numerical applications the third integral is better conserved if it is truncated at higher and higher orders, up to a certain maximum order. The usefulness of these series was emphasized by Moser [61].

The most important result of these studies was that, in general, dynamical systems are neither integrable nor ergodic. This was a surprise, because it was generally assumed that generic dynamical systems are either integrable or ergodic [48]. This change of paradigm was emphasized by Lynden Bell [54].

The existence of ordered domains in generic dynamical systems is the basic content of the famous Kolmogorov–Arnold–Moser (KAM) theorem. Kolmogorov [47] announced a theorem proving the existence of invariant tori in dynamical systems. Such tori contain quasi-periodic motions with frequencies whose ratios are far from all rationals (see Sect. “[Formal Integrals. KAM and Nekhoroshev Theory](#)”).

The details of the proof were given by Arnold [2,3] in the analytical case, and independently by Moser [59,60] in sufficiently differentiable cases (with 333 derivatives!). More recently the existence of such tori was proven for systems that are differentiable only a few times. Furthermore, invariant tori appear generically near stable periodic orbits. Such orbits appear even in systems with arbitrarily large perturbations. Thus complete lack of ordered regions containing invariant tori is rather exceptional.

Classification

There are two main types of dynamical systems:

- (1) Maps (or mappings) of the form $\vec{x}_{n+1} = \vec{f}(\vec{x}_n)$, where \vec{x}_n is a vector of N -dimensions and \vec{f} is a set of N -functions.
- (2) Systems of differential equations of the form $\dot{\vec{x}} = \vec{f}(\vec{x}, t)$, where \vec{x} is a N -dimensional vector, and the dot denotes derivative with respect to a continuous time t .

Another separation of dynamical systems, both of maps and differential equations, is in conservative and dissipative systems. Conservative systems preserve the volume in phase space, while in dissipative systems the volume decreases on the average. If we reverse the time direction we have systems with increasing volume.

A large class of systems of differential equations are the Hamiltonian systems, with conjugate variables $\vec{x}(x_1, x_2, \dots, x_N)$ and $\vec{y}(y_1, y_2, \dots, y_N)$ that satisfy equations of the form:

$$\dot{\vec{x}} = \frac{\partial H}{\partial \vec{y}}, \quad \dot{\vec{y}} = -\frac{\partial H}{\partial \vec{x}} \quad (1)$$

where H is the Hamiltonian function $H = H(\vec{x}, \vec{y}, t)$ and N is the number of degrees of freedom. The equations of motion (1) are called canonical, or Hamiltonian equations. The space of the variables \vec{x} and \vec{y} is called phase

space. The phase space of a map and of a system of differential equations is N -dimensional, while the phase space of a Hamiltonian is $2N$ -dimensional. A change of variables $(\vec{x}, \vec{y}) \rightarrow (\vec{x}', \vec{y}')$ is called canonical if the equations of motion in the new variables are also canonical.

The most important separation of dynamical systems is between integrable systems and chaotic systems.

The most simple integrable systems are the solvable systems, i. e. systems that can be solved explicitly, to give the variables as functions of time. This definition is too restricted, if by functions we mean known functions of time. A more general definition is in terms of single-valued functions of time [79], even if such functions can be given only numerically.

A more general definition of integrable systems is systems that have N independent analytic integrals of motion in involution $I_i(\vec{x}, t) = I_i(\vec{x}_0, t_0)$, $i = 1, 2, \dots, N$. (By involution we mean that any two such integrals have a zero Poisson bracket (see Eq. 6)).

As regards chaotic systems, their definition and classification is a difficult task. The basic property of chaos is sensitive dependence on initial conditions. Namely two orbits of a compact system starting very close to each other deviate considerably later on (exponentially fast in time).

A particular class of chaotic systems are the ergodic systems, in which the orbits go everywhere on a surface of constant energy. Namely, the orbits pass through the *neighborhood* of every point of the energy surface. In an ergodic system the time average of a function is equal to its phase average, according to Birkhoff's theorem [6]. An ergodic system is called “mixing” if two nearby particles on different (ergodic) orbits can go very far from each other.

A further distinction of mixing systems is related to the speed of deviation of nearby orbits. If the deviation is exponential in time the system is called “Kolmogorov” or K-system. The deviation $\vec{\xi}$ of nearby orbits that are initially (at time t_0) at an infinitesimal distance $\vec{\xi}_0$ is measured by the Lyapunov characteristic number (LCN) (or simply the Lyapunov exponent).

For almost all deviations $\vec{\xi}_0$ from the initial condition \vec{x}_0 of an orbit the Lyapunov characteristic number is the same, equal to

$$\text{LCN} = \lim_{t \rightarrow \infty} \frac{\ln |\vec{\xi}/\vec{\xi}_0|}{t} \quad (2)$$

In a system of N degrees of freedom one can define N Lyapunov characteristic numbers. If not specified explicitly otherwise the term Lyapunov characteristic number means the maximal LCN. A system is said to have sensitive dependence on the initial conditions if its LCN is positive.

A simple system with positive LCN is given by $\xi = \xi_0 \exp(qt)$. In this system $\ln(\xi/\xi_0) = qt$, therefore $\text{LCN} = q$ (constant). A system in which the deviation ξ increases linearly in time (or as a power of time) is not Kolmogorov although it is mixing. For example, if $\xi = \xi_0 + \alpha t$, we have $\ln(\xi/\xi_0) \approx \ln t$, and the limit $\text{LCN} = \lim_{t \rightarrow \infty} (\ln t/t)$ is zero.

If in a Kolmogorov system the LCN for all orbits is between certain positive limits, i. e. if $C_1 \leq \text{LCN} \leq C_2$ the system is called “Anosov”, or C-system.

A C-system has infinite periodic orbits, but none of them is stable. Furthermore the Anosov systems are hyperbolic, i. e. the stable and unstable manifolds of each unstable periodic orbit intersect transversally. The Anosov systems are *structurally stable*, i. e. a small perturbation of an Anosov system leads to another Anosov system.

The usual classification [50] of dynamical systems is the following: Integrable systems, ergodic systems, mixing systems, Kolmogorov systems, Anosov systems.

This classification represents the view that if a system is not integrable it is at least ergodic. The first edition of the book of Landau and Lifshitz [48] separated all dynamical systems into two classes, integrable and ergodic. (However after the 3rd edition this part was omitted). This classification misses the most important property of chaos, the fact that in general chaos co-exists with order in the same system.

In fact, in most systems that have been studied numerically up to now, one finds both chaotic and ordered orbits. Such systems are called systems with divided phase space. Chaotic orbits have a positive Lyapunov characteristic number, while ordered orbits have $\text{LCN} = 0$. Both chaotic and ordered orbits cover parts of the phase space. This is best seen in numerical studies of systems of two degrees of freedom. The ordered and chaotic domains are intricately mixed. However, there are regions where order is predominant, and other regions where chaos is predominant.

Systems of two degrees of freedom that are close to integrable have only small regions of chaos and most orbits are ordered. On the other hand, systems that are far from integrable may seem completely chaotic, but after a closer look we usually find small islands of stability in them.

Completely integrable systems are exceptional. On the other hand, truly ergodic systems are also exceptional. For example, some piecewise linear maps, that are given modulo a constant, or systems with abrupt reflections, like the *stadium* have been proven to be ergodic. But if we add generic nonlinear terms in these maps the structure of phase space changes and islands of stability appear.

The present day classification of dynamical systems is

	Ordered	Chaotic		Random
Compact	Integrable	(General Case) Systems with Divided Phase Space	(Limiting Cases) Ergodic Mixing Kolmogorov Anosov	
Non-compact	Integrable with escapes	Nonintegrable with escapes	–	

A particular class are the noncompact systems (i. e. systems with escapes). In such systems some initial conditions lead to escapes, while others do not. These systems are either integrable, or chaotic. In the latter case chaos appears in the form of chaotic scattering. But properly speaking chaos refers only to compact systems. Finally, the random systems are limiting cases of Anosov systems, because they have infinitely large Lyapunov characteristic numbers.

Integrable Systems

The simplest integrable systems can be solved explicitly. Autonomous Hamiltonian equations of the form (1) with one degree of freedom ($N = 1$) are always solvable.

A very useful set of variables are the action-angle variables $(\vec{J}, \vec{\theta})$. In particular the change of variables

$$x_i = \sqrt{2J_i} \sin \theta_i, \quad y_i = \sqrt{2J_i} \cos \theta_i, \quad (i = 1, 2, \dots, N) \quad (3)$$

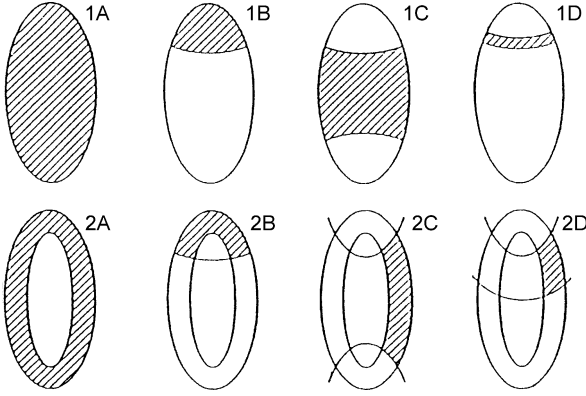
is canonical and the new variables $(\vec{J}, \vec{\theta})$ satisfy canonical equations of the form

$$\dot{\theta} = \frac{\partial H}{\partial J}, \quad \dot{J} = -\frac{\partial H}{\partial \theta} \quad (4)$$

In the particular case that H is a function of the actions only $H = H(\vec{J})$ the actions $\vec{J}(J_1, J_2, \dots, J_N)$ are integrals of motion, because $\dot{J}_i = 0$, and consequently the derivatives of H with respect to J_i are constant, i. e. $\partial H / \partial \vec{J} = \vec{\omega}(\vec{J})$ where $\vec{\omega}$ represents the frequencies $(\omega_1, \omega_2, \dots, \omega_N)$. Then the angles $(\theta_1, \theta_2, \dots, \theta_N)$ are linear functions of the time $\theta_i = \omega_i(t - t_{i0})$. In such a case the motion takes place on a N -dimensional torus.

A function $I(\vec{x}, \vec{y}, t)$ in a Hamiltonian system is called an integral of motion if it remains constant along any orbit, i. e. its total derivative is zero. Then

$$\frac{dI}{dt} = \frac{\partial I}{\partial \vec{x}} \frac{d\vec{x}}{dt} + \frac{\partial I}{\partial \vec{y}} \frac{d\vec{y}}{dt} + \frac{\partial I}{\partial t} = [I, H] + \frac{\partial I}{\partial t} = 0 \quad (5)$$



Astrophysics: Dynamical Systems, Figure 2

Regions filled by the various types of orbits in a Stäckel potential

where \vec{x}, \vec{y} are additive and

$$[I, H] \equiv \frac{\partial I}{\partial \vec{x}} \frac{\partial H}{\partial \vec{y}} - \frac{\partial I}{\partial \vec{y}} \frac{\partial H}{\partial \vec{x}} \quad (6)$$

is called the *Poisson bracket* between the functions I and H .

An important class of integrable systems are the Stäckel potentials [33,71,76]. They are given in elliptical coordinates (λ, μ) by potentials of the form

$$V = -\frac{[F_1(\lambda) - F_2(\mu)]}{\lambda - \mu} \quad (7)$$

where λ and μ are the roots of the second degree equation in τ

$$\frac{x^2}{\tau - a^2} + \frac{y^2}{\tau - b^2} = 1 \quad (8)$$

with Cartesian coordinates x, y and $a^2 \geq b^2$. The Eq. (8) represents confocal ellipses and hyperbolas. From every point (x, y) passes an ellipse (defined by λ) and

a hyperbola (defined by μ). Both have the same foci at $y = \bar{a} = \pm(a^2 - b^2)^{1/2}$. The orbits in the plane (x, y) fill regions between such ellipses and hyperbolas (Fig. 2).

The Stäckel potentials are in general non-rotating. Up to now only one rotating Stäckel potential has been found, besides the trivial case of a homogeneous ellipsoid, namely the potential

$$V = -\frac{k_1}{[(x^2 + y^2)^2 + 2\bar{a}^2(x^2 - y^2) + \bar{a}^4]^{1/2}} + \frac{1}{2}\Omega_s^2 r^2 \quad (9)$$

The orbits in the rotating case are in general tubes around both foci, or tubes around one focus only [24] (Fig. 3).

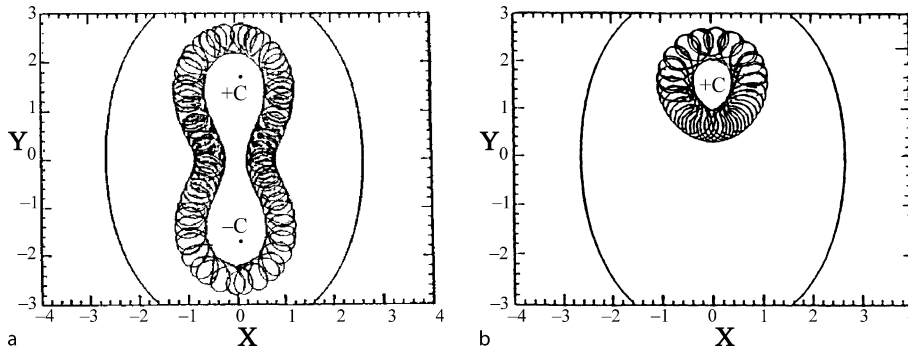
Stäckel potentials in three degrees of freedom were considered by many authors like [31,45,53]. There are 32 possible forms of 3-D Stäckel orbits [21].

The Stäckel potentials have been used extensively in recent years in constructing self-consistent models of elliptical galaxies [7,32,46,72]. Such models represent well real galaxies.

Formal Integrals. KAM and Nekhoroshev Theory

If a system is close to an integrable one, we may use perturbation methods to find approximate integrals that are valid for long intervals of time, and sometimes for all times. Usually such integrals are in the form of formal power series. Formal integrals for polynomial Hamiltonians were first derived by Whittaker [77,78].

Another form of such integrals was derived by Birkhoff [5] and by Cherry [9]. Formal integrals appropriate for galactic dynamics were derived by Contopoulos [14,15]. In the case of an axisymmetric galaxy an integral of this form is called a *third integral* (the first integral



Astrophysics: Dynamical Systems, Figure 3

Orbits in a rotating Stäckel potential

being the energy and the second the z component of the angular momentum).

Birkhoff [5] and Cherry ([9] use Hamiltonians of the form)

$$H = i\omega_1 \xi_1 \eta_1 + i\omega_2 \xi_2 \eta_2 + \dots + i\omega_N \xi_N \eta_N + H_3 + H_4 + \dots \quad (10)$$

where H_k is of degree k in $(\xi_\lambda, \eta_\lambda)$, and $\xi_\lambda, \eta_\lambda$ are complex variables ($\lambda = 1, 2, \dots, N$).

By successive canonical changes of variables the Hamiltonian takes the *normal form* $\bar{H} = \bar{H}_2 + \bar{H}_4 + \dots$ where \bar{H}_k is a function of the products $\Omega_\lambda = (\bar{\xi}_\lambda \bar{\eta}_\lambda)$ only (in the nonresonant case), thus it is an even function of the variables. Then the quantities Ω_λ are integrals of motion, and the equations of motion can be solved explicitly.

The third integral in a galaxy refers to a Hamiltonian of the form [14]

$$H = \frac{1}{2} \sum_{\lambda=1}^2 (y_\lambda^2 + \omega_\lambda^2 x_\lambda^2) + H_3 + H_4 + \dots \quad (11)$$

where H_k is a function of degree k in x_λ, y_λ ($\lambda = 1, 2$). This Hamiltonian represents a galactic potential on a plane of symmetry. If $r = r_0$ represents a circular orbit on this plane, we have $x_1 = r - r_0, x_2 = z$ and $y_1 = \dot{x}_1, y_2 = \dot{x}_2$. Then the term H_3 of the Hamiltonian takes the form

$$H_3 = -\epsilon x_1 x_2^2 - \frac{\epsilon'}{3} x_1^3 \quad (12)$$

An integral of motion has its total derivative equal to zero. The successive terms of the third integral Φ can be found from the equation

$$\frac{d\Phi}{dt} = [\Phi, H] = 0 \quad (13)$$

(where $\Phi = \Phi_2 + \Phi_3 + \dots$, and $[\Phi, H]$ is the Poisson bracket) by equating terms of equal degree.

The Hamiltonian H_2 has two independent integrals of motion $\Phi_2 = \frac{1}{2} [y_1^2 + \omega_1^2 x_1^2]$ and $\Phi'_2 = H_2 - \Phi_2$. Then, we find

$$\Phi_3 = -\frac{1}{(\omega_1^2 - 4\omega_2^2)} \left[(\omega_1^2 - 2\omega_2^2) x_1 x_2^2 - 2x_1 y_2^2 + 2y_1 x_2 y_2 \right] \quad (14)$$

etc. The expression (14) and the higher order terms Φ_k of the third integral contain denominators of the form $(m\omega_1 - n\omega_2)$ that become zero if $\omega_1/\omega_2 = n/m$ (rational). Then Φ contains secular terms. For example, the form (14)

of Φ_3 cannot be valid if $\omega_1 - 2\omega_2 = 0$. It is not valid also if $\omega_1 - 2\omega_2$ is small (case of a “small divisor”) because then Φ_3 is large, while the series expansion of Φ implies that the successive terms Φ_2, Φ_3 , etc. become smaller as the order increases.

In such resonance cases we can construct a third integral by a different method. If ω_1/ω_2 is rational there are further isolating integrals of motion of the lowest order Hamiltonian H_2 beyond the partial energies Φ_2, Φ'_2 . For example, if $\omega_1/\omega_2 = n/m$ then H_2 has also the integrals

$$\begin{aligned} \frac{S_0}{C_0} &= (2\Phi_2)^{m/2} (2\Phi'_2)^{n/2} \frac{\sin}{\cos} [n\omega_2 T - m\omega_1 (T - T_0)] \\ &= \text{constant} \end{aligned} \quad (15)$$

which can be written as polynomials of degree $m + n$ in the variables x_λ, y_λ [15]. The integrals S_0, C_0 can be used for constructing series of the form $S = S_0 + S_1 + S_2 + \dots$, $C = C_0 + C_1 + C_2 + \dots$. These series also contain secular terms. However, we can prove that there are combinations of the three integrals Φ_2, Φ'_2 and S_0 (or C_0) that do not contain secular terms of any order, thus they are formal integrals of motion.

The series of the third integral are in general not convergent. However, they may give very good applications, if truncated at an appropriate level. A simple example of such a series was provided by Poincaré ([66], Vol. II). The series

$$f_1 = \sum_{n=0}^{\infty} \frac{n!}{(1000)^n} \quad (16)$$

is divergent (i. e. f_1 is infinite), but if it is truncated at any large order n smaller than 1000, it gives approximately the same value, and it seems to converge, as n increases, to a finite value. On the other hand the series

$$f_2 = \sum_{n=0}^{\infty} \frac{(1000)^n}{n!} \quad (17)$$

is convergent, but in order to give an approximate value of f_2 one has to take more than 1000 terms.

There is an important theorem establishing the existence of N -dimensional invariant tori in systems of N degrees of freedom. This is the celebrated KAM theorem, announced by Kolmogorov [47]. The details of the proof were given by Arnold [2,3] and independently by Moser [59,60].

The KAM theorem states that in an autonomous Hamiltonian system, close to an integrable system expressed in action-angle variables, there are N -dimensional

invariant tori, i.e. tori containing quasi-periodic motions with frequencies ω_i , satisfying a diophantine condition (after the ancient Greek mathematician Diophantos) $|\vec{\omega}\vec{k}| > \gamma/|\vec{k}|^{N+1}$, where $\vec{\omega}\vec{k} = \omega_1 k_1 + \omega_2 k_2 + \dots + \omega_N k_N$ with k_1, k_2, \dots, k_N integers and $|\vec{k}| = |k_1| + |k_2| + \dots + |k_N| \neq 0$, while γ is a small quantity depending on the perturbation. The set of such tori has a measure different from zero if γ is small.

In the case of two degrees of freedom the ratio ω_1/ω_2 is far from all resonances, if it satisfies a diophantine condition $|\omega_1/\omega_2 - n/m| > \epsilon/m^3$. If we exclude the sets of all irrationals that are near every rational n/m , on both its sides, the set of the excluded irrationals is smaller than

$$\sum_{m=1}^{\infty} \sum_{n=1}^{m-1} \frac{2\epsilon}{m^3} < 2\epsilon \sum_{m=1}^{\infty} \frac{1}{m^2} = 2C\epsilon \quad (18)$$

where $C = \pi^2/6 \approx 1.64$. Therefore the set of diophantine numbers ω_1/ω_2 in the interval $(0, 1)$ has a measure larger than $1 - C\epsilon$ and if ϵ is small this is not only positive, but it can even be close to 1.

The formal integrals Φ are convergent only in integrable cases. In other cases Φ are asymptotic series that may not represent a particular functions. The question then arises for how long the formal integrals $\Phi = \Phi_2 + \Phi_3 + \dots$ are valid within a given approximation. The answer to this question is provided by the theory of Nekhoroshev [62]. This theory establishes that the formal integrals are valid over exponentially long times. Namely, if the deviation from an integrable system is of order ϵ the Nekhoroshev time is of order

$$t = t_* \exp\left(\frac{M}{\epsilon^m}\right) \quad (19)$$

where t_* , M and m are positive constants. The formal integrals must be truncated at an optimal order, such that the variation of the truncated integrals is minimum. An analytical estimate of the optimal order of truncation was provided recently by Efthymiopoulos et al. [36].

Let us consider the orbits in the phase space of an autonomous Hamiltonian system of two degrees of freedom $H \equiv H(x, y, p_x, p_y) = h$. As the energy of the system, h , is an integral of motion, one variable, say p_y , can be derived from H if the other three variables are known. All orbits with fixed energy equal to h , lie on the above 3-D surface, in the 4-D phase space. Thus, we may consider the orbits in the 3 dimensional phase space (x, y, p_x) .

If every orbit is intersected by a particular surface $S(p_x, p_y, x, y) = \text{const}$ within every fixed interval of time T , then this surface is called a Poincaré surface of section. The intersections of the points of S are called con-

sequents; they define a map on the surface of section S that is called a Poincaré map. In particular, a periodic orbit of multiplicity three is represented by three points (e.g. O_1, O_2, O_3 in Fig. 4).

Periodic Orbits

The most important orbits in a dynamical system are the periodic orbits. If the periodic orbits are stable they trap around them a set of nonperiodic orbits with the same average topology. If they are unstable they repel the orbits in their neighborhood. In integrable systems the main unstable orbits separate the resonant orbits from the nonresonant orbits. In nonintegrable systems the unstable periodic orbits introduce chaos in the system. Thus, the study of a dynamical system usually starts with an exploration of its periodic orbits.

In the case of an area preserving map $x_1 = f(x_0, y_0, \epsilon)$, $y_1 = g(x_0, y_0, \epsilon)$, where ϵ is a parameter, a periodic orbit is an invariant point $x_1 = x_0$, $y_1 = y_0$. Its stability is found by linearizing the equations around this point. We have $\Delta x_1 = a\Delta x_0 + b\Delta y_0$, $\Delta y_1 = c\Delta x_0 + d\Delta y_0$, where $a = \partial f/\partial x_0$, $b = \partial f/\partial y_0$, $c = \partial g/\partial x_0$, $d = \partial g/\partial y_0$. The conservation of areas implies $ad - bc = 1$.

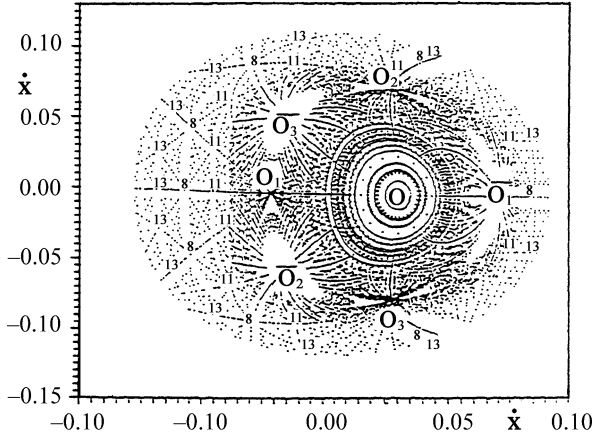
In a Hamiltonian system of two degrees of freedom the periodic orbits and the deviations $\vec{\xi} = (\Delta x, \Delta y)$ are defined in the 4-D phase space (x, y, \dot{x}, \dot{y}) . Along particular directions in the 4-D phase space the original deviation $\vec{\xi}_0 = (\Delta x_0, \Delta y_0)$ becomes $\vec{\xi}_1 = (\Delta x_1, \Delta y_1)$ after one iteration, where $\vec{\xi}_1$ is equal to $\vec{\xi}_0$ multiplied by a factor λ . Such vectors $\vec{\xi}_0$ are called *eigenvectors* (with arbitrary length) and the factors λ are called *eigenvalues*. The eigenvector forms an angle Φ with the x -axis, given by $\tan \Phi = (\lambda - a)/b = c/(\lambda - d)$, that has two opposite solutions Φ and $\Phi + 180^\circ$ for each real eigenvalue λ .

The two eigenvalues λ_1, λ_2 are found from the equation

$$\begin{vmatrix} a - \lambda & b \\ c & d - \lambda \end{vmatrix} = \lambda^2 - (a + d)\lambda + 1 = 0 \quad (20)$$

The roots λ_1, λ_2 of this equation are inverse. If $|a + d| > 2$ the roots λ are real ($|\lambda_1| > 1$ and $|\lambda_2| < 1$). Then, the orbit is unstable and the eigenvectors $\vec{\xi}_{01}, \vec{\xi}_{02}$ form the angles Φ with the x -axis. On the other hand if $|a + d| < 2$ the roots are complex conjugate with $|\lambda_i| = 1$ and the orbit is stable.

Ordered orbits lie on invariant tori that intersect a surface of section along invariant curves. Orbits that do not lie on closed invariant curves and do not escape to infinity are chaotic. On a surface of section such orbits are represented by the irregular distribution of their consequents.



Astrophysics: Dynamical Systems, Figure 4

Distribution of periodic orbits of various multiplicities in the potential $V = \frac{1}{2}(\omega_1^2 x_1^2 + \omega_2^2 x_2^2) - \epsilon x_1 x_2^2$ with $\omega_1^2 = 1.6$, $\omega_2^2 = 0.9$, $\epsilon = 0.3$

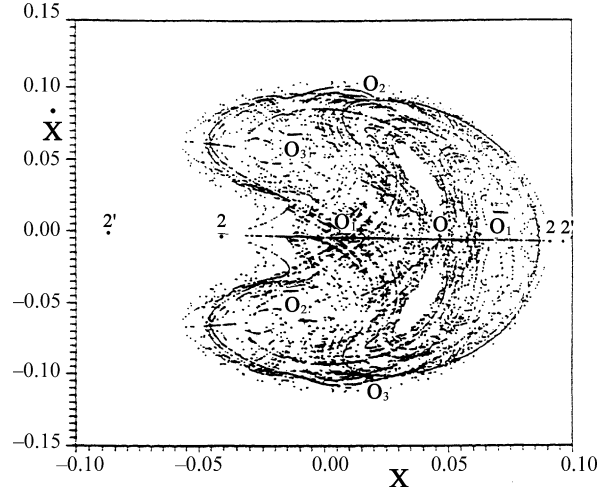
Such chaotic orbits appear near every unstable periodic orbit of a nonintegrable system. The domains filled by the chaotic orbits surround the corresponding islands of stability of the same chaotic zone that contains the unstable periodic orbits. For example, the chaotic domains of the Hamiltonian (11) with $H_3 = -\epsilon x_1 x_2^2$ are near the unstable points O_1, O_2, O_3 of type $n/m = 2/3$ in Fig. 4. These chaotic domains surround also the islands around the stable periodic orbit $(\overline{O_1 O_2 O_3})$.

For relatively small perturbations the various resonances are well separated by invariant curves that close around the central periodic orbit O , therefore chaos is limited. But when the perturbation increases the various chaotic domains increase and join each other, destroying the separating invariant curves, and producing a large connected chaotic domain (Fig. 5) (for the same Hamiltonian but with $\epsilon = 4.5$). This is a manifestation of a “resonance overlap”, or “resonance interaction”.

The asymptotic curves of a hyperbolic point of an integrable system join into one separatrix, unless these curves extend to infinity. The separatrix may be considered as an orbit (homoclinic orbit), or as an invariant curve containing the images of the initial points on this curve. An orbit starting very close to the origin along the unstable branch U will move far from O , but eventually it will return close to O along the stable branch S .

However, in a nonintegrable system there are no closed separatrices, and the unstable and stable asymptotic curves intersect at infinite points called homoclinic points.

All initial points on an asymptotic curve generate *asymptotic orbits*, i.e. orbits approaching the periodic orbit O either in the forward or in the backward direction of



Astrophysics: Dynamical Systems, Figure 5

As in Fig. 4 for $\epsilon = 4.5$

time. The homoclinic points define *doubly asymptotic orbits* because they approach the orbit O both for $t \rightarrow -\infty$ and for $t \rightarrow \infty$.

The intersecting asymptotic curves of an unstable periodic orbit form the so-called *homoclinic tangle*. If we start an orbit in this region, its consequents fill the region in a practically random way, forming the chaotic domain that we see near every unstable periodic orbit of a nonintegrable system.

Transition from Order to Chaos

Dissipative Systems. The Logistic Map

A simple one dimensional dissipative system, where we can see the transition from order to chaos, is the logistic map

$$x_{i+1} = f(x_i) = 4\lambda x_i(1 - x_i) \quad (21)$$

This is the prototype of a quadratic map, and it is studied extensively in the pioneering article of May [56] and in books on chaos (see e.g. [1,12]). Thus, we will only describe here its main properties. We consider only values of λ between 0 and 1 in order to have x always between 0 and 1. The logistic map has two simple periodic orbits on the intersection of the parabola (21) with the diagonal $x_{i+1} = x_i$, namely the points $x = 0$ and $x = x_0 = 1 - \frac{1}{4\lambda}$.

A periodic orbit x_0 is stable if the derivative $f' = df/dx_0$ is absolutely smaller than 1. This happens if $1/4 < \lambda < \Lambda_1 = 3/4$.

It is easily found that when the orbit $x_0 = 1 - (1/4\lambda)$ of period 1 becomes unstable, then a period -2 family of

periodic orbits (x_a, x_b) bifurcates, which is stable if $\Lambda_1 < \lambda < \Lambda_2 = (1 + \sqrt{6})/4$.

There is an infinite set of period doubling bifurcations (Fig. 6a). The intervals between successive bifurcations decrease almost geometrically at every period doubling, i. e.

$$\lim_{n \rightarrow \infty} \frac{\Lambda_n - \Lambda_{n-1}}{\Lambda_{n+1} - \Lambda_n} = \delta = 4.669201609 \dots \quad (22)$$

This relation is asymptotically exact (i. e. for $n \rightarrow \infty$) [28,37]. The number δ is universal, i. e. it is the same in generic dissipative systems.

The curves $F^m(\lambda)$, with $m = 2^n$, starting at the n th period doubling bifurcation, are similar to each other, decreasing by a factor $\alpha = 2.50 \dots$ in size at each successive period doubling bifurcation.

As the ratios (22) decrease almost geometrically, the value Λ_n converges to the limiting value $\Lambda_\infty = 0.893$.

In 2-D maps the areas are reduced at every iteration in the dissipative case and conserved in the conservative case. An example is provided by the Hénon map [42]

$$x_{i+1} = 1 - Kx_i^2 + y_i, y_{i+1} = bx_i \pmod{1} \quad (23)$$

(the original map was not given modulo 1). The Jacobian of the transformation is $J = b$. If $0 < b < 1$ the system is dissipative and if $b = 1$ it is conservative.

In the dissipative case there are *attractors*, i. e. manifolds to which tend many orbits as $t \rightarrow \infty$ (or $i \rightarrow \infty$ in maps). These attractors may be points, curves (limit cycles), or strange attractors [39,43,52], which are composed of infinite lines.

Other mechanisms of transition to chaos in dissipative systems are reviewed by Contopoulos (Section 2.6.3 in [21]).

Conservative Systems

The first numerical study of the transition to chaos in a conservative system was provided by Hénon and Heiles [44]. The transition to chaos follows a number of scenarios.

Infinite Period Doubling Bifurcations This scenario is similar to the corresponding scenario of the dissipative case. In fact a large degree of chaos is introduced after an infinite number of period doubling bifurcations along the characteristics of the periodic orbits ($x = x(\lambda)$). However, there are three differences in the conservative case.

(1) Infinite period doubling bifurcations appear generically only in systems of two degrees of freedom.

- (2) The bifurcation ratio is different ($\delta \cong 8.72$ in conservative systems versus $\delta \cong 4.67$ in dissipative systems). Also the scaling factors are different.
- (3) The introduction of chaos follows a different pattern. While in the dissipative case chaos appears only *after* infinite bifurcations (Fig. 6a), in the conservative systems some chaos appears around every unstable orbit. As the perturbation increases the chaotic domains increase in size and they merge to form a large connected domain (Fig. 6b).

In some cases the appearance of infinite bifurcations may be followed by their corresponding disappearance, as the perturbation increases further. Then the infinite unstable families terminate in the opposite way forming infinite bubbles.

Infinite Bifurcations from the same Family While in the infinite bifurcations scenario we have pitchfork bifurcations from the successive bifurcating families (Fig. 6), in the present case the original family of periodic orbits becomes successively stable and unstable, an infinite number of times. For example, this happens in the case of the Hamiltonian

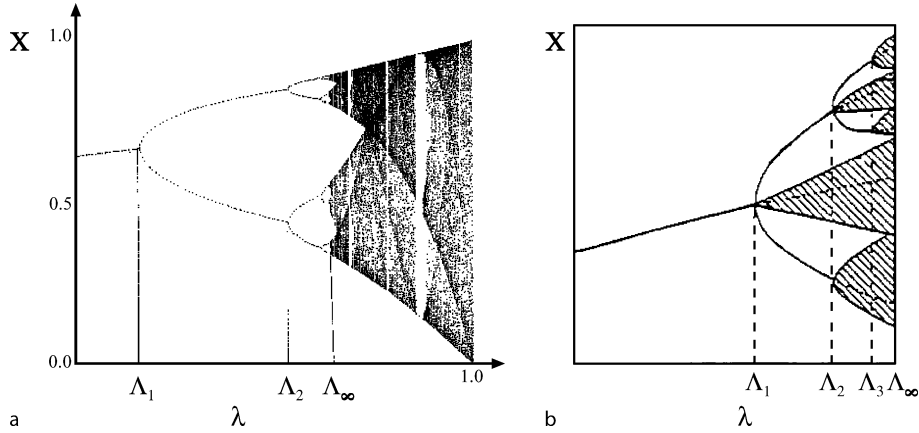
$$H = \frac{1}{2}(\dot{x}^2 + \dot{y}^2 + x^2 + y^2) + xy^2 = h \quad (24)$$

[26]. The stable bifurcating families undergo further period doubling bifurcations. The successive stable and unstable intervals along the original family have a bifurcation ratio $\delta \cong 9.22$. But this bifurcation ratio is not universal. It depends on the particular dynamical system considered. In the case (24) it was proven analytically by Heggie [41] that this ratio is $\delta = \exp(\pi/\sqrt{2})$ but in other systems it is different. In all cases near the limiting point there is an infinity of unstable families that produce a large degree of chaos.

Infinite Gaps In a dynamical system representing a rotating galaxy, the central periodic family, which consists of circular orbits in the unperturbed case, has two basic frequencies. These are the rotational frequency in the rotating frame ($\Omega - \Omega_s$) (where Ω is the angular velocity along a circular orbit and Ω_s the angular velocity of the system, e. g. of the spiral arms), and the epicyclic frequency κ of radial oscillations from the circular orbit.

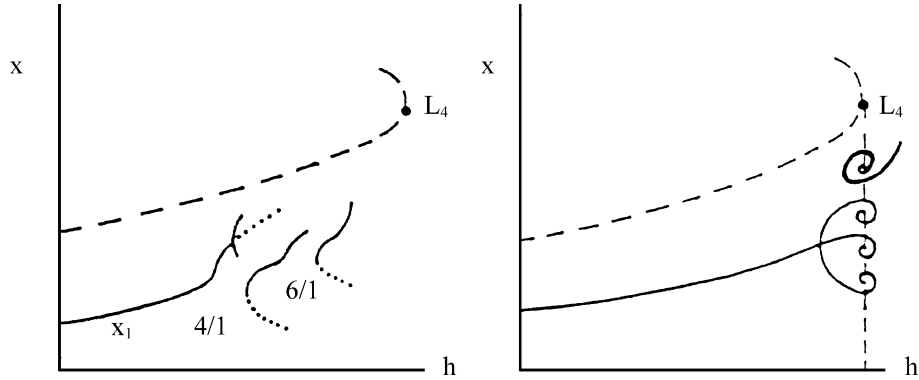
The ratio of the two frequencies (rotation number) goes through infinite rational values $\kappa/(\Omega - \Omega_s) = n/m$ (resonances) as the energy increases.

The most important resonances are those with $m = 1$. At every even resonance ($n = 2n_1$, $m = 1$), we have a bifurcation of a resonant family of periodic orbits in the



Astrophysics: Dynamical Systems, Figure 6

Bifurcations and chaos: **a** in a dissipative system (the logistic map) and **b** in a conservative system



Astrophysics: Dynamical Systems, Figure 7

a Gaps appear along the characteristic of the family x_1 at all even resonances $2n_1/m$. **b** Spiral characteristics near corotation (L_4)

unperturbed case, which becomes a gap in the perturbed case (when a spiral, or a bar, is added to the axisymmetric background, Fig. 7a). The gap is formed by separating the main family into two parts and joining the first part with a branch of the resonant family, and the second part containing the other branch of the resonant family.

As we approach corotation (where $\Omega = \Omega_s$) the rotation number tends to infinity. Therefore, an infinity of gaps are formed, followed by the appearance of an infinity of unstable periodic orbits, and this leads to a large degree of chaos [19].

Infinite Spirals There is an infinite number of families of periodic orbits whose characteristics form spirals near corotation (Fig. 7b). These families contain an infinite number of unstable periodic orbits that interact with each other, producing chaos [21,65].

Resonance Overlap

In a nonintegrable Hamiltonian we have several types of resonant islands and their size can be found by means of the third integral, calculated for each resonance separately.

We consider a Hamiltonian of the form (11) with ω_1/ω_2 near a resonance n/m . The maximum size D of an island of type n/m is of $O(\epsilon^{(m+n-4)/2})$ [17]. The same is true for the area covered by the islands of the resonance n/m .

If the islands of various nearby resonances are well separated, then there is no resonance overlap and no large degree of chaos. However, as the perturbation increases the various islands increase in size and the theoretical islands overlap. As this overlapping is not possible in reality, what really happens is that the region between the islands becomes chaotic.

There are two ways to estimate the critical perturbation of the resonance overlap.

- (1) We calculate the areas of the various islands. When this quantity becomes equal to the total area of phase space on the surface of section, we start to have large degree of chaos [17].
- (2) We may find the critical value for the production of large chaos between two main resonances of the system, say $n/m = 4/3$ and $n/m = 3/2$. The positions of the triple and double periodic orbits are given by the corresponding forms of the third integral as functions of the perturbation ε (characteristics). The theoretical characteristics intersect for a certain value of ε . On the other hand, the real characteristics cannot intersect. Instead, between the two characteristics, a chaotic region is formed, due to the resonance overlap.

The resonance overlap criterion for large chaos was considered first by Rosenbluth et al. [67] and Contopoulos [17]. It was later described in detail by Walker and Ford [75], Chirikov et al. [11], Chirikov [10], and many others.

Arnold Diffusion

Nearly integrable autonomous Hamiltonian systems of N degrees of freedom have a $2N - 1$ dimensional phase space of constant energy, and a large set of N -dimensional invariant surfaces (invariant tori), according to the KAM theorem. If $N = 2$ there are $N = 2$ dimensional surfaces separating the $2N - 1 = 3$ dimensional phase space, and diffusion through these surfaces is impossible. But if $N \geq 3$ the N -dimensional surfaces do not separate the $2N - 1$ dimensional phase space, and diffusion is possible all over the phase space. This is called “Arnold diffusion” [3]. Such a diffusion is possible even if the perturbation is infinitesimal.

In order to understand better this phenomenon consider a system of three degrees of freedom, that has 3-D tori in a 5-D phase space of constant energy. If we reduce the number of both dimensions by two we have 1-D tori (lines) in a 3-D space. In an integrable case there is a line passing through every point of the space and all motions are regular. However, in a nonintegrable case there are gaps between the lines and these gaps communicate with each other. Therefore, if a chaotic orbit starts in such a gap it may go everywhere in the 3-D space.

Extensive numerical studies (e. g. [49]) in a 4-D map have shown that in the same dynamical system there are both ordered domains where diffusion is very slow

(Arnold diffusion), and chaotic domains, where diffusion is dominated by resonance overlap.

As an example we consider two coupled standard maps

$$\begin{aligned} x'_1 &= x_1 + y'_1, \\ y'_1 &= y_1 + \frac{K}{2\pi} \sin 2\pi x_1 - \frac{\beta}{\pi} \sin 2\pi(x_2 - x_1) \\ &\quad (\text{mod } 1) \end{aligned} \quad (25)$$

$$\begin{aligned} x'_2 &= x_2 + y'_2, \\ y'_2 &= y_2 + \frac{K}{2\pi} \sin 2\pi x_2 - \frac{\beta}{\pi} \sin 2\pi(x_1 - x_2) \end{aligned}$$

Here K is the nonlinearity and β the coupling constant [25].

Taking the same initial conditions, we calculate the time of diffusion from an ordered region to the large chaotic sea. If the coupling β is larger than a critical value $\beta = \beta_c \cong 0.305$, the diffusion time T increases exponentially with decreasing β . If however, $\beta < \beta_c$, the time T increases superexponential as β decreases. We can identify the exponential increase case as due to resonance overlap diffusion and the superexponential increase as due Arnold diffusion.

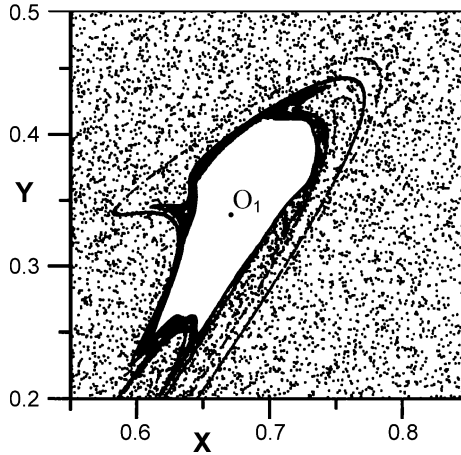
Cantori

When the perturbation increases and a particular torus is destroyed, it becomes a cantorus, i. e. a Cantor set of points that is nowhere dense [4,64]. The cantorus has a countable infinity of gaps, but it contains a noncountable infinity of orbits. The most surprising property of the cantori is that their measure is zero and their fractal dimension is also zero.

In the case of the standard map, cantori with golden rotation number $rot = (\sqrt{5} - 1)/2$ (expressed in continuous fraction representation as $[1, 1, 1, \dots]$) exist for all values of K larger than a critical value $K_{cr} = 0.972$ ([55]).

The cantorus is formed when all the periodic orbits corresponding to the various truncations of the golden number become unstable. In the limit $K = K_{cr}$ the invariant curve (torus) is a fractal with self-similar structure on all scales [40,69].

What happens in general around an island of stability is the following. Before the destruction of the last KAM curve surrounding the island, there is a chaotic layer, just inside the last KAM curve, while outside the last KAM curve there is a large chaotic sea. When the last KAM curve becomes a cantorus the orbits from the inner chaotic layer can escape to the large chaotic sea, but only after a long time. Thus, the region just inside a cantorus with small holes is differentiated by the larger density of the points inside it (Fig. 8). This is the phenomenon of *stickiness*.



Astrophysics: Dynamical Systems, Figure 8

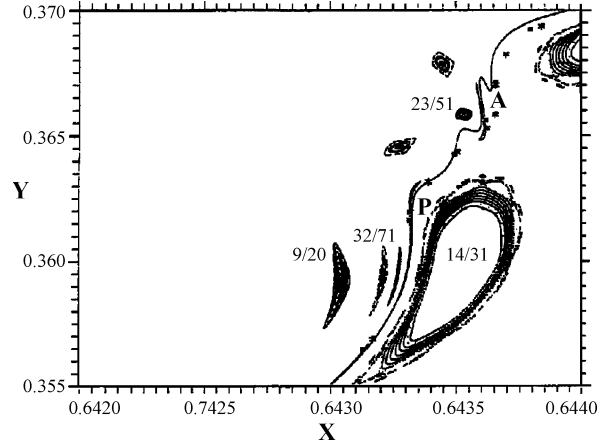
Stickiness in the standard map for $K = 5$. The sticky region (dark) surrounds an island of stability, and is surrounded by a large chaotic sea

Stickiness

This phenomenon was found numerically by Contopoulos [18]. Stickiness has been observed in practically all nonintegrable dynamical systems that contain islands [57]. In Fig. 8 the sticky zone (the dark region surrounding the island O_1) is limited on its inner side by the outer boundary of the island O_1 and on its outer side by a set of cantori, but mainly by the cantorus that has the smallest gaps. An orbit starting in the sticky zone requires a long time to escape outside. The details of the escape can be found if we calculate the asymptotic curves of unstable periodic orbits just inside the main cantorus.

As an example consider an unstable asymptotic curve from the periodic orbit of period 215 in the case of the standard map for $K = 5$ (Fig. 9; [35]). If we start with a small interval along the asymptotic curve, we find an image of this curve that crosses the cantorus after two iterations along the same asymptotic curve. In Fig. 9 we see that the asymptotic curve starts from P by going first to the left and downwards and then it moves to the right and upwards, crossing the cantorus in one of its largest gaps (marked A in Fig. 9). But then the asymptotic curve makes several oscillations back and forth, entering again inside the cantorus several times, before going to a large distance in the chaotic sea. Then it moves around for a long time before coming back and getting trapped again in the sticky zone close to the island.

Another type of stickiness refers to the unstable asymptotic curves that extend far into the chaotic sea (dark lines in Fig. 8). These asymptotic curves attract nearby orbits and thus thick dark lines are formed [23].



Astrophysics: Dynamical Systems, Figure 9

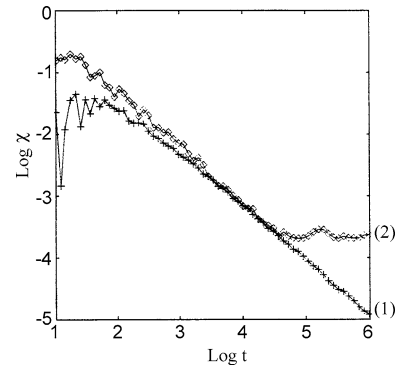
An asymptotic curve of the periodic orbit P in the standard map with $K = 5$ makes several oscillations inside the cantorus (stars) and then escapes. We mark some islands inside the cantorus (on the right) and outside it (on the left)

Dynamical Spectra

Despite the importance of the Lyapunov characteristic numbers in distinguishing between order and chaos, their practical application is limited by the very long calculations that are usually required for their evaluation. In fact, if we calculate the finite time LCN

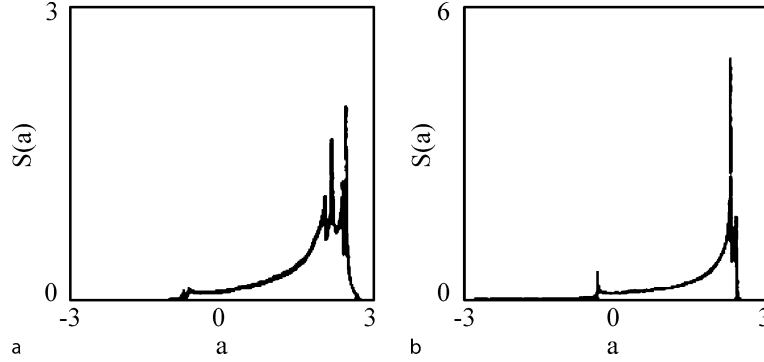
$$\chi = \frac{\ln |\xi/\xi_0|}{t} \quad (26)$$

the variation of χ is irregular for relatively small t and only for large t the value of χ of a chaotic orbit stabilizes and tends to a constant limiting value (curve (2) of Fig. 10), which is the Lyapunov characteristic number



Astrophysics: Dynamical Systems, Figure 10

The variation of $\chi(t)$ of (1) an ordered orbit, and (2) a chaotic orbit



Astrophysics: Dynamical Systems, Figure 11

The spectra of stretching numbers: **a** of the standard map and **b** of the Hénon map

$LCN = \lim_{t \rightarrow \infty} \chi$. If, on the other hand, χ varies approximately as $1/t$, the LCN is zero (ordered orbit (1) of Fig. 10).

Of special interest is to find a finite time LCN after the shortest possible time. In the case of a map the shortest time is one iteration ($t = 1$). Thus one finds the quantities

$$a_i = \ln |\xi_{i+1}/\xi_i| \quad (27)$$

which are called Lyapunov indicators by Froeschlé et al. [38], or stretching numbers by Voglis and Contopoulos [73].

The distribution of successive values of the stretching numbers a_i along an orbit is called the “spectrum of stretching numbers”. Namely the spectrum gives the proportion dN/N of the values of a_i in a given interval $(a, a + da)$ divided by da , i. e.

$$S(a) = dN/(Nda) \quad (28)$$

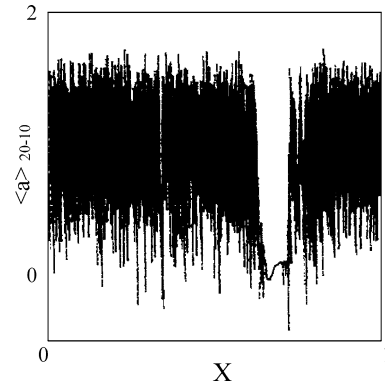
where da is a small quantity and N a large number of iterations.

The main properties of the spectra of stretching numbers are [73]:

- (1) The spectrum is invariant along the same orbit.
- (2) The spectrum does not depend on the initial deviation ξ_0 from the same initial point in the case of two dimensional maps.
- (3) The spectrum does not depend on the initial conditions of orbits in the same connected chaotic domain.
- (4) In the case of ordered orbits in two dimensional maps the spectrum does not depend on the initial conditions on the same invariant curve.

The Lyapunov characteristic number is equal to $LCN = \int S(a) a da$. The spectrum gives much more information about a system than LCN. E. g. the standard map

$$x' = x + y', \quad y' = y + \frac{K}{2\pi} \sin 2\pi x \pmod{1} \quad (29)$$



Astrophysics: Dynamical Systems, Figure 12

The average value of the stretching number, $\langle a \rangle$, for 10 iterations (beyond the first 10 transients) as a function of x for constant y

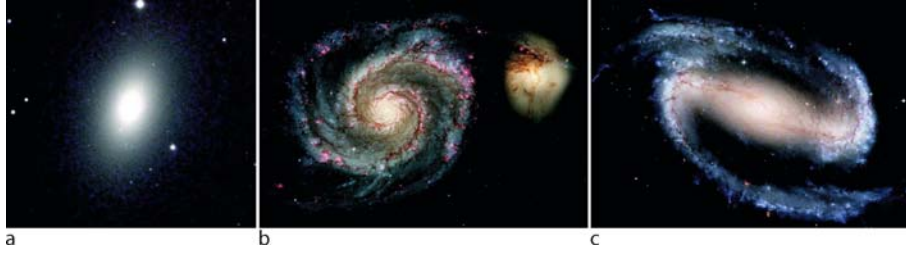
for $K = 10$ and the Hénon map (23) for $K' = 7.407$ and $b = 1$ give two apparently random distributions of points with the same $LCN = 1.62$. However, their spectra are quite different (Fig. 11).

The use of the stretching numbers allows a fast distinction between ordered and chaotic orbits. E.g. while the calculation of the LCN requires $10^5 - 10^6$ iterations, in Fig. 12 we can separate the chaotic from the ordered orbits after only 20 iterations. In Fig. 12 the noise is large, nevertheless the chaotic orbits have roughly the same value of $LCN = \langle a \rangle > 0$, while the ordered orbits have $\langle a \rangle$ very close to zero. Further details about this topic can be found in the book of Contopoulos [21].

Application: Order and Chaos in Galaxies

Galactic Orbits

Galaxies are composed of stars, gas (including dust), and dark matter. The stars and the dark matter produce the



Astrophysics: Dynamical Systems, Figure 13

The basic types of Galaxies **a** M59/NGC 4621 Elliptical [E], **b** M51/NGC 5194 normal spiral [Sb] and **c** NGC 1300 barred spiral [SBb]

main part of the galactic potential and force, while the contribution of the gas in the potential and force is small.

In the Hubble classification of galaxies, the elliptical galaxies (E) and the early type spiral galaxies (Sa, Sb) and barred galaxies (SBa, SBb) have less gas than the late-type galaxies (Sc, SBc) and the irregular galaxies (I) (Fig. 13).

Galaxies that have a well developed spiral structure (usually two symmetric spiral arms) are called “grand design”, while galaxies with irregular and multiple fragments of spirals are called “flocculent”.

The study of stellar (or mainly stellar) galaxies is based on a systematic exploration of their orbits.

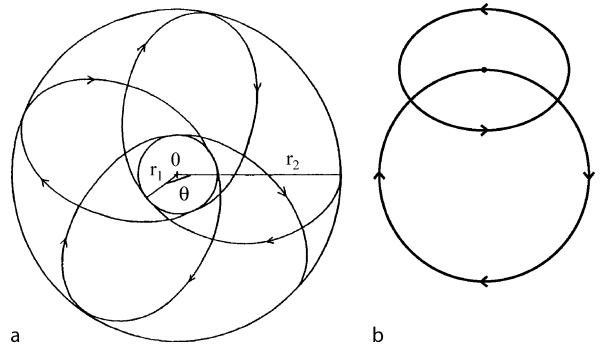
The exploration of orbits in galaxies is very important, because the orbits are needed in constructing self-consistent models of galaxies.

Self-consistency is a new type of problem that does not appear, e.g. in accelerators, or in the solar system. It requires the construction of appropriate sets of orbits of stars, such that their superposition gives a response density that matches the imposed density of the model. Such a construction is done in many cases by taking a grid of initial conditions and calculating the orbits numerically in a computer and then populating these orbits with stars. But it is much more helpful and illuminating if one studies the types of orbits that form the building blocks of the galaxies.

Thus, the first step in understanding the structure and dynamics of a galaxy is to calculate its orbits, periodic, quasi-periodic and chaotic.

The most important orbits in a galactic system are the periodic orbits. The stable orbits trap around them sets of quasi-periodic orbits, that give the main features of the galaxy. On the other hand the unstable orbits separate the various types of ordered orbits, and also characterize the chaotic domains in a galaxy.

The simplest galactic orbits are the circular periodic orbits on the plane of symmetry of an axisymmetric galaxy. The circular orbits in such a galaxy are in general stable.



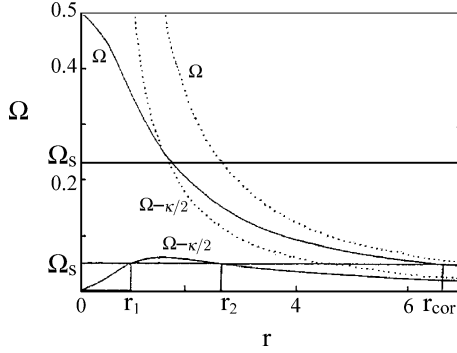
Astrophysics: Dynamical Systems, Figure 14

a A rosette orbit in an axisymmetric galaxy. **b** The corresponding epicyclic orbit

Orbits close to the stable circular periodic orbits are called epicyclic orbits. Such orbits fill circular rings in the plane of symmetry of the galaxy and are called “rosette” orbits (Fig. 14a). In a frame of reference rotating with the angular velocity (frequency) Ω of the center of the epicycle (epicenter, or guiding center) these orbits are closed and they are approximately ellipses around the center of the epicycle (Fig. 14b).

If V_0 is the axisymmetric potential, we have $\Omega^2 = V_0''/r$ and $\kappa^2 = V_0'' + 3V_0'/r$. The frequency κ along the epicycle is called “epicyclic frequency”.

In most cases (spiral or barred galaxies) the frame of reference is not inertial, but rotates with angular velocity Ω_s . This is called “pattern velocity”. In this rotating frame the form of the galaxy is stationary. Then the two basic frequencies of a moving star are $(\Omega - \Omega_s)$ and κ . If the ratio $\kappa/(\Omega - \Omega_s) = (n/m)$ is rational, then we have a resonant periodic orbit in the rotating system. Two most important resonances are the Lindblad resonances $\kappa/(\Omega - \Omega_s) = \pm 2/1$ (+ inner Lindblad resonance, ILR, – outer Lindblad resonance, OLR). A third most important resonance is corotation (or particle resonance), where $\Omega = \Omega_s$. The angular velocity Ω of an axisymmet-



Astrophysics: Dynamical Systems, Figure 15

The curves Ω and $\Omega - \kappa/2$ in two cases: (1) with only one ILR (dotted) and (2) two ILR's (solid), for two different values of the pattern velocity Ω_s .

ric family is a function of the distance r . Ω is a monotonically decreasing function of r (Fig. 15).

Corotation and the inner and outer Lindblad resonances appear at the intersections of the line $\Omega = \Omega_s$ (Fig. 15) with the curves Ω , $\Omega - \kappa/2$ and $\Omega + \kappa/2$. There is only one distance corresponding to corotation and one distance corresponding to the outer Lindblad resonance OLR. However, depending on the model used, we may have one inner Lindblad resonance, or two ILRs, or no ILR at all.

The study of nonperiodic orbits on the plane of symmetry of a galaxy is realized by means of a surface of section.

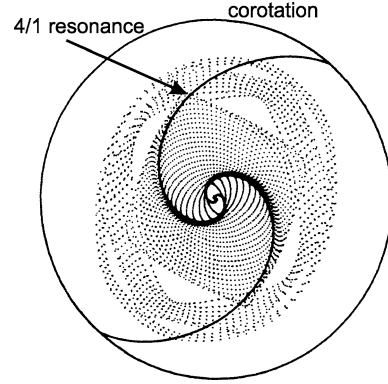
The invariant curves of the ordered nonperiodic orbits surround either the point x_1 or the point x_4 , that represent direct or retrograde periodic orbits that are reduced to circular orbits in the circular case.

The periodic orbits of type x_1 in the case of a spiral are like ellipses close to the inner Lindblad resonance (Fig. 16), but become like squares near the 4/1 resonance. These orbits and the nonperiodic orbits around them support the spiral up to the 4/1 resonance. However, beyond the 4/1 resonance these orbits are out of phase and do not support the spiral (Fig. 16). Thus, self-consistent spirals should terminate near the 4/1 resonance [20]. On the other hand, in strong bars we may have spirals outside corotation (see Sect. "Chaotic Orbits"). More details about orbits in galaxies can be found in [63] and [21].

Integrals of Motion in Spiral and Barred Galaxies

The Hamiltonian on the plane of symmetry of a galaxy (spiral or barred) rotating with angular velocity Ω_s is

$$H = \frac{\dot{r}^2}{2} + \frac{J_0^2}{2r^2} + V_0(r) - \Omega_s J_0 + V_1 \quad (30)$$



Astrophysics: Dynamical Systems, Figure 16

Periodic orbits in a spiral galaxy support the spiral up to the 4/1 resonance, but not beyond it

where \dot{r} is the radial velocity, J_0 the angular momentum, $V_0(r)$ the axisymmetric potential and V_1 the spiral (or barred) perturbation. The value of H is conserved (Jacobi constant). The Hamiltonian in action-angle variables is of the form $H = H_0 + V_1$, where

$$H_0 = \omega_1 I_1 + \omega_2 I_2 + a I_1^2 + 2b I_1 I_2 + c I_2^2 + \dots \quad (31)$$

and

$$\begin{aligned} V_1 &= \text{Re} \{ A(r) \exp[i(\Phi(r) - 2\theta)] \} \\ &= \text{Re} \sum_{mn} V_{mn}(I_1, I_2) \exp[i(m\theta_1 - n\theta_2)] \end{aligned} \quad (32)$$

The expansion in action-angle variables is found by expanding A , Φ and θ in terms of two angles, θ_1 (epicyclic angle) and θ_2 (azimuth of the epicenter measured from a certain direction), and $r - r_c = (2I_1/\omega_1)^{1/2}$, where r_c is the radius of a circular orbit that has the same Jacobi constant as the real orbit.

The main terms of V_1 are of order A and contain the trigonometric terms $\cos(\theta_1 - 2\theta_2)$, $\cos(\theta_1 + 2\theta_2)$ and $\cos 2\theta_2$. Away from the main resonances of the galaxy (inner and outer Lindblad resonances and corotation) it is possible to eliminate these terms by means of a canonical transformation that gives V_1 as a function of new action variables I_1^* , I_2^* . Thus, in this case the Hamiltonian H is a function of I_1^* , I_2^* only, i. e. I_1^* and I_2^* are integrals of motion.

The usual density wave theory of spiral structure deals with the forms of the integrals I_1^* , I_2^* , truncated after the first order terms.

However, the transformation from I_i to I_i^* contains denominators of the forms $\omega_1 - 2\omega_2$, $\omega_1 + 2\omega_2$ or ω_2 , which

tend to zero close to the inner and outer Lindblad resonances and corotation respectively. For example, near the inner Lindblad resonance we cannot eliminate the term with $\cos(\theta_1 - 2\theta_2)$, because the transformation would have the denominator $\omega_1 - 2\omega_2$ which is close to zero. In this case we eliminate only the terms with $\cos(\theta_1 + 2\theta_2)$ and $\cos 2\theta_2$. Instead of eliminating $\cos(\theta_1 - 2\theta_2)$, we write this term as $\cos \psi_1$, introducing now resonant action-angle variables $J_1 = I_1^*$, $J_2 = I_2^* + 2I_1^*$, $\psi_1 = \theta_1^* - 2\theta_2^*$, $\psi_2 = \theta_2^*$. Then H is expressed in terms of J_1 , J_2 and ψ_1 , but does not contain ψ_2 . Therefore, the conjugate action J_2 is an integral of motion. Thus, we have again two integrals of motion, namely J_2 and the Jacobi constant H .

The resonant form of H explains the forms of the orbits near the inner Lindblad resonance. In particular, near this resonance we have not one but two periodic orbits roughly perpendicular to each other.

The resonant theory is an extension of the linear density wave theory and it is applicable near the ILR. In a similar way we find the forms of the integrals near the outer Lindblad resonance and near corotation.

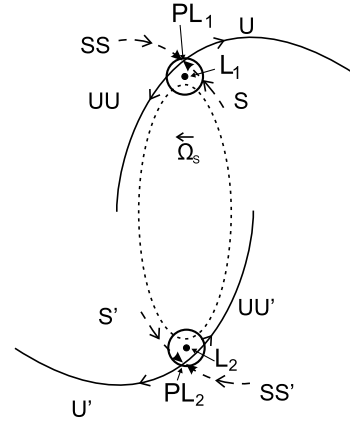
The three resonant forms of the Hamiltonian tend to the nonresonant form away from all resonances. In fact, if the amplitude of the spiral, or the bar, is small, the resonant effects are restricted in small regions around each resonance. However, if the amplitude of a spiral, or a bar, is large, we have an overlapping of resonances and the second integral is no more applicable.

Chaotic Orbits

Chaos in galaxies is always present near corotation, because this region contains an infinite number of higher order resonances that interact with each other.

Besides corotation, chaos may appear near the center of the galaxy when there is a large mass concentration (e.g. a central black hole). Furthermore large scale chaos appears when the amplitude of the spiral, or the bar, is large. Appreciable chaos has been found in N -body simulations of galaxies [27,58,68].

Chaos in barred galaxies near corotation appears around the unstable Lagrangian points L_1 and L_2 . Around these points there are short period unstable periodic orbits for values of the Jacobi constant larger than the value $H(L_1)$ appropriate for L_1 . Every unstable periodic orbit, PL_1 , PL_2 of this family has a stable (S etc) and an unstable (U etc) manifold attached to it. Orbits near L_1 or L_2 follow the unstable manifolds of the corresponding orbits PL_1 and PL_2 and form trailing spiral arms along U and U' (Fig. 17), which are clearly seen in strong bars [22,74].



Astrophysics: Dynamical Systems, Figure 17

Orbits starting close to the unstable orbits PL_1 , PL_2 move towards PL_1 , PL_2 along the stable asymptotic curves S , S' and away from PL_1 , PL_2 along the unstable asymptotic curves U , UU and U' , UU'

Thus, in strong bars chaos is important in explaining the outer spiral arms.

Future Directions

The subject of dynamical systems in astrophysics is progressing along three main directions: (1) Exploration of further basic phenomena, especially in systems of more than two degrees of freedom, (2) Rigorous mathematical theorems, and (3) Applications on various problems of astrophysics.

- (1) Much exploratory work exists now on systems of two degrees of freedom, but relatively little work has been done with systems of three or more degrees of freedom. In particular it is important to find the applicability of Arnold diffusion in various cases. A better separation of ordered and chaotic domains in phase-space is also necessary. Extensions of this work to find ordered and chaotic domains in quantum mechanical systems is also important. ([34] and references therein). Finally N -body problems with large N are also of great interest, especially as regards the validity and applications of statistical mechanics. In particular problems connected with the evolution of dynamical systems, are quite important.
- (2) Many problems that have been explored numerically up to now require rigorous mathematical proofs. For example, the use of formal integrals of motion, the application of basic theorems, like the KAM and the Nekhorosev theorems and the applicability of many statistical methods require a better mathematical foundation.

(3) Presently much is being done with astrophysical applications of the theory of dynamical systems. For example, galactic dynamics has experienced an important expansion in recent years, with the exploration of the role of chaos in galaxies, especially in the formation and evolution of the outer spiral arms, and in the evolution of the central black holes. There are many other astrophysical problems where order and chaos play an important role, like the structure and evolution of stars, stellar variability, solar and stellar activity, the role of magnetic fields in stars and galaxies, the properties of extrasolar planetary systems, and the evolution of the whole Universe.

As regards dynamical astronomy a list of 72 problems of current interest that require further serious analytical and numerical work (possible theses topics) is provided at the end of the book of Contopoulos [21], together with many references.

Bibliography

- Argyris J, Faust G, Haase M (1994) *An Exploration of Chaos*. North Holland, Amsterdam
- Arnold VI (1961) *Sov Math Dokl* 2:245; (1963) *Russ Math Surv* 18(5):9
- Arnold VI (1964) *Sov Math Dokl* 5:581
- Aubry S (1978) In: Bishop AR, Schneider T (eds) *Solitons and Condensed Matter Physics*. Springer, New York, p 264
- Birkhoff GD (1927) *Dynamical Systems*. Am Math Soc, Providence
- Birkhoff GD (1931) *Proc Nat Acad Sci* 17:656
- Bishop JL (1987) *Astrophys J* 322:618
- Chapront-Touzé M, Henrard J (1980) *Astron Astrophys* 86:221
- Cherry TM (1924) *Mon Not R Astr Soc* 84:729
- Chirikov BV (1979) *Phys Rep* 52:263
- Chirikov BV, Keil E, Sessler AM (1971) *J Stat Phys* 3:307
- Collet P, Eckmann J-P (1980) *Iterated Maps on the Interval as Dynamical Systems*. Birkhäuser, Boston
- Contopoulos G (1958) *Stockholm Ann* 20(5)
- Contopoulos G (1960) *Z Astrophys* 49:273
- Contopoulos G (1963) *Astron J* 68:763
- Contopoulos G (ed) (1966) *The Theory of Orbits in the Solar System and in Stellar Systems*. IAU Symp 25. Academic Press, London
- Contopoulos G (1967) In: Hénon M, Nahon F (eds) *Les Nouvelles Méthodes de la Dynamique Stellaire*. *Bull Astron* 2(3):223
- Contopoulos G (1971) *Astron J* 76:147
- Contopoulos G (1983) *Lett Nuovo Cim* 37:149
- Contopoulos G (1985) *Comments Astrophys* 11:1
- Contopoulos G (2002) *Order and Chaos in Dynamical Astronomy*. Springer, New York; Reprinted 2004
- Contopoulos G (2008) In: Contopoulos G, Patsis P (eds) *Chaos in Astronomy*. Springer, New York, p 3
- Contopoulos G, Harsoula M (2008) *Int J Bif Chaos*, in press
- Contopoulos G, Vandervoort P (1992) *Astrophys J* 389:118
- Contopoulos G, Voglis N (1996) *Cel Mech Dyn Astron* 64:1
- Contopoulos G, Zikides M (1980) *Astron Astrophys* 90:198
- Contopoulos G, Efthymiopoulos C, Voglis N (2000) *Cel Mech Dyn Astron* 78:243
- Coullet P, Tresser C (1978) *J Phys* 39:C5–25
- Delaunay C (1867) *Theorie du Mouvement de la Lune*. Paris
- Deprit A, Henrard J, Rom A (1971) *Astron J* 76:273
- de Zeeuw T (1985) *Mon Not R Astr Soc* 216:273
- de Zeeuw T, Hunter C, Schwarzschild M (1987) *Astrophys J* 317:607
- Eddington AS (1915) *Mon Not R Astr Soc* 76:37
- Efthymiopoulos C, Contopoulos G (2006) *J Phys A* 39:1819
- Efthymiopoulos C, Contopoulos G, Voglis N, Dvorak R (1997) *J Phys A* 30:8167
- Efthymiopoulos C, Giorgilli A, Contopoulos G (2004) *J Phys A* 37:1831
- Feigenbaum M (1978) *J Stat Phys* 19:25
- Froeschlé C, et al. (1993) *Cel Mech Dyn Astron* 56:307
- Grassberger P, Procaccia I (1983) *Phys D* 9:189
- Greene JM et al. (1981) *Phys D* 3:468
- Heggie DC (1983) *Cel Mech* 29:207
- Hénon M (1969) *Quart J Appl Math* 27:291
- Hénon M (1976) *Commun Math Phys* 50:69
- Hénon M, Heiles C (1964) *Astron J* 69:73
- Hunter C (1988) In: Buchler et al (eds) *Integrability in Dynamical Systems*. N Y Acad Sci Ann 536:25
- Hunter C (1990) In: Buchler et al (eds) *Galactic Models*. N Y Acad Ann 596:187
- Kolmogorov AN (1954) *Dokl Akad Nauk SSSR* 98:527
- Landau LD, Lifshitz EM (1960) *Mechanics*, 1st edn; 3rd edn (1976) Pergamon Press, Oxford
- Laskar J (1993) *Cel Mech Dyn Astron* 56:191
- Lichtenberg AJ, Leiberman MA (1992) *Regular and Chaotic Dynamics*, 2nd edn. Springer, New York
- Lindblad PO (1960) *Stockholm Ann* 21:3; 21:4
- Lorentz EN (1963) *J Atmos Sci* 20:130
- Lynden-Bell D (1962) *Mon Not R Astr Soc* 124:1
- Lynden-Bell D (1998) In: Buchler et al (eds) *Nonlinear Dynamics and Chaos in Astrophysics*. N Y Acad Sci Ann 867:3
- Mather JN (1982) *Erg Theory Dyn Syst* 2:397; (1982) *Topology* 21:457
- May RM (1976) *Nature* 261:459
- Menjuk CR (1985) *Phys Rev A* 31:3282
- Merritt D, Valluri M (1999) *Astron J* 118:1177
- Moser J (1962) *Nachr Acad Wiss Göttingen II. Math Phys Kl*:1
- Moser J (1967) *Math Ann* 169:136
- Moser J (1968) *Mem Amer Math Soc* 81:1
- Nekhoroshev NN (1977) *Russ Math Surv* 32(6):1
- Ollongren A (1965) In: Blaauw A et al (eds) *Stars and Stellar Systems V Galactic Structure*. Univ of Chicago Press, Chicago, p 501
- Percival IC (1979) In: Month M, Herrera JC (eds) *Nonlinear Dynamics and the Beam-Beam Interaction*. Am Inst Phys, New York, p 302
- Pinotsis A (1988) In: Roy AE (ed) *Long Term Behaviour of Natural and Artificial N-Body Systems*. Kluwer, Dordrecht, p 465
- Poincaré H (1892) *Les Méthodes Nouvelles de la Mécanique Céleste* Gauthier Villars. Paris I (1892); Paris II (1893); Paris III (1899); Dover edn (1957)
- Rosenbluth MN, Sagdeev RA, Taylor JB, Zaslavsky GM (1966) *Nucl Fusion* 6:217
- Schwarzschild M (1993) *Astrophys J* 409:563

69. Shenker SJ, Kadanoff LP (1982) *J Stat Phys* 27:631
70. Siegel CL (1956) *Vorlesungen über Himmelsmechanik*. Springer, Berlin
71. Stäckel P (1890) *Math Ann* 35:91
72. Statler TS (1987) *Astrophys J* 321:113
73. Voglis N, Contopoulos G (1994) *J Phys A* 27:4899
74. Voglis N, Tsoutsis P, Efthymiopoulos C (2006) *Mon Not R Astr Soc* 373:280
75. Walker GH, Ford J (1969) *Phys Rev* 188:416
76. Weinacht J (1924) *Math Ann* 91:279
77. Whittaker ET (1916) *Proc Roy Soc Edinburgh* 37:95
78. Whittaker ET (1937) *A Treatise on the Analytical Dynamics of Particles and Rigid Bodies*, 4th edn. Cambridge Univ Press, Cambridge
79. Wintner A (1947) *Analytical Foundations of Celestial Mechanics*. Princeton Univ Press, Princeton