

# **Jornada de Iniciação Científica e Tecnológica do LNCC**

Petrópolis, 15 e 16 de outubro de 2020.

## **Laboratório Nacional de Computação Científica – LNCC**

Diretor

Augusto Cesar Gadelha

Coordenação de Gestão e Administração - COGEA

Sergio Ferreira de Figueiredo

Coordenação de Métodos Matemáticos e Computacionais - COMAC

Frédéric Gerard Christian Valentin

Coordenação de Modelagem Computacional - COMOD

Márcio Arab Murad

Coordenação de Pós-Graduação e Aperfeiçoamento - COPGA

Artur Ziviani

Coordenação de Tecnologia da Informação e Comunicação - COTIC

Wagner Vieira Léo

Programa Institucional de Bolsas de Iniciação Científica&

Programa Institucional de Bolsas de Iniciação em Desenvolvimento Tecnológico e Inovação  
Marcos Garcia Todorov

## **Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq**

Presidente

**Evaldo Ferreira Vilela**

Coordenadora Geral do PIBIC/PIBITI

**Lucimar Batista de Almeida**

# **Jornada de Iniciação Científica e Tecnológica do LNCC**

## **Comissão Interna do PIBIC/PIBITI-LNCC**

Marcos Garcia Todorov  
Eduardo Lucio Mendes Garcia  
Helio José Corrêa Barbosa  
Jack Baczynski

## **Avaliadores Externos**

Franklin Marquezino - UFRJ  
Carlos Cristiano Hasenklever Borges - UFJF

# **Apresentação**

O LNCC realiza este ano a XVII Edição da Jornada de Iniciação Científica e Tecnológica, que é um fórum de divulgação das pesquisas desenvolvidas no contexto dos Programas Institucionais de Bolsas de Iniciação Científica (PIBIC) e de Bolsas de Iniciação Tecnológica (PIBITI) fomentados pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq). No período de agosto de 2019 a julho de 2020, o PIBIC e PIBITI congregaram alunos de várias instituições de ensino e de diversas áreas do conhecimento. Este volume apresenta os resumos dos trabalhos desenvolvidos pelos bolsistas no período. Durante a Jornada, os trabalhos são apresentados pelos bolsistas oralmente e avaliados por um comitê científico externo.

Nesta XVII Edição da Jornada, o Comitê Externo de Avaliação do PIBIC/PIBITI tem a seguinte composição:

Prof. Franklin Marquezino - UFRJ

Prof. Carlos Cristiano Hasenlever Borges - UFJF

Destacamos o papel relevante do PIBIC/PIBITI do LNCC no desenvolvimento das pesquisas no LNCC e, principalmente, na formação complementar dos bolsistas, promovendo o aprimoramento do conhecimento, espírito criativo, reflexão crítica e ética. Estas características têm contribuído para suas inserções no mercado de trabalho e em programas de pós-graduação, como o PPG em Modelagem Computacional do LNCC. Este é o resultado do esforço e dedicação de todos os participantes.

# Agradecimentos

Agradecemos ao CNPq pelas bolsas concedidas, à Direção do LNCC pelo apoio e à Comissão Interna do PIBIC e PIBITI no LNCC.

Agradecemos a disponibilidade e contribuição dos membros do Comitê Externo de Avaliação. O sucesso desta Jornada, e do Programa como um todo, é o resultado da dedicação e do esforço de toda a comunidade do LNCC. Expressamos em particular nosso reconhecimento ao apoio concedido pela secretaria do PPG-LNCC e, em particular, à Sra. Roberta Machado.

Marcos Garcia Todorov  
*Coordenador do PIBIC/PIBITI - LNCC*

## **Índice**

### **Bolsistas PIBIC ativos**

Simulação de Fluidos Direcionados por Campos Tensoriais para Animação

Bolsista: Arthur Gonze Machado

Orientador: Gilson Giraldi

Exploração de Módulos Paralelo-Híbrido de Bioinformática para Ambientes GP de Supercomputação

Bolsista: Guilherme Freire da Silva Dornelas

Orientadores: Carla Osthoff Ferreira de Barros, Kary Ann del Carmen Ocaña Gautherot e André Elias Rodrigues Soares

Projeto eImplementação de Workflows Científicos Reprodutíveis de Alto Desempenho

Bolsista: Lucas da Cruz Silva

Orientador: Luiz Gadelha

Método dos Volumes Finitos aplicados para a solução de problemas de mudança de fase

Bolsista: Ophir Valiante Duarte Neto

Orientador: Elson Toledo e Maicon William Niebus Odone

Inversão de Dados Sísmicos.

Bolsista: Vinícius Theobaldo Jorge

Orientador: Marcio Rentes Borges

Uso de Aprendizado de Máquina para Melhorias no Desempenho e no Consumo de Energia em Sistemas HPC

Bolsista: Vitor Carlos Soares Sá

Orientadores: Bruno Schulze e Mariza Ferro

Análise de Desempenho e Características de Paralelização dos Algoritmos de Aprendizado de Máquina

Bolsista: Vitor Goncalves Vieira

Bolsista: Bruno Schulze e Mariza Ferro

### **Bolsistas PIBITI ativos**

Aplicação das Ferramentas Intel Parallel Studio para modernização de código para métodos numéricos de diferenças finitas para solução de equações diferenciais parciais em arquitetura Intel Haswel/Broadwell

Bolsista: Gabriel Pinheiro da Costa

Orientador: Carla Osthoff e Frederico Luis Cabral

Gerência de Aplicações Científicas no Portal da Rede Nacional de Bioinformática

(Bioinfo-Portal)

Bolsista: Mayconn Luiz Bispo dos Santos

Orientador: Kary Ann del Carmen Ocaña Gautherot, Antonio Tadeu Azevedo Gomes e Marcelo Monteiro Galheigo

## **Ex bolsistas**

Implementação de estrutura de indexação de dados para otimizar as consultas no SAVIME

Bolsista: André Muniz Demori

Orientadores: Fabio Porto e Douglas Ericson de Oliveira

Bioinfo-BD: Integração Analítica das Aplicações no Bioinfo-Portal com Bancos de Dados Relacionais

Bolsista: Guilherme da Silva Vieira

Orientadores: Kary Ann del Carmen Ocaña Gautherot, Fábio André Machado Porto e Douglas Ericson Marcelino de Oliveira

Avaliação de métodos de aprendizado de máquina em aplicações de Bioinformática

Bolsista: Isabela Canuto Ramos

Orientadores: Kary Ann del Carmen Ocaña Gautherot, Fábio André Machado Porto e Douglas Ericson Marcelino de Oliveira

Segmentação de Imagens Utilizando Algoritmos Bio-Inspirados e Redes Neurais com Aplicações para Imagens Medicas e Biológicas

Bolsista: Lucas Pampolin Laheras

Orientadores: Gilson Antonio Giraldi e Paulo Sérgio Silva Rodrigues

Algoritmos Quânticos e Criptogradia

Bolsista: Marcos Vinicius Ternavisk de Lima

Orientador: Renato Portugal

A interação da cevada com os nutrientes do solo e o pulgão afídeo

Bolsista: Priscila Luana Lopes de Barros Weisz

Orientador: Mauricio Vieira Kritz e Lucas dos Anjos

Avaliação de Métodos de Aprendizado em Aplicações de Saúde

Bolsista: Raquel de Abreu Junqueira Gritz

Orientadores: Fábio André Machado Porto e Douglas Ericson Marcelino de Oliveira

Simulação numérica e computacional do tráfego viário

Bolsista: René Constancio Nunes de Lima

Orientadores: Elson M. Toledo e Regina Célia P. Leal Toledo

Estudo de Modelo de Simulação e de Previsão do Tráfego Viário

Bolsista: Tiago Maia de Lacerda Brasil

Orientadores: Sandra Malta e Regina Célia P. Leal Toledo

Identificação de lentes gravitacionais utilizando técnicas de aprendizagem profunda

Bolsista: Viviane de Mattos Matioli

Orientador: Fabio Porto

# Relatório de Atividades: Simulação de Fluidos Direcionados por Campos Tensoriais para Animação

Bolsista: Arthur Gonze Machado

Fone: (32) 98503-4279, Email: arthur.gonze@ice.ufjf.br

Instituição do Bolsista: Universidade Federal de Juiz de Fora

Instituto de Ciências Exatas, Fone: (32) 2102-3301, Email: secretaria.ice@ice.ufjf.br

Instituição de Bolsa: Laboratório Nacional de Computação Científica (LNCC)

Coordenação de Métodos Matemáticos e Computacionais - (COMAC),

Fone: (24) 2233-6150, Email: mroberta@lncc.br

Orientador: Gilson Antonio Giraldi

Fone: (24) 2233-6088, Email: gilson@lncc.br

Co-orientador: Marcelo Bernardes Vieira (UFJF)

Fone: (32) 2102-3301, Email: marcelo.bernardes@ufjf.edu.br

Tipo da Bolsa: PIBIC-LNCC

Período do Relatório: 01/04/2020 a 31/07/2020

31 de julho de 2020

## 1 Objetivo

Este projeto tem como objetivo estudar e desenvolver novos métodos para animação de fluidos, com controle obtido via campos tensoriais. Desta forma, o objetivo primário e geral do trabalho é desenvolver um modelo de projeção anisotrópica robusto que viabilize o uso de campos tensoriais como forma efetiva de controlar o fluido. Neste contexto, destaca-se nesta etapa, que o principal foco do trabalho foi desenvolver um método de meta-heurística para grades regulares, visando obter campos tensoriais que reduzem erros residuais na operação de projeção.

## 2 Introdução

Métodos para controlar simulações de fluido têm sido foco de pesquisas há mais de uma década, tendo computação gráfica como uma das inúmeras aplicações. Métodos para animação de fluidos são computacionalmente custosos dada a natureza inherentemente complexa da mecânica de fluidos, tornando ainda mais difícil a criação de métodos robustos para controle da simulação sem perder a plausibilidade física [12].

Nesse projeto, temos o interesse em desenvolver meios naturais de controlar fluidos, tanto em relação ao caminho, quanto ao estado final. Para isso, foram utilizados campos tensoriais como parte inerente do meio. Foi empregada uma abordagem baseada em malha, na qual define-se um campo tensorial que permite desviar o fluido, alterando os vetores de velocidades de forma a atender os objetivos da animação. Essa abordagem vem sendo desenvolvida pelos orientadores dessa bolsa nos últimos dois anos

[9, 13], tendo como proposta central a utilização de tensores simétricos semidefinidos positivos de segunda ordem para controle de fluidos. O método é uma adaptação de [10], em que o campo tensorial é inserido na formulação matemática de forma que ele altera localmente o momento linear do fluido. Cada uma das etapas básicas da simulação (advecção, difusão e projeção) foi alterada para levar em consideração a influência do tensor, refletindo as equações adaptadas.

Tendo o trabalho de [13] como ponto de partida, durante o período deste relatório, foi desenvolvido uma adaptação da meta-heurística denominada Simulated Annealing para obter campos tensoriais definidos positivos aplicando tendo em vista a decomposição de Helmholtz anisotrópica em 2D. Além disso, também foi desenvolvido uma versão, ainda em teste, deste mesmo método para um software de simulação de fluídos em 3D.

## 2.1 Fundamentos

### 2.1.1 Projeção Anisotrópica de Helmholtz

Dado um campo de velocidades arbitrário  $\mathbf{u}(\mathbf{r})$  obtido através da solução da equação de conservação de momento, a operação de projeção é o processo de encontrar um campo vetorial  $\mathbf{w}(\mathbf{r}) = P(\mathbf{u}(\mathbf{r}))$ , que satisfaça a condição de incompressibilidade  $\nabla \cdot \mathbf{w} = 0$ . Muitos modelos de simulação forçam a incompressibilidade usando a decomposição clássica de Helmholtz-Hodge [2].

O campo anisotrópico, consequentemente, necessita de um método de projeção próprio. Como discutido em [9], a decomposição isotrópica de Helmholtz-Hodge pode violar as restrições locais, representadas pelo tensor. A projeção deve ter como resultado um campo vetorial que não só é livre de divergente, mas também atende as restrições de troca impostas pelo campo tensorial. Considere o  $\mathbf{S}$ -divergente e o  $\mathbf{T}$ -divergente aplicado a um campo vetorial arbitrário  $\mathbf{f}(\mathbf{r})$ :

$$\nabla_{\mathbf{T}} \cdot \mathbf{f}(\mathbf{r}) = \nabla_{\mathbf{T}} \cdot \nabla_{\mathbf{T}} \phi(\mathbf{r}) + \nabla_{\mathbf{T}} \cdot \nabla_{\mathbf{S}} \times \mathbf{a}(\mathbf{r}), \quad (1)$$

$$\nabla_{\mathbf{S}} \cdot \mathbf{f}(\mathbf{r}) = \nabla_{\mathbf{S}} \cdot \nabla_{\mathbf{T}} \phi(\mathbf{r}) + \nabla_{\mathbf{S}} \cdot \nabla_{\mathbf{S}} \times \mathbf{a}(\mathbf{r}). \quad (2)$$

O objetivo é encontrar um campo vetorial livre de divergente através destas decomposições. Note que apenas a Equação 2 pode nos dar um campo vetorial livre de divergente. Se ambos  $\mathbf{S} = \mathbf{I}$  e  $\mathbf{T} = \mathbf{I}$ , a decomposição é  $\nabla \cdot \mathbf{f}(\mathbf{r}) = \nabla \cdot \nabla \phi(\mathbf{r}) + \nabla \cdot \nabla \times \mathbf{a}(\mathbf{r}) = \nabla \cdot \nabla \phi(\mathbf{r})$ , e o campo vetorial do termo  $\nabla \times \mathbf{a}(\mathbf{r})$  é livre de divergente em respeito a base canônica. Esta é a decomposição isotrópica de Helmholtz-Hodge.

Nosso interesse é usar campos tensoriais de forma que  $\mathbf{S} \neq \mathbf{I}$  ou  $\mathbf{T} \neq \mathbf{I}$ . Nesse projeto, foi procurado soluções usando apenas um campo tensorial diferente da identidade. Primeiramente, considere  $\mathbf{T} = \mathbf{I}$  e  $\mathbf{S} \neq \mathbf{I}$ , reduzindo a Equação 2 para:

$$\nabla_{\mathbf{S}} \cdot \mathbf{f}(\mathbf{r}) = \nabla_{\mathbf{S}} \cdot \nabla \phi(\mathbf{r}) + \nabla_{\mathbf{S}} \cdot \nabla_{\mathbf{S}} \times \mathbf{a}(\mathbf{r}) = \nabla_{\mathbf{S}} \cdot \nabla \phi(\mathbf{r}),$$

que pode gerar um campo vetorial livre de divergente  $\nabla_{\mathbf{S}} \times \mathbf{a}(\mathbf{r})$ . No entanto, esse campo é solenoidal em respeito ao campo tensorial  $\mathbf{S}(\mathbf{r})$ . De forma alternativa, ao definir  $\mathbf{T} \neq \mathbf{I}$  e  $\mathbf{S} = \mathbf{I}$ , obtém-se

$$\nabla \cdot \mathbf{f}(\mathbf{r}) = \nabla \cdot \nabla_{\mathbf{T}} \phi(\mathbf{r}) + \nabla \cdot \nabla \times \mathbf{a}(\mathbf{r}) = \nabla \cdot \nabla_{\mathbf{T}} \phi(\mathbf{r}).$$

que pode gerar um campo livre de divergente  $\nabla \times \mathbf{a}(\mathbf{r})$ , solenoidal em respeito à base canônica. Desta forma, a projeção anisotrópica  $P_{\mathbf{T}}(\cdot)$  de um campo vetorial arbitrário  $\mathbf{f}(\mathbf{r})$  restrinido pelo campo tensorial  $\mathbf{T}(\mathbf{r})$  é obtida solucionando a equação  $\mathbf{T}$ -Poisson

$$\nabla \cdot \nabla_{\mathbf{T}} \phi(\mathbf{r}) = \nabla \cdot \mathbf{f}(\mathbf{r}), \quad (3)$$

de modo a obter o campo vetorial projetado como:

$$P_{\mathbf{T}}(\mathbf{f}(\mathbf{r})) = \mathbf{f}(\mathbf{r}) - \nabla_{\mathbf{T}} \phi(\mathbf{r}), \quad (4)$$

que é livre de divergente em respeito a base canônica e sujeito à anisotropia de  $\mathbf{T}$  e adequado para advecção incompressível em coordenadas cartesianas.

### 2.1.2 Equações Anisotrópicas para Fluidos Incompressíveis

Como em [9], mantém-se a ideia de que o tensor deflete localmente as velocidades, alterando o momento em um processo que pode ser parametrizado para atingir o controle desejado.

Seja  $\mathbf{T}$  um campo tensorial estacionário, simétrico e definido positivo, com autovalores  $\{t_1, t_2, t_3\}$  e autovetores  $\{\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3\}$ . Como usual, as quantidades dinâmica envolvidas são a velocidade do fluido  $\mathbf{u}$ , densidade  $\rho$ , pressão  $p$ , e um campo de força externa  $\mathbf{g}$ . A principal ideia é modificar as equações de momento e massa usando a direção do campo tensorial e o termo de força externa, de modo a aplicar a projeção anisotrópica ao invés da tradicional usada em [9]. Deste modo, a decomposição anisotrópica de Helmholtz torna-se parte central da formulação contínua, definida pelas seguintes expressões [13]:

$$\frac{\partial \mathbf{u}}{\partial t} = -\mathbf{u} \cdot \nabla \mathbf{u} - \frac{1}{\rho} \nabla_{\mathbf{T}} p + \nu \nabla \cdot \nabla_{\mathbf{T}} \mathbf{u} + \mathbf{T} \mathbf{u} - \beta \mathbf{u} + \mathbf{g}, \quad (5)$$

$$\nabla \cdot \mathbf{u} = 0, \quad (6)$$

$$\frac{\partial \rho}{\partial t} = -\mathbf{u} \cdot \nabla \rho + \theta \nabla \cdot \nabla_{\mathbf{T}} \rho - \alpha \rho + s, \quad (7)$$

com condições iniciais  $\mathbf{u}(\mathbf{r}, 0) = \mathbf{0}$  e  $\rho(\mathbf{r}, 0)$  arbitrárias, para  $\mathbf{r} \in \Omega \subset \mathbb{R}^3$ , onde a viscosidade  $\nu$  e o parâmetro  $\theta$  controlam a influência de termos difusivos para a velocidade e densidade, respectivamente, que são afetados pelo campo tensorial  $\mathbf{T}$ . Além disso,  $s$  representa a fonte de densidade e o parâmetro real  $\alpha$  pondera a taxa de matéria que desaparece ao longo da simulação. O operador  $\mathbf{T}$ -gradiente ( $\nabla_{\mathbf{T}}$ ) é definido para solucionar a expressão (3).

Esta formulação é usada para demonstrar como a projeção anisotrópica pode beneficiar o controle do fluido, mas o método desenvolvido pode ser adaptado para outros modelos de simulação.

A equação (3) pode ser resolvida usando condição de contorno do Dirichlet  $\phi(\mathbf{r}) = h(\mathbf{r})$ , com  $h$  sendo uma função suave na fronteira  $\partial\Omega$ . Podemos utilizar também a condição de fronteira de Neumann, porém esta demanda uma análise mais cuidadosa, porque o efeito do campo tensorial  $\mathbf{T}(\mathbf{r})$  na solução próximo da fronteira  $\partial\Omega$  deve ser considerado.

### 2.1.3 Simulated Annealing

Simulated Annealing (ou Recozimento simulado), é um método probabilístico proposto por [7] e [4] com o propósito de encontrar o mínimo global de uma função custo que possa ter vários mínimos locais. Esse método simula o funcionamento de um processo físico onde um material é levado a uma alta temperatura e lentamente é resfriado até que eventualmente se solidifique, isso acontece quando ele se encontra em uma configuração de baixa energia [1]. Para que o Simulated Annealing seja utilizado, alguns elementos básicos devem ser verificados:

1. Um conjunto finito de estados  $\mathbf{E}$ ;
2. Uma função custo  $\mathbf{C} \in \mathbb{R}$  e definida em  $\mathbf{E}$ ;
3.  $\forall \mathbf{e} \in \mathbf{E}, \exists$  um conjunto de estados  $\mathbf{E} - \{\mathbf{e}\}$  chamado de conjunto de estados vizinhos a  $\mathbf{e}$ ;
4. Uma função decrescente de temperatura  $\mathbf{T}$ ;
5. Um estado inicial  $\mathbf{i} \in \mathbf{E}$ .

Com os elementos acima satisfeitos, o algoritmo de Simulated Annealing consiste de uma cadeia de Markov discreta e não-homogênea [1]. A evolução do algoritmo acontece da seguinte forma: a partir de um estado inicial  $\mathbf{i}$ , escolha de forma aleatória um vizinho  $\mathbf{v}$  de  $\mathbf{i}$ . Uma vez que tenha escolhido um vizinho, é verificado o valor da função custo deste vizinho e este valor é comparado com o valor da função custo do estado original, se este valor custo for menor então este vizinho é selecionado como nosso estado atual e a busca se reinicia a partir dele. Caso o valor da função custo seja maior, a seguinte função ditará a probabilidade de aceitação do estado:

$$P = \exp\left(\frac{-\Delta C}{T}\right), \quad (8)$$

onde  $\Delta C$  representa a diferença de “energia” entre o estado vizinho e o estado atual. Essencialmente, isso é a distribuição de Boltzmann contribuindo para a função de particionamento da mecânica estatística do sistema [6].

### 3 Metodologia

O principal fato explorado nessa etapa do trabalho é a existência e unicidade da decomposição anisotrópica de Helmholtz [5]. Atualmente, temos desenvolvido um método para projeção em grades regulares. Desta forma, será possível avaliar se a utilização da projeção anisotrópica pode ser um veículo para reduzir erros residuais em uma projeção isotrópica equivalente. A ideia é inicialmente evidenciar a hipótese através do uso de meta-heurísticas.

Como primeiro objetivo, para melhor entendimento do assunto a ser trabalhado no projeto, foi feito uma revisão de conteúdos, principalmente em relação a base matemática necessária para o bom entendimento da dinâmica dos fluidos e funcionamento das equações de Navier-Stokes, mostradas na seção 2.1.2. Portanto, assuntos como cálculo vetorial, e isso inclui a decomposição de Helmholtz mostrada na seção 2.1.1, métodos numéricos e as equações incompressíveis de Euler presentes em [3] foram estudados. Tal livro foi escolhido pela abordagem destes temas com maior foco em simulação de fluidos.

Após essa etapa, o estudo da implementação desses conceitos foi o próximo passo, e artigos como [10] e [11] serviram como base para as primeiras implementações e um melhor entendimento da transposição da teoria para a prática, assim como os problemas enfrentados ao fazer essa transposição. A introdução de tensores e o papel deles na dinâmica dos fluidos ocorreu logo após essa etapa, e com isso o artigo [8] foi essencial para melhor visualização da atuação dos campos tensoriais no controle do fluído.

Por fim, foi feito um estudo sobre meta-heurísticas e, mais especificamente, sobre o Simulated Annealing mencionado na seção 2.1.3. Os artigos [1] e [6] serviram como base para este estudo e para a primeira implementação feita no software MatLab.

### 4 Resultados e Discussão

Por simplicidade, os primeiros códigos foram desenvolvidos para um ambiente 2D, visando obter resultados iniciais do impacto do Simulated Annealing para o problema do projeto. Essa meta-heurística possui várias versões e modificações presentes na literatura e portanto, para esse primeiro momento, o algoritmo implementado foi o padrão, também chamado de Boltzmann Annealing (BA), onde a função de aceitação probabilística de vizinhos é dada pela equação 8, e a função de temperatura é dada por uma generalização feita por Kirkpatrick a partir do algoritmo de Metropolis Monte Carlo dada por:

$$\text{Temp}(t) = \frac{1}{\log(t)}, \quad (9)$$

onde,  $t$  é uma medida de tempo artificial para o resfriamento da temperatura. A partir dessa implementação inicial foi possível observar alguns pontos que devem ser levados em consideração:

- O algoritmo possui muitos parâmetros a serem configurados e com isso gera uma dificuldade na adaptação do mesmo para um problema específico. Essa característica do algoritmo já foi abordada por [6] e no futuro do projeto pode ser ponto de partida para uma nova versão.
- O algoritmo acrescenta um custo computacional considerável para a execução da simulação. Uma análise mais criteriosa da complexidade e ganho na diminuição do divergente do campo vetorial deve ser feita.
- A ordem de grandeza do valor obtido na equação 9 deve ser a mesma da variação da energia para o correto funcionamento da equação 8.

Após esta primeira implementação em um ambiente 2D, e observando os resultados promissores, uma segunda implementação foi iniciada para um ambiente de simulação de fluidos em 3D. Essa segunda implementação continua em desenvolvimento dado a necessidade de uma análise criteriosa do correto funcionamento do algoritmo de simulação. A ideia é que, a partir da otimização de campos tensoriais 2D, possamos definir estratégias mais elaboradas para o caso 3D que tem uma complexidade de tempo esperada bem maior. Em uma avaliação preliminar, foi possível reduzir o erro do divergente nulo na operação de projeção de campos vetoriais controlados. Sendo assim, uma das tarefas a serem realizadas na sequência é uma avaliação mais elaborada de campos tensoriais gerados pelo método Simulated Annealing.

## 5 Conclusão e Trabalhos Futuros

Os resultados obtidos no ambiente 2D se mostraram promissores, e ressaltam a importância deste trabalho para a redução de erro do divergente ao se utilizar campos tensoriais como meio de controle de simulações de fluidos.

Para o futuro do projeto é necessário verificar a corretude do método implementado visto os problemas de adaptação do Simulated Annealing. Após essa verificação, será possível comparar os três tipos de projeção implementados: projeção clássica, projeção com campo tensorial e projeção com campo tensorial e Simulated Annealing. E assim verificar o desempenho dos métodos implementados na redução de erro de divergente no processo de projeção de campos vetoriais arbitrários. Essas atividades deverão ser feitas em conjunto com decisões tomadas em reuniões com os orientadores. É importante salientar que relatórios técnicos deverão ser gerados no decorrer do projeto para documentação e preparação futura de artigos. Por fim, é esperado que com os resultados obtidos seja possível a elaboração e submissão de um artigo qualificado no fim de 2020 ou no início de 2021.

## Referências

- [1] Dimitris Bertsimas, John Tsitsiklis, et al. Simulated annealing. *Statistical science*, 8(1):10–15, 1993.
- [2] Robert Bridson. *Fluid simulation for computer graphics*. AK Peters/CRC Press, 2015.
- [3] Robert Bridson. *Fluid simulation for computer graphics*. CRC press, 2015.
- [4] Vladimír Černý. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of optimization theory and applications*, 45(1):41–51, 1985.
- [5] George Dassios and Ismo V Lindell. Uniqueness and reconstruction for the anisotropic helmholtz decomposition. *Journal of Physics A: Mathematical and General*, 35(24):5139, 2002.

- [6] Lester Ingber. Simulated annealing: Practice versus theory. *Mathematical and computer modelling*, 18(11):29–57, 1993.
- [7] Scott Kirkpatrick, C Daniel Gelatt, and Mario P Vecchi. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.
- [8] Jonathan Palacios, Chongyang Ma, Weikai Chen, Li-Yi Wei, and Eugene Zhang. Tensor field design in volumes. In *SIGGRAPH ASIA 2016 Technical Briefs*, pages 1–4. 2016.
- [9] Marcelo Caniato Renhe, Marcelo Bernardes Vieira, and Claudio Esperança. A stable tensor-based method for controlled fluid simulations. *Applied Mathematics and Computation*, 343:195–213, 2019.
- [10] Jos Stam. Stable fluids. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 121–128. ACM Press/Addison-Wesley Publishing Co., 1999.
- [11] Jos Stam. Real-time fluid dynamics for games. In *Proceedings of the game developer conference*, volume 18, page 25, 2003.
- [12] Adrien Treuille, Antoine McNamara, Zoran Popović, and Jos Stam. Keyframe control of smoke simulations. *ACM Transactions on Graphics (TOG)*, 22(3):716–723, 2003.
- [13] Marcelo Bernardes Vieira, Gilson Antônio Giraldi, Allan Carlos Amaral Ribeiro, Marcelo Caniato Renhe, and Claudio Esperança. Anisotropic helmholtz decomposition for controlled fluid simulation. *Applied Mathematics and Computation*, December 2019, Submitted.

## **RELATÓRIO DE ATIVIDADES**

**Título do Projeto:** Exploração de Módulos Paralelo-Híbrido de Bioinformática para Ambientes GPU de Supercomputação

**Nome do bolsista:**

Guilherme Freire da Silva Dornelas

**Nome do orientador:**

D.Sc. Carla Osthoff Ferreira de Barros (Tecnologista Sênior – CENAPAD/LNCC)

**Nome do coorientador:**

D.Sc. Kary Ann del Carmen Ocaña Gautherot (Tecnologista Pleno – LABINFO/LNCC)

Micaella Coelho (Bolsista PCI– CENAPAD/LNCC)

**Tipo de bolsa:** PIBIC

**Período do relatório:** 01/08/2019 a 31/07/2020

## **OBJETIVOS**

A presente proposta visa analisar o desempenho de aplicações CPU e GPU de bioinformática em ambientes de Computação de Alto Desempenho (CAD), mais especificamente usando como infraestrutura computacional o supercomputador Santos Dumont (SDumont, <https://sdumont.lncc.br/>).

O projeto apresenta três principais frentes:

- Migrar e acoplar os programas de bioinformática Beast e Beast2 para a arquitetura do SDumont.
- Explorar esses programas quando acoplados à biblioteca de alto desempenho BEAGLE 3 nos ambientes CPU e GPU do SDumont.
- Realizar análises de desempenho e escalabilidade, visando otimizar o uso do ambiente computacional.

## **INTRODUÇÃO**

A Computação de Alto Desempenho (CAD) é um termo ligado ao uso de supercomputadores ou clusters que executam atividades com uma grande demanda de processamento e.g. cálculos numéricos complexos. Como forma de colaborar nesse processo tem sido aplicado o uso de GPU (Unidade de Processamento Gráfico), para tornar a execução mais dinâmica [2].

O presente projeto envolve a colaboração em conjunto do Centro Nacional de Processamento de Alto Desempenho (CENAPAD, <http://www.cenapad-rj.lncc.br/>) e o Laboratório de Bioinformática (LABINFO, <https://www.labinfo.lncc.br/>) do Laboratório Nacional de Computação Científica (LNCC, <https://www.lncc.br/>). LABINFO faz o uso de várias aplicações que são utilizadas pela comunidade científica e são consideradas tempo e computacionalmente intensivas. Prévias análises de desempenho de aplicações de bioinformática como o RAxML em ambientes de CPU [1] foram realizadas com sucesso dentro do LNCC, obtendo resultados interessantes sobre o acoplamento e

otimização do tempo computacional desse tipo de programas em ambientes CAD. A nossa motivação foi explorar os ambientes GPU com programas de bioinformática como o Beast, Beast2 e Mr. Bayes de maneira a otimizar o uso desses programas em ambientes de supercomputador, dando apoio a pesquisadores de bioinformática para obter os resultados de maneira eficiente e em tempo viável [3-5].

Existe um interesse crescente pela comunidade de bioinformática em desenvolver aplicações GPU e explorar esses ambientes de forma eficiente. GPU diminui significativamente o tempo de execução de aplicações com operações na criação de elementos geométricos, aplicação de cores, inserção de efeitos. O SDumont permitirá a alocação dessas tarefas e será realizado o estudo do comportamento em termos de eficiência e desempenho desses programas de bioinformática usando ferramentas de perfilamento as quais estão instaladas no SDumont.

## METODOLOGIA

Beast é um pacote multi-plataforma de análise Bayesiana para sequências biológicas que usa o algoritmo Markov chain Monte Carlo (MCMC) para calcular o espaço da árvore e pondera à sua probabilidade posterior [4]. Beast é considerado tempo e computacionalmente intensivo, pois realiza o cálculo exaustivo das probabilidades nas árvores. Pacotes de bioinformática como Beast e o MrBayes podem ser acoplados a ambiente de CAD por meio do uso da biblioteca BEAGLE [3], estratégia que foi usada neste projeto de pesquisa.

A seguir detalhamos as atividades que foram realizadas, como relatadas no projeto de pesquisa original e incluímos uma nova atividade (7) relacionada à publicação dos resultados em workshop nacional.

1. Acoplar os programas de filogenia baseados em inferência Bayesiana Beast, Beast2 e MrBayes no ambiente do SDumont.
2. Desenhar um experimento em larga escala que consuma grande volume de dados genômicos, visando a distribuição e paralelismo de tarefas.
3. Melhorar o desempenho, escalabilidade e usabilidade dos programas com a biblioteca de CAD para filogenética estatística, BEAGLE 3.
4. Realizar análises de desempenho e escalabilidade em larga escala nos ambientes CPU e GPU do SDumont.
5. Mais especificamente, realizar uma análise comparativa do Beast acoplado ao BEAGLE 3, nas suas versões CPU, GPU e CPU/GPU.
6. Modelar um workflow científico de filogenia envolvendo esses programas, visando uma melhor gerência das execuções no SDumont.

## RESULTADOS

A proposta inicial foi desenvolver pesquisas para entender cada parte do projeto, alguns artigos foram utilizados como referência:

- Avaliação do RAxML no Supercomputador Santos Dumont [1], que realiza análises comparativa sobre o impacto produzido pela variabilidade nas configurações de ambiente, programas e características dos dados genômicos. Esse estudo foi realizado em CPU;
- Artigo de Simulação em algoritmos MCMC [2]: Estudo Comparativo e Simulações, MCMC: Compreendem uma classe de algoritmos para amostragem a partir de uma distribuição de probabilidade. Muito utilizado em estatística bayesiana, física computacional, biologia computacional e linguística computacional, usado pelos programas de inferência Bayesiana Beast e MrBayes;

- Artigo BEAGLE 3: Improved Performance, Scaling, and Usability for a High-Performance Computing Library for Statistical Phylogenetics [3], a Beagle é uma biblioteca de alto desempenho, pode fazer uso de processadores altamente paralelos, como aqueles em placas gráficas (GPUs), a mesma trabalha em conjunto com o Beast, Beast2 e MrBayes, fazendo a alternância entre os cálculos de verossimilhança em CPU e GPU.

Os principais programas que fazem parte do projeto: Beast, Beast2 e MrBayes e as ferramentas de perfilagem do SDumont, foram exploradas, buscando o entendimento de cada uma, suas variações com CPU e GPU, para assim entender sobre o comportamento em ambientes de CAD e encontrar aqueles cenários que ofereçam melhor resultado. Dessa maneira, esses estudos de desempenho desses programas em GPU no SDumont, podem levar a um uso eficiente e racional do ambiente computacional e a um melhor entendimento de todas as especificações técnicas dos programas e parâmetros.

O uso da biblioteca de alto desempenho BEAGLE permite executar os cálculos centrais do algoritmo Bayesiano da maioria dos pacotes de filogenética. A biblioteca é aplicada junto aos programas para análise Bayesiana como o Beast. No presente projeto o Beast e o BEAGLE foram acoplados e testes de desempenho e arquiteturas CPU e GPU foram realizadas no SDumont. Os resultados foram publicados na forma de artigo completo no Workshop Nacional e-Science (BreSci 2020) e será realizada uma apresentação oral dos resultados:

- Ocaña, K.A.C.S.; Coelho, M.; Freire, G.; Osthoff, C. High-Performance Computing of BEAST/BEAGLE in Bayesian Phylogenetics using SDumont Hybrid Resources. In: Brazilian e-Science Workshop (BreSci 2020), 2020, Cuiabá, Mato Grosso. Anais do XIV Brazilian e-Science Workshop, 2020.

## CONCLUSÕES

Programas de inferência Bayesiana são computacionalmente intensivos, especialmente se grandes volumes de dados são executados. A exploração de ambientes de computação de alto desempenho é uma tarefa importante em experimentos em larga escala. Estudos de desempenho foram realizados no programa de inferência Bayesiana Beast quando acoplada à biblioteca BEAGLE 3 em arquiteturas CPE e GPU no ambiente do SDumont.

Um dos desafios deste projeto foi acoplar a biblioteca BEAGLE 3 aos programas e explorar os parâmetros que ofereçam melhor ganho computacional no ambiente SDumont. Foram utilizados diversos parâmetros do Beast e BEAGLE relacionados a exploração de ambientes de CAD. A continuação da pesquisa está relacionada a exploração de ambientes multiGPU e realizar uma execução em larga escala das implementações finais: em uma única GPU e multiGPU.

## REFERÊNCIAS BIBLIOGRÁFICAS

1. K. Ocaña, J. Pedro, M. Coelho, M. Galheigo, e C. Osthoff, “Avaliação do uso eficiente de algoritmos paralelos de filogenia em supercomputadores,” in Anais do 12o BreSci - Brazilian eScience Workshop, Natal, RN, 2018.
2. C.-L. Hung, Y.-S. Lin, C.-Y. Lin, Y.-C. Chung, e Y.-F. Chung, “CUDA ClustalW: An efficient parallel algorithm for progressive multiple sequence alignment on Multi-GPUs,” Comput Biol Chem, vol. 58, pp. 62– 68, May 2015.
3. Z. Jin e J. D. Bakos, “Extending the BEAGLE library to a multi-FPGA platform,” BMC Bioinformatics, vol. 14, no. 1, p. 25, 2013.
4. M. A. Suchard, P. Lemey, G. Baele, D. L. Ayres, A. J. Drummond, and A. Rambaut. Bayesian phylogenetic and phylodynamic data integration using BEAST 1.10. Virus Evol., vol. 4, no. 1, 2018.

5. J. P. Huelsenbeck e F. Ronquist, “MRBAYES: Bayesian inference of phylogenetic trees,” *Bioinformatics*, vol. 17, no. 8, pp. 754–755, Aug. 2001.

## **Relatório de Atividades**

### **1. Dados gerais**

**Título do projeto:** Projeto e Implementação de Workflows Científicos Reprodutíveis de Alto Desempenho

**Bolsista:** Lucas da Cruz Silva

**Orientador:** Luiz Manoel Rocha Gadelha Júnior

**Tipo de bolsa:** Iniciação Científica

**Período do relatório:** 01/12/2019 - 31/07/2020

### **2. Objetivos**

A ciência de modo geral, no decorrer dos anos, tem se apoiado muito aos métodos de computação, a fim de obter soluções para experimentos de extrema complexidade. Esses experimentos, em geral, são de longa duração, utilizam de grandes volumes de dados e se compõem em etapas ou atividades dependentes entre si. Portanto, através da motivação de facilitar o gerenciamento (definição, execução e monitoração) de atividades extensivas, complexas e computacionalmente custosas, este trabalho tem por objetivo, modelar um workflow científico com processamento de alto desempenho (PAD), tomando como caso de estudo o experimento de bioinformática de sequenciamento RNA (RNA-Seq).

### **3. Introdução**

Em diversas áreas da ciência existem experimentos que podem ser computacionalmente intensivos, como é o caso do experimento de bioinformática RNA-Seq. Esse experimento trata da análise de expressão de genes e sequenciamento genômico [Love et al. 2014]. É um típico experimento em que todas as suas etapas se encadeiam entre si. Podendo, inclusive, ser classificado como um experimento Big Data, pois cada etapa possui uma atividade, que produz e consome um grande volume de dados. Além disso, podem levar dias para que o cientista obtenha os resultados finais do experimento. Sob essa condição, a ciência tem se apoiado muito em procedimentos e uso de infraestruturas computacionais no decorrer dos anos [Cordeiro et al. 2013].

É através dessa conexão, das mais diversas áreas da ciência com a área da computação, que surge o termo conhecido como e-Science, que propõe um apoio computacional a pesquisadores de diversas vertentes de pesquisa, de modo que produções científicas possam ser realizadas em larga escala [Braghetto and Cordeiro 2014]. Utilizando, comumente, os métodos mais sofisticados de computação, tal como Computação de Alto Desempenho (CAD) [Mattoso et al. 2008].

O *workflow científico* que é, basicamente, uma forma de representar computacionalmente experimentos científicos, é, também, uma das importantes ferramentas da e-Science. São utilizados por pesquisadores para obtenção de resultados melhores e mais eficientes, por meio da automatização dos processos experimentais [Braghetto and Cordeiro 2014].

Além disso, concedem benefícios acerca da extração dos dados gerados pelo experimento (dados de proveniência) e a garantia de que todas as atividades sejam executadas e finalizadas adequadamente. Para isso, é necessário utilizar um sistema que forneça

a infraestrutura para definir, executar e monitorar o workflow, chamado de Sistema de Gerência de Workflows Científicos (SGWfC) [Mattos et al. 2008]. O SGWfC permitirá que toda a modelagem abstrata feita pelo projetista seja traduzida a nível de máquina.

## 4. Metodologia

Esta seção se divide em quatro tópicos: A modelagem do workflow científico e o SGWfC utilizado; As configurações do supercomputador utilizado para executar o workflow; e, por fim, as informações sobre configuração e disposição dos dados utilizados.

### 4.1. Modelagem do Workflow Científico RNA-Seq

Para se chegar a versão final de um workflow científico, inicialmente, é requerido que as atividades ou etapas do experimento sejam mapeadas. Ou seja, fazer uma abstração de todos os procedimentos que o cientista realizaria sem a utilização de um ambiente computacional, a fim de fazer a modelagem do workflow. A modelagem requer o conhecimento de quantas atividades são realizadas, qual a ordem dessas atividades e as dependências entre elas.

O workflow RNA-Seq, possui três atividades que são executadas por dependência, na seguinte seqüência: Bowtie, é um software complexo e computacionalmente intensivo, utilizado para alinhar o sequenciamento [Langmead and Salzberg 2012]. Permite, inclusive, execução multithread; A atividade HTSeq, que é utilizada para analisar de forma estatística os dados sequenciados [Anders et al. 2014]; e, por fim, a atividade DESeq2 que nada mais é do que um script R que compara todos os dados estatísticos gerados pela atividade anteriormente citada [Love et al. 2014].

#### 4.1.1. Parsl - Parallel Script Library

Parsl é uma biblioteca da linguagem de programação Python que, como um SGWfC, é capaz de executar workflows com processamento de alto desempenho [Babuji et al. 2019]. Para isso, é necessário definir e indicar quais são as etapas que devem ser executadas de forma concorrente. Ao prover o paralelismo, respeita as dependências dos dados entre elas, mostrando um diferencial em relação a gerência da dependência do fluxo de dados do workflow [Silva et al. 2014]. O interessante em utilizar essa ferramenta é que pelo uso da linguagem de alto nível é possível ter uma maior abstração das etapas do workflow.

Além disso, é uma ferramenta que, em termos computacionais, permite fácil troca de ambiente, ou seja, ela se comporta da mesma forma seja em uma máquina local ou em um ambiente remoto (cluster, nuvem, supercomputadores, etc.). Isso porque a biblioteca Parsl foi projetada justamente para abstrair os detalhes de nível mais baixo da arquitetura, por este motivo, o Parsl utiliza um arquivo de especificação, em que o programador pode indicar qual provedor e qual modelo de execução deseja usar.

Na primeira etapa da pesquisa está sendo utilizado um modelo de execução chamado, ThreadPoolExecutor. Nesse modelo de execução, que dá suporte à execução multithreading em recursos locais, o programador pode indicar um número máximo de threads que o mesmo deseja alocar. Para o Parsl, isso também indica quantas tarefas poderão

ser executadas de forma concorrente. Uma vez que a forma de implementação de um modelo que dê suporte a threads seja, além de trabalhoso, impactante no desempenho computacional [Silva and Yokoyama 2011]. O uso desse modelo proporciona ao programador ganhos de produtividade e redução na ocorrência de erros, pois não existe preocupação com gerência de threads.

#### 4.2. Configuração do SDumont

O supercomputador Santos Dumont<sup>1</sup> - SDumont - é um dos supercomputadores mais poderosos da América Latina e está entre as 500 máquinas mais poderosas do mundo. Possui uma capacidade de processamento instalada de 5,1 Petaflop/s por segundo com 34.688 CPUs multi-cores, distribuídas em 1132 nós computacionais que são interligados por uma rede de interconexão Infiniband FDR/HDR isso proporciona um alto rendimento e uma baixa latência para a comunicação entre os processos e também no acesso ao sistema de arquivos, Lustre. O sistema de arquivo paralelo Lustre, é integrado a rede InfiniBand, dispondo de uma capacidade bruta de armazenamento em torno de 1,7 PBytes, juntamente com um sistema de arquivos secundários, com capacidade bruta de 640 TBytes.

O SDumont contém, também, dispositivos com arquitetura many-core, tanto GPU - Graphics Processing Unit - quanto a MIC - Many Integrated Cores. Os nós computacionais GPU possuem 2 x CPU Ivy Bridge Intel Xeon E5-2695v2 (12c @2,4GHz) e 64Gb de memória RAM e GPU Nvidia K40. O nó MESCA2 é diferenciado, com 240 cores - núcleos - e uma arquitetura de memória compartilhada com uma capacidade de armazenamento de 6 Tb. Por fim, existe também um nó especial para aplicações de Inteligência Artificial dispondo de 8 GPUs NVIDIA Tesla V100-16Gb com NVlink, no total são 40.960 CUDA-core e 5.120 Tensor-core.

Inicialmente, para fins de melhor análise de desempenho computacional do workflow, os testes realizados até o momento são pela utilização de apenas um único nó, para 24 núcleos por tarefa.

#### 4.3. Dados de Entrada

A análise de expressão gênica é sempre feita dividindo os dados em grupos que representam a condição da amostra, positivos ou negativos. As informações sobre o grupo a que cada arquivo pertence foram disponibilizadas pelo bioinformata, assim como os arquivos de entrada. Essas informações são de extrema importância para manter a integridade dos dados futuramente gerados e, até mesmo, para a validação dos resultados finais, portanto, estarão presentes em um arquivo no formato CSV, que será utilizado na última atividade, DESeq2. A tabela 1 indica as informações e disposição desses arquivos.

Para os testes do workflow RNA-Seq foram utilizados seis dados de entrada e a divisão foi feita em dois grupos de três. Há de se notar que o número de dados de entrada e a divisão dos grupos podem variar conforme o experimento. Então, inicialmente, a análise de desempenho está ligada somente a esses seis arquivos, como será observado na próxima sessão.

---

<sup>1</sup>Disponível em <https://sdumont.lncc.br/>. Último acesso: 25/07/2020.

sampleName	fileName	condition
tissue control 1	SRR5445794.fastq	control
tissue control 2	SRR5445795.fastq	control
tissue control 3	SRR5445796.fastq	control
tissue wntup 1	SRR5445797.fastq	wntup
tissue wntup 2	SRR5445798.fastq	wntup
tissue wntup 3	SRR5445799.fastq	wntup

**Tabela 1. Informações sobre sobre os dados de entrada.**

## 5. Resultados e Discussão

Apesar das análises de desempenho e os estudos de escalabilidade ainda estarem sendo realizados, pode-se dizer, por enquanto, que a execução do workflow RNA-Seq foi bem sucedida quanto aos dados finais gerados, pois esses foram validados pelo bioinformata. Com isso, é possível constatar também que o tempo de execução do experimento foi reduzido, pois esse workflow, utilizando os mesmos dados de teste, é executado em cerca de 2 a 3 dias no sistema Galaxy<sup>2</sup> e com a utilização da ferramenta Parsl no supercomputador SDumont durou cerca de 11 horas no pior caso, conforme destacado na tabela 2.

Das três atividades presentes no workflow, somente uma delas possui recurso multithread, que é a atividade Bowtie. Portanto, as análises devem ser feitas sob três condições: a primeira é feita pela gerência paralelizada a partir das ferramentas do Parsl; a segunda, é pela paralelização apenas da atividade Bowtie; e, a terceira é a união das duas anteriores, ou seja, utilizar multithreading tanto pelo Parsl, quanto pela atividade Bowtie.

Conforme colocado na sessão anterior, o modelo de execução que está sendo utilizado na autal etapa de teste é o Thread Pool. Por essa razão, a 1<sup>a</sup> coluna da tabela 2, indica o número máximos de threads desse executor e, as demais, o tempo em minutos dos testes realizados. Algo a se complementar é que esses testes não exploram o recurso multithread da primeira atividade (Bowtie). No entanto, é possível notar que a ferramenta Parsl, como esperado, está paralelizando naturalmente o workflow.

Threads	Teste 1 (min)	Teste 2 (min)	Teste 3 (min)
1	644,984	643,228	em execução
2	308,4761	306,9106	307,9105
4	194,9675	195,1739	196,7201
6	138,3363	135,8258	135,846
12	107,0217	106,1943	107,281
18	106,1011	106,3006	107,689
24	106,7224	107,4749	107,1992

**Tabela 2. Relação do tempos em minutos, dos testes de execução, conforme o aumento do número máximo de threads da Thread Pool.**

Todos os testes estão sendo realizados na fila nvidia\_long, utilizando os recursos CPU. Na tabela 2, apenas três testes se fazem suficientes uma vez que o desvio padrão entre os casos de testes tem baixa dispersão.

<sup>2</sup>Disponível em <https://galaxyproject.org/>. Último acesso: 25/07/2020.

Apesar de ainda não terem sido realizados testes o suficiente, as tabelas 3 e 4 trazem um cenário do que acontece com a utilização do recurso multithread da atividade Bowtie. Nos dados dessas tabelas, os testes foram feitos mantendo fixa a Thread Pool, que é administrada pela ferramenta Parsl - no caso, para a tabela 3, o número máximo de threads é 6 e para a tabela 4, o máximo é 12 - e as threads que variaram são do recurso multithread do Bowtie.

Threads	Teste 1 (min)
2	95,7208
3	83,4526
4	78,0171
6	75,8384
12	75,0004
18	73,5365
24	76,8399

**Tabela 3. Thread Pool fixa em 6**

Threads	Teste 1 (min)
2	73,5745
3	74,5916
4	71,4997
6	71,067
12	72,2836
18	73,1788
24	75,4327

**Tabela 4. Thread Pool fixa em 12**

Com isso, é possível observar, que a melhor configuração de threads, até o momento, é a da tabela 4: 6 threads para a atividade Bowtie e 12 para o modelo Thread Pool do Parsl, com execução completa do workflow em torno de 1 hora e 11 minutos.

Um ponto importante a se destacar, também levantado na sessão anterior, é a questão da quantidade de dados de entrada. Ainda não foram realizados testes para um conjunto maior de dados, portanto, não é possível enxergar um padrão de comportamento do workflow para centenas de arquivos e qual seria, por exemplo, a melhor configuração de threads a medida que a quantidade de dados aumentam.

Cabe também ressaltar que a ferramenta Parsl ainda não está sendo utilizada em todo o seu potencial, pois, como levantado na sessão anterior, os testes hoje são feitos com a utilização de apenas um nó, para fins de análises de desempenho. O ideal seria utilizar os executores que dão suporte à tarefas que exigem alto rendimento, alocando múltiplos nós no supercomputador, como por exemplo o HighThroughputExecutor.

## 6. Conclusão

O estudo de caso do workflow científico RNA-Seq não foi finalizado ainda, pois apresenta uma atividade que também utiliza um recurso multithread, o que torna a análise complexa do ponto de vista da CAD. Então, por essa razão, ainda possui testes em execução. Por outro lado, a análise de bioinformática dos dados de saída gerados, foram validados. Comprovando, a redução no tempo de execução. Portanto, o workflow foi modelado pelo Parsl, acoplado ao SDumont e apresenta resultados parciais que se mostram consideráveis frente aos objetivos inicialmente propostos, de facilitar a execução de atividades extensivas, complexas e computacionalmente custosas através da utilização de workflows científicos com PAD. O código fonte para o workflow RNA-Seq está disponível no GitHub<sup>3</sup>.

Como passos futuros, após a finalização dos testes atuais, serão feitas execuções do workflow em larga escala, explorando mais os recursos do Parsl. Com execuções

<sup>3</sup>Disponível em <https://github.com/lucruzz/rna-seq>. Último acesso: 25/07/2020.

em larga escala, também pode ser feito os arquivos de configuração global que foram removidos na última atualização do SDumont, em 2019. Para que todas as configurações de filas do SLURM fiquem disponíveis aos usuários, apenas carregando um módulo pelo Parsl. Além do uso de perfiladores para fins de otimização, também será estudado a melhor forma de manipular e armazenar os dados de proveniência.

## Referências

- [Anders et al. 2014] Anders, S., Pyl, P. T., and Huber, W. (2014). HTSeq—a Python framework to work with high-throughput sequencing data. *Bioinformatics*, 31(2):166–169.
- [Babuji et al. 2019] Babuji, Y., Woodard, A., Li, Z., Katz, D. S., Clifford, B., Kumar, R., Lacinski, L., Chard, R., Wozniak, J., Foster, I., Wilde, M., and Chard, K. (2019). Parsl: Pervasive parallel programming in python. In *28th ACM International Symposium on High-Performance Parallel and Distributed Computing (HPDC)*.
- [Braghetto and Cordeiro 2014] Braghetto, K. R. and Cordeiro, D. (2014). Introdução à modelagem e execução de workflows científicos. *Atualizações em Informática. Ied. Porto Alegre: SBC*, pages 1–40.
- [Cordeiro et al. 2013] Cordeiro, D., Braghetto, K. R., Goldman, A., and Kon, F. (2013). Da ciência à e-ciência: paradigmas da descoberta do conhecimento. *Revista USP*, (97):71–81.
- [Langmead and Salzberg 2012] Langmead, B. and Salzberg, S. L. (2012). Fast gapped-read alignment with bowtie 2. *Nature methods*, 9(4):357.
- [Love et al. 2014] Love, M. I., Huber, W., and Anders, S. (2014). Moderated estimation of fold change and dispersion for rna-seq data with deseq2. *Genome biology*, 15(12):550.
- [Mattos et al. 2008] Mattos, A., Silva, F., Ruberg, N., and Cruz, M. (2008). Gerência de workflows científicos: uma análise crítica no contexto da bioinformática. *COPPE/UFRJ*.
- [Mattoso et al. 2008] Mattoso, M., Werner, C., Travassos, G., Braganholo, V., and Murta, L. (2008). Gerenciando experimentos científicos em larga escala. *SBC-SEMISH*, 8:121–135.
- [Silva and Yokoyama 2011] Silva, R. R. and Yokoyama, R. S. (2011). Avaliação do desempenho de threads em user level utilizando sistema operacional linux. *Revista de Informática Teórica e Aplicada*.
- [Silva et al. 2014] Silva, V., Oliveira, D., and Mattoso, M. (2014). Scicumulus 2.0: Um sistema de gerência de workflows científicos para nuvens orientado a fluxo de dados. *Sessão de Demos do XXIX Simpósio Brasileiro de Banco de Dados*.

# **RELATÓRIO DE TRABALHO DO BOLSISTA**

**Método dos Volumes Finitos aplicados para a solução de problemas de  
mudança de fase**

Ophir Valiante Duarte Neto (bolsista)

Elson Magalhães Toledo (orientador)

Maicon William Niebus Odone (coorientador)

## Objetivos

- Estudo de diferentes modelos e métodos numéricos relacionados aos processos de mudança de fase para aplicação da simulação computacional de problemas de cunho real.
- Exploração e estudo do código OpenFOAM com sua customização para o tratamento de problemas de mudança de fase.
- Estudo e utilização no *softwares* GMSH, para geração de malhas, e ParaView, para realização de pós processamento.
- Validação do OpenFOAM na resolução de diferentes problemas da mecânica do contínuo: escoamentos simples, escoamentos com mudança de fase, problemas de transferência de calor, condição de contorno de convecção e análise de tensões em sólidos com geometria axissimétrica.

Para estas finalidades, foram realizados testes *benchmarks* com experimentos consolidados na literatura e com problemas que possuem soluções analíticas.

## Introdução

Diversas aplicações da mecânica do contínuo envolvem processos como a solidificação e fusão de materiais, convecção natural, tensão e deformação em sólidos e outros fenômenos relacionados. Dada a complexidade dos fenômenos envolvidos nestes processos, a modelagem numérica-computacional é uma ferramenta poderosa e de baixo custo para compreender e simular de forma satisfatória problemas que possam envolver um ou mais destes fenômenos complexos. Um conjunto de modelos matemáticos representando alguns destes processos foi estudado, e implementado através do Método dos Volumes Finitos utilizando o pacote de código aberto OpenFOAM, voltado para a resolução de problemas de mecânica do contínuo, incluindo a dinâmica dos fluidos computacional. Algumas das funcionalidades básicas do OpenFOAM foram exploradas, relacionando algumas das vantagens e desvantagens em sua utilização, bem como as principais características oferecidas por ele.

Foram considerados problemas de *benchmark* da mecânica do contínuo com o intuito de apreender os conceitos por trás da modelagem computacional e simulação destes fenômenos, e também para a aplicação e validação do OpenFOAM na resolução destes problemas. Além disto, o código e a implementação do OpenFOAM foram estudados para melhor compreensão de seu funcionamento e estruturação. Por fim, também foi feito um estudo da literatura que contempla a dinâmica dos fluidos computacional, problemas de mudança de fase e problemas de mecânica dos sólidos, a fim de compreender bem os conceitos teóricos desses temas e avançar na implementação de um módulo para cálculo de tensões térmicas residuais relacionadas com a fase sólida em um processo de solidificação. O estudo contemplou uma gama relativamente ampla da mecânica do contínuo e da modelagem computacional destes problemas, bem como a exploração da documentação e de módulos do OpenFOAM que são pouco abordados.

## Materiais, Métodos e Metodologia

O desenvolvimento do projeto requereu o estudo das equações de Navier-Stokes acopladas com a equação da energia. A implementação computacional foi baseada no Método dos Volumes Finitos e no método da entalpia-porosidade para problemas de mudança de fase. Além disso, a implementação se baseia em algoritmos iterativos como o SIMPLE, PISO e o PIMPLE do OpenFOAM.

Um dos focos do trabalho foi o estudo sobre o *solver* do OpenFOAM buoyantPimpleFoam, que resolve escoamentos com transferência de calor usando o algoritmo iterativo PIMPLE. Para a resolução dos problemas de mudança de fase empregou-se o módulo solidificationMeltingSource. Este módulo tem a função de implementar diversos tipos de termos fonte, dentre eles, a implementação do método da entalpia-porosidade para problemas de mudança de fase. O acoplamento destes dois permite realizar a simulação de problemas de mudança de fase no OpenFOAM. Assim, todas as atividades realizadas tiveram como objetivo estudar e validar o funcionamento desta parte do *software*, que ainda não contempla uma boa bibliografia com informações e testes, ou pesquisar formas de ampliar sua funcionalidade, como para o cálculo de tensões residuais. Destaca-se aqui a complexidade teórica deste assunto, visto que tratar-se de um problema de dinâmica dos fluidos computacional com transferência de calor, mudança de fase e tensões térmicas.

O algoritmo iterativo principal implementado no buoyantPimpleFoam é conhecido como PIMPLE, tratando-se essencialmente de uma combinação entre os bem conhecidos algoritmos SIMPLE e PISO. Estes algoritmos fazem parte de uma família de métodos iterativos que tomam por base o algoritmo SIMPLE e incluem, dentre outros, o SIMPLER e o SIMPLEC. Mesmo sendo amplamente empregados, em geral esses algoritmos apresentam instabilidades e dependem da utilização de fatores de relaxação. No contexto do OpenFOAM o algoritmo PISO itera apenas na realização de “correções” na equação da pressão. O PIMPLE contempla uma iteração tipo SIMPLE, chamada de *outer loop*, realizada sob as equações de Navier-Stokes; todavia, dentro desta iteração ele também contempla a iteração na equação da pressão, chamada de *inner loop*, tal como o PISO. Assim, o PIMPLE consegue obter uma estabilidade mais robusta. Além disso, permite maior liberdade ao usuário (que pode escolher operar com um algoritmo PISO ou SIMPLE comum, por exemplo).

Outro ponto importante da análise foi a implementação da condição de contorno da convecção. Em muitos problemas reais de mudança de fase, há a troca de calor por convecção nos contornos da geometria do problema, modelada através de uma implementação específica da condição de contorno conhecida como condição de Robin ou de terceiro tipo. Assim, estudou-se sua formulação e de como ela pode ser implementada no OpenFOAM, seguida de *benchmarks* para sua validação. Para este estudo, foram utilizadas três implementações desta condição de contorno: com a biblioteca laplacianFoamRobinKappaConstant, presente no OpenFOAM; a codedMixed; e a Robin, implementada por um grupo de pesquisa da UFRJ.

Podemos compilar as atividades exercidas durante o período da bolsa na seguinte lista:

- Problema da cavidade (*lid-driven*): determinação de um campo de pressão e velocidade a partir da resolução de um sistema de equações diferenciais parciais utilizando o algoritmo iterativo SIMPLE, através do *solver* simpleFoam. O problema da cavidade é um problema clássico de *benchmark* utilizados em códigos CFD.
- Problema da viga em balanço, um caso recorrente de mecânica dos sólidos envolvendo tensão e deformação, resolvido com o *solver* solidDisplacementFoam.
- Problema do cilindro pressurizado, que possui geometria axissimétrica e solução analítica.
- Estudo de tensões térmicas com o caso de uma barra em dilatação.
- Estudo do problema advecção-difusão e o funcionamento do esquema de discretização *upwind*.
- Estudo do funcionamento do *solver* buoyantPimpleFoam acoplado ao módulo solidificationMeltingSource para resolução de problemas de mudança de fase.
- O funcionamento do algoritmo iterativo PIMPLE, uma combinação dos algoritmos SIMPLE e PISO.

- Realização de *benchmarks* com experimentos físicos e simulações numéricas dos experimentos de fusão do gálio e de derretimento do estanho, casos de mudança de fase consolidados na literatura.
- Análise da implementação da condição de contorno de convecção no OpenFOAM, incluindo estudo da formulação, da implementação e *benchmark* com o problema da parede plana com convecção, que possui solução analítica.
- Estudos iniciais sobre materiais viscoelásticos (modelos de Voigt-Kelvin e de Maxwell) e suas equações constitutivas com vista para uma possível inclusão em um modelo para cálculo de tensões térmicas em mudança de fase.
- Estudo sobre a implementação de um módulo que calcule as tensões térmicas em um problema de mudança de fase no OpenFOAM, baseados na literatura.

## Resultados e Discussões

Os resultados dos problemas da cavidade, da fusão do gálio, da viga em balanço e do problema de advecção-difusão atestaram com rigor a proficiência do OpenFOAM em resolver estes tipos de problema. Estes dados foram compilados em um artigo aceito no XL CILAMCE, contemplando também uma breve descrição do funcionamento do OpenFOAM e da teoria aplicada em cada análise. Mais tarde, foi realizado também o experimento de solidificação do estanho, com resultados que novamente atestam a capacidade do OpenFOAM de realizar problemas de mudança de fase.

Os experimentos realizados sobre o problema do cilindro pressurizado demonstraram aptidão do OpenFOAM em resolver problemas axissimétricos. Ademais, pôde-se compreender melhor o funcionamento do *solver* solidDisplacementFoam, pouco documentado, através deles. Ainda na parte de mecânica dos sólidos, o problema da barra em dilatação é um primeiro passo no entendimento de problemas mais complexos que envolvam tensões térmicas. Por fim, o estudo de materiais viscoelásticos e a modelagem do surgimento de tensões durante a mudança de fase seguem em andamento, avançando-se no entendimento desses materiais. Ainda há o que se trabalhar antes de finalizar um módulo que conte com o cálculo de tensões em mudança de fase, como foi proposto.

Além do CILAMCE, foram realizadas apresentações no Seminário de Modelagem Computacional do GET Computacional da UFJF; na Jornada de Iniciação Científica do LNCC; e no Seminário de Iniciação Científica, da UFJF.

## Conclusões

Foi realizado um estudo com vistas ao entendimento da teoria e dos procedimentos a se realizar na simulação de problemas de mudança de fase. Além disto, obteve-se domínio na ferramenta OpenFOAM, que foi validada nos experimentos realizados. Com isto, avança-se à próxima etapa de obter a mesma compreensão na modelagem e cálculo de tensões térmicas residuais que surgem durante o processo, com o intuito de implementar este modelo no OpenFOAM e validá-lo.

No geral, foi realizado um trabalho extenso sobre problemas de escoamento com mudança de fase; sobre mecânica dos sólidos e análise de tensões; e sobre o OpenFOAM. Isto rendeu um entendimento sobre estes assuntos, traduzindo-se não só na aceitação de um artigo no CILAMCE, mas também em apresentações no seminário do GET Computacional da UFJF, na Jornada de Iniciação Científica do LNCC e no Seminário de Iniciação Científica da UFJF. Pretende-se preparar uma compilação deste conhecimento em publicação que possa contribuir no avanço do desenvolvimento de modelos computacionais para a solução de problemas de mudança de fase e na validação e entendimento do OpenFOAM, que ainda é, infelizmente, mal

documentado. Softwares livres e gratuitos com grande reputação, como o OpenFOAM, mostram-se interessantes estrategicamente como alternativa a compra de soluções proprietárias e hardwares caros.

## Referências

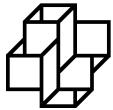
- Bergman, T. L., Incropera, F. P., DeWitt, D. P., & Lavine, A. S. (2008). *Fundamentos de Transferência de Calor e de Massa* (6 ed.). LTC.
- Brent, A. D., Voller, V. R., & Reid, K. T. (1988). Enthalpy-porosity technique for modeling convection-diffusion phase change: application to the melting of a pure metal. *Numerical Heat Transfer, Part A Applications*, 13, 297–318.
- Brooks, A. N., & Hughes, T. J. (1982). Streamline upwind/Petrov-Galerkin formulations for convection-dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer methods in applied mechanics and engineering*, 32, 199–259.
- Gau, C., & Viskanta, R. (1986). Melting and solidification of a pure metal on a vertical wall. *Journal of Heat Transfer*, 108, 174–181.
- Ghia, U. K., Ghia, K. N., & Shin, C. T. (1982). High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *Journal of computational physics*, 48, 387–411.
- Holzmann, T. (2016). Mathematics, numerics, derivations and OpenFOAM®. Loeben, Germany: Holzmann CFD.
- Jasak, H., & Weller, H. G. (2000). Application of the finite volume method and unstructured meshes to linear elasticity. *International journal for numerical methods in engineering*, 48, 267–287.
- Maric, T., Hopken, J., & Mooney, K. (2014). *The OpenFOAM Technology Primer*. Sourceflux.
- Odone, M. W. (2014). *Estudos numéricos na solução de problemas de mudança de fase*. Tese de Mestrado, Universidade Federal de Juiz de Fora - Programa de Pós-Graduação em Modelagem Computacional, Juiz de Fora, Brasil.
- Odone, M. W., Barra, L. P., Sutter, L. F., & Toledo, E. M. (2015). Uma Implementação do Método dos Volume Finitos Para Resolução de Problemas de Mudança de Fase. Rio de Janeiro: XXXVI CILAMCE.
- Odone, M. W., Neto, O. V., Toledo, E. M., Barra, L. P., & Soares, A. C. (2019). A Study of Phase Change Cast Modelling Using Open-FOAM. Natal: XL CILAMCE.
- Odone, M. W., Toledo, E. M., Barra, L. P., & Da Silva, C. E. (2013). Um Estudo Numérico Sobre o Método da Entalpia-Porosidade na Solução de Problemas de Mudança de Fase. Pirenópolis: XXXIV CILAMCE.
- Rady, M. A., & Mohanty, A. K. (1996). Natural convection during melting and solidification of pure metals in a cavity. *Numerical Heat Transfer, Part A Applications*, 29, 49–63.
- Soares, A. C. (2017). *Modelagem numérica da solidificação em moldes: aplicação a granadas de grande calibre*. Tese de Mestrado, Universidade Federal do Rio de Janeiro - Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia, Rio de Janeiro, Brasil.
- Teskeredžić, A., Demirdžić, I., & Muzaferija, S. (2002). Numerical method for heat transfer, fluid flow, and stress analysis in phase-change problems. *Numerical Heat Transfer: Part B: Fundamentals*, 42, 437–459.

Velasco, A. R. (7 de 2017). Boundary Conditions for Convective Heat Transfer in OpenFOAM. Fonte:  
[https://foamingtime2.files.wordpress.com/2017/07/convection-bc\\_1.pdf](https://foamingtime2.files.wordpress.com/2017/07/convection-bc_1.pdf)

Versteeg, W. (2007). *An introduction to computational fluid dynamics: the finite volume method*. Pearson education.

Voller, V. R., Brent, A. D., & Prakash, C. (1989). The modelling of heat, mass and solute transport in solidification systems. *International Journal of Heat and Mass Transfer*, 32, 1719–1731.

Wolff, F., & Viskanta, R. (1988). Solidification of a pure metal at a vertical wall in the presence of liquid superheat. *International journal of heat and mass transfer*, 31, 1735–1744.



Laboratório  
Nacional de  
Computação  
Científica

**Projeto de Pesquisa de Iniciação Científica**  
**Programa Institucional de Bolsas**  
**de Iniciação Científica - PIBIC**

# **Inversão de Dados Sísmicos**

**Bolsista:** Vinícius Theobaldo Jorge  
**Orientador:** Marcio Rentes Borges

Petrópolis-RJ  
27 de julho de 2020

## **Sumário**

1	Identificação . . . . .	1
2	Objetivos . . . . .	1
3	Introdução . . . . .	1
4	Metodologia . . . . .	2
4.1	Método de Diferenças Finitas . . . . .	2
5	Resultados e Discussão . . . . .	3
6	Conclusões . . . . .	5

## **1. IDENTIFICAÇÃO**

---

### **1 Identificação**

- **Título do projeto:** Inversão de Dados Sísmicos
- **Bolsista:** Vinícius Theobaldo Jorge
  - Aluno do curso de graduação de Geologia da UFRJ.
  - **E-mail:** *vinicius.tj@gmail.com*
- **Orientador:** Marcio Rentes Borges
  - Pesquisador Adjunto do Laboratório Nacional de Computação Científica (LNCC).
  - **E-mail:** *mrborges@lncc.br*
- **Bolsa:** PIBIC
- **Período do relatório:** julho de 2019 a julho de 2020

### **2 Objetivos**

O objetivo principal deste projeto é a inversão de dados sísmicos utilizando uma abordagem estocástica (Sen e Biswas, 2017). Entretanto, na presente fase, o objetivo foi o estudo de métodos numéricos para aproximação da equação da onda. Este relatório apresenta experimentos numéricos, tridimensionais, realizados com o código computacional desenvolvido para aproximar solução da equação da onda em meios elásticos heterogêneos.

### **3 Introdução**

A geofísica desempenha papel crucial na descoberta de novos reservatórios de petróleo e, ainda hoje, o principal uso de dados sísmicos é identificar a geometria dos refletores. Isso é possível devido ao fato de que as ondas sísmicas refletem nas interfaces entre materiais de diferentes propriedades acústicas (Frazer et al., 2008). Entretanto, recentemente, estudos sísmicos tem sido usados para a caracterização das rochas e monitoramento dos reservatórios (sísmica 4D), pela transformação de dados de reflexão em propriedades de rochas. Portanto, a inversão de dados sísmicos é uma ferramenta essencial na determinação das propriedades elásticas do subsolo.

## 4. METODOLOGIA

---

### 4 Metodologia

Seja  $I$  um intervalo de tempo e  $\Omega \in \mathbb{R}^3$  um domínio limitado contendo um material elástico heterogêneo pelo qual o som é propagado através de pequenas vibrações elásticas. O modelo matemático para descrever as deformações neste meio elástico é baseado na segunda lei de Newton e na lei constitutiva que relaciona a pressão  $p$  com o campo de deslocamentos  $\vec{u}$  (Lei de Hooke). Assumindo que as forças de corpo, as tensões de cisalhamento e o gradiente da massa específica ( $\rho$ ) podem ser negligenciados, temos que:

$$\frac{\partial^2}{\partial t^2} p(\vec{x}, t) = c^2(\vec{x}) \nabla^2 p(\vec{x}, t) + \varphi(\vec{x}, t), \quad (1)$$

onde,  $c(\vec{x}) = \sqrt{K(\vec{x})/\rho(\vec{x})}$  é a velocidade de propagação,  $K$  é o módulo elástico e  $\varphi$  o termo de fonte. Em conjunto com a Eq. (1) condições iniciais e de contorno devem ser fornecidas.

Foi implementado e testado um código computacional, escrito em FORTRAN 90, para aproximar a solução da Eq. (1) utilizando o método de diferenças finitas de segunda ordem, em um domínio tridimensional.

#### 4.1 Método de Diferenças Finitas

O método de diferenças finitas foi utilizado para aproximar a solução da Eq. (1). Os domínios, espacial e temporal, foram discretizados por um conjunto de pontos, nos quais a aproximação foi calculada. Mais especificamente, utilizamos uma malha espacial regular com espaçamentos  $\Delta x$ ,  $\Delta y$  e  $\Delta z$ . Para as derivadas temporal e espacial foram utilizadas aproximações centradas de segunda ordem. Definindo  $P_{i,j,k}^n$  como uma aproximação para  $p(x_i, y_j, z_k, t^n)$ , onde os índices  $i, j, k$  se referem aos pontos da malha nas direções  $x, y, z$ , respectivamente, enquanto índice  $n$  representa o número do passo de tempo da simulação. Portanto, podemos escrever a seguinte aproximação:

## 5. RESULTADOS E DISCUSSÃO

---

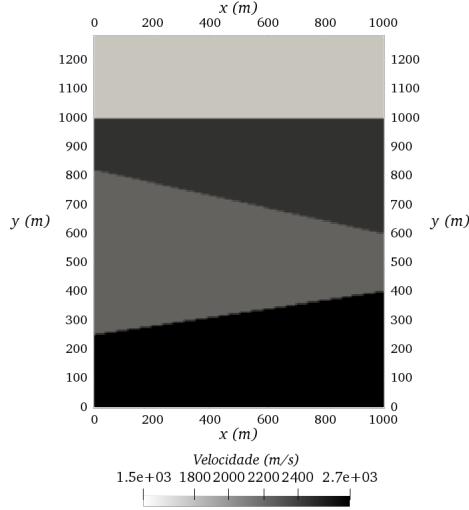


Figura 1: Velocidades das camadas do domínio  $\Omega$ .

$$\begin{aligned}
 P_{i,j,k}^{n+1} = & C \left( \frac{\Delta t}{\Delta x} \right)^2 [P_{i+1,j,k}^n - 2P_{i,j,k}^n + P_{i-1,j,k}^n] + \\
 & C \left( \frac{\Delta t}{\Delta y} \right)^2 [P_{i,j+1,k}^n - 2P_{i,j,k}^n + P_{i,j-1,k}^n] + \\
 & C \left( \frac{\Delta t}{\Delta z} \right)^2 [P_{i,j,k+1}^n - 2P_{i,j,k}^n + P_{i,j,k-1}^n] + \\
 & 2P_{i,j,k}^n - P_{i,j,k}^{n-1} - \Delta t^2 \delta(i, i_\varphi) \delta(j, j_\varphi) \delta(k, k_\varphi) \varphi^n
 \end{aligned} \tag{2}$$

onde  $C = c_{i,j,k}^2 = \left( \frac{K(x_i, y_j, z_k)}{\rho(x_i, y_j, z_k)} \right)^2$ . Os pontos  $i_\varphi, j_\varphi, k_\varphi$  indicam a posição da fonte nos nós da malha, portanto,  $\delta(\alpha, \alpha_\varphi) = 1$  caso  $\alpha = \alpha_\varphi$  e  $\delta(\alpha, \alpha_\varphi) = 0$ , caso contrário, onde  $\alpha = i, j, k$ .

## 5 Resultados e Discussão

Consideramos um domínio  $\Omega = [1000m \times 1280m \times 1000m]$ , discretizado em uma malha de  $251 \times 321 \times 251$  pontos. O domínio  $\Omega$  apresenta quatro camadas de materiais com diferentes velocidades (Figura 1). Uma fonte sísmica, descrita matematicamente por

$$\varphi(\vec{x}, t) = [1 - 2\pi (\pi f_c t_d)^2] e^{-\pi (\pi f_c t_d)^2}, \tag{3}$$

## 5. RESULTADOS E DISCUSSÃO

foi localizada na posição  $(x, y, z) = (500m, 1280m, 500m)$ . Aqui  $f_c = f_r/3\sqrt{\pi}$  é a frequência de corte,  $t_d = t - 2\sqrt{\pi}/f_r$  e  $f_r$  a frequência da onda (neste exemplo,  $f_r = 50Hz$ ). Nas fronteiras do domínio foram consideradas condições de contorno absorventes.

A Figura 2 mostra a propagação da onda sísmica em vários instantes de tempo. Podemos observar que a onda, ao atingir a camada com propriedade diferente, parte dela é refletida e parte é transmitida com velocidade diferente, conforme esperado.

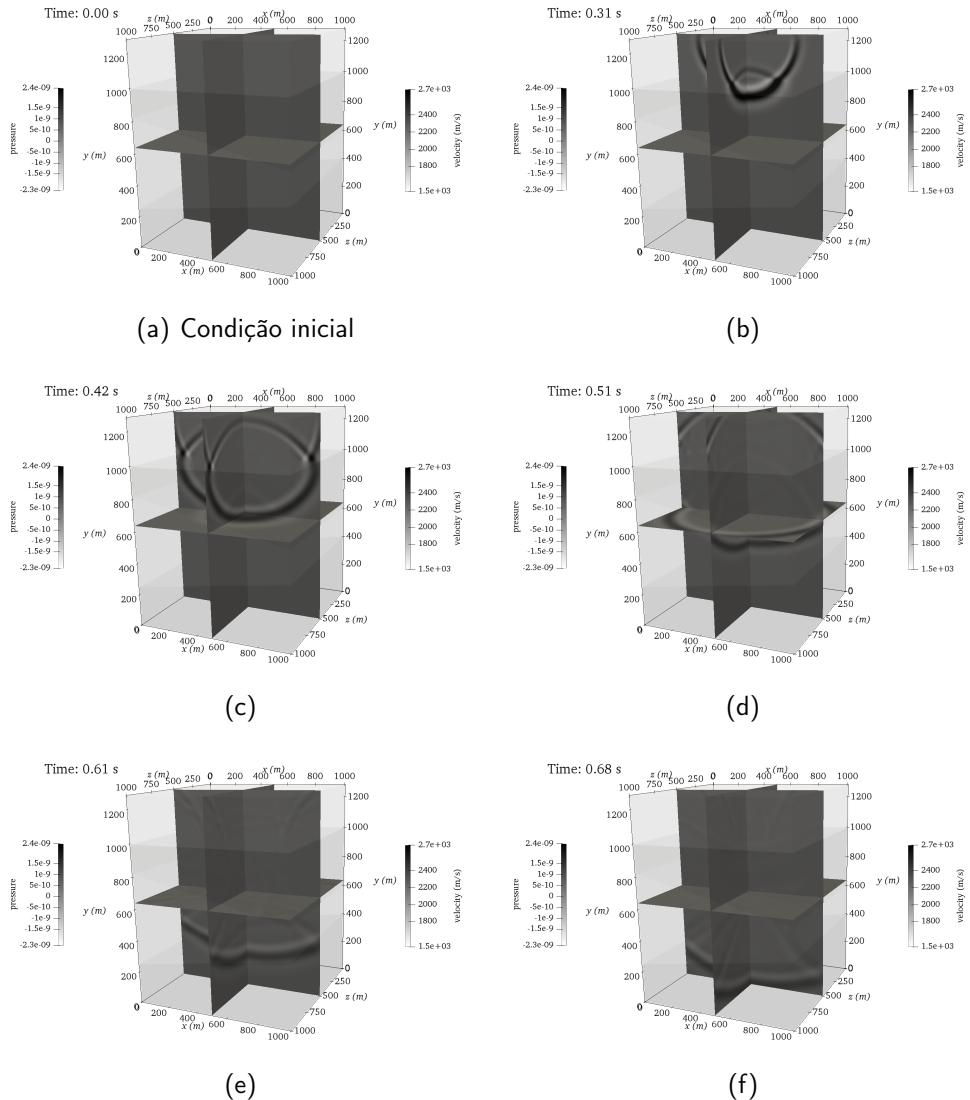


Figura 2: Pressão em tempos destintos.

## 6 Conclusões

O método de diferenças finitas proporcionou uma boa aproximação da solução da Eq. (1), capturando o comportamento físico esperado para ambos problemas estudados. Com isso obtivemos um programa que está qualificado para usar dados reais de estudos relacionados a toda cadeia produtiva do petróleo.

## Referências

- B. Frazer, A. Bruun, K. B. Rasmussen, J. C. Alfaro, A. Cooke, D. Cooke, D. Salter, R. Godfrey, D. Lowden, S. McHugo, H. Özdemir, S. Pickering, F. G. Anderson Pineda, J. Herwanger, S. Volterrani, A. Murneddu, A. Rasmussen, e R. Roberts. Seismic inversion: Reading between the lines. *Oilfield Rev*, 20(1):42–63, 2008.
- M. K. Sen e R. Biswas. Transdimensional seismic inversion using the reversible jump hamiltonian monte carlo algorithm. *Geophysics*, 82(3):119–134, 2017.

---

**Laboratório Nacional de Computação Científica  
Coordenação de Ciência da Computação (CCC)**

**Relatório de atividades**

**Programa Institucional de Bolsas de Iniciação Científica  
PIBIC**

**Uso de Aprendizado de Máquina para Melhorias no  
Desempenho e no Consumo de Energia em Sistemas HPC**

**Vitor Carlos Soares de Sá**

**Orientador - Prof. Bruno Schulze**

**Período Agosto/2019 a Julho/2020**

---

**Julho 2020**

---

## Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
<b>2</b>	<b>Objetivos</b>	<b>1</b>
<b>3</b>	<b>Metodologia</b>	<b>2</b>
<b>4</b>	<b>Resultados e Discussão</b>	<b>3</b>
4.1	Publicações . . . . .	5
<b>5</b>	<b>Conclusões</b>	<b>6</b>

## Dados Referentes à Bolsa

1. **Instituição:** Laboratório Nacional de Computação Científica (LNCC)
2. **Bolsista:** Vitor Carlos Soares de Sá
3. **Orientador:** Bruno Schulze (CCC-LNCC)
4. **Co-Orientadora:** Mariza Ferro
5. **Período de Vigência da Bolsa:** Agosto de 2019 até Julho de 2020
6. **Modalidade:** Bolsa de Iniciação Científica - PIBIC
7. **Projeto Relacionado:** ComCiDis (Computação Científica Distribuída)

## 1 Introdução

A Computação Científica é uma ferramenta de grande importância para pesquisas em diversas áreas da ciência. Entretanto, algumas áreas da ciência ainda necessitam de maior poder computacional para execução de simulações cada vez mais precisas. Por outro lado, com o aumento do poder computacional está o aumento do consumo de energia elétrica.

Desse modo, compreender e caracterizar o desempenho dessas aplicações quando executadas em diferentes arquiteturas torna-se tarefa essencial para o uso eficiente dos recursos computacionais. Contudo, esta é uma tarefa complexa, pois entre os maiores desafios estão o monitoramento e a coleta precisa dos dados sobre as simulações e relacioná-los com os recursos de *hardware* e energia, além de identificar os aspectos que limitam o desempenho das aplicações. Portanto, neste trabalho será explorado o uso da metodologia *Roofline* para HPC, com o objetivo de obter conhecimentos sobre os requisitos computacionais das aplicações e os fatores que limitam seu desempenho. Além disso, estes conhecimentos serão a base para a seleção de atributos relevantes nas tarefas de Aprendizado de Máquina que serão desenvolvidas para entender e caracterizar o desempenho e consumo de energia das aplicações.

## 2 Objetivos

O objetivo geral deste projeto é o estudo do comportamento de aplicações científicas em relação a arquitetura de *hardware*, ciente de seus requisitos computacionais, visando melhorias no equilíbrio entre o desempenho e a economia de energia. Os objetivos específicos são:

- Monitorar de forma precisa e eficiente a execução das aplicações.
- Compreender as aplicações científicas quanto aos requisitos de hardware mais importantes para resolução dos problemas modelados utilizando diferentes métodos matemáticos.
- Compreender as arquiteturas computacionais quanto a capacidade computacional e a eficiência energética.
- Identificar os fatores que limitam o desempenho das aplicações, utilizando medidas de Intensidade Aritmética (IA) e compreender como estes fatores se relacionam com o consumo de energia.
- Modelar e predizer o desempenho e o consumo de energia das aplicações em arquiteturas heterogêneas.

### 3 Metodologia

Com base nos objetivos apresentados na Seção 2 foram propostas as seguintes metas para a realização do projeto:

- Meta 1: Levantamento bibliográfico sobre o estado da arte.

Revisão bibliográfica sobre os conceitos de HPC, computação científica, computação paralela, classes Motifs, sistemas operacionais distribuídos e aplicações científicas [Ferro 2015]. Abordagens de monitoramento de desempenho e energia voltadas para a computação científica, ambiente de análise e visualização de dados, incluindo metodologias de coleta de dados empregadas em ferramentas de monitoramento, dentre elas: amostragem, cronometragem, rastreamento e contagem. Análise de resultados empregados em ferramentas de monitoramento: abordagem *online* e abordagem *offline* [Silva et al. 2017].

- Meta 2: Definição das aplicações científicas e *benchmarks* representativos a serem utilizados para o estudo.

Foi realizado o estudo do conjunto de benchmarks do *NAS*<sup>1</sup>, os quais representam as aplicações de HPC, além de sua instalação, compilação, conjunto de aplicações e tamanhos de experimentos. O *NAS* tem aplicações que representam as classes Motifs utilizadas como referencial neste trabalho. As aplicações BT (*Block Tri-diagonal*) e LU (*Lower-Upper Gauss-Seidel*), representam aplicações de Álgebra Linear Densa (DLA); CG (*Conjugate Gradient*) para Álgebra Linear Esparsa (SLA); MG (*Multi-Grid*) e SP (*Penta-diagonal scalar*) representam aplicações de Grades Estruturadas (SG); UA (*Unstructured Adaptive mesh*) e DC (*Data Cube*), Grades Não-Estruturadas (UG).

- Meta 3: Monitoramento da execução das aplicações, coleta de contadores de *hardware*, desempenho e consumo de energia e identificação dos parâmetros mais relevantes.

Monitorar a execução de aplicações em diferentes arquiteturas pode não ser uma tarefa trivial, pois os contadores de *hardware* podem variar entre os modelos de processadores e as diferentes arquiteturas. Ainda, as ferramentas disponíveis para o monitoramento não coletam todos os parâmetros relacionados ao desempenho ou ao consumo de energia. Desta forma, foi feita uma análise técnica de um conjunto de ferramentas de monitoramento de desempenho e energia, verificando suas vantagens, limitações e como essas ferramentas apresentam os dados para análise, avaliando a facilidade de instalação, *overhead*, utilização e interpretação destes resultados. Dentre as ferramentas estudadas e utilizadas durante o projeto, temos: *perf*<sup>2</sup>, por ser uma ferramenta de análise de desempenho que permite a coleta de informações dos registradores de *hardware*. A ferramenta SMCis [Silva et al. 2018], desenvolvida pelo grupo ComCiDis do LNCC, que coleta os parâmetros relacionados ao consumo de *hardware* e as informações de potência e temperatura utilizando o IPMITool<sup>3</sup>. A ferramenta *Likwid* [Eitzinger et al. 2011], utilizando a métrica perfctr, que consiste em obter os dados diretamente dos contadores de *hardware*. Os principais parâmetros que foram utilizados para obter informações relevantes para o desenvolvimento da pesquisa utilizando a ferramenta *Likwid* foram: FLOPS\_DP (informações de desempenho de ponto flutuante), ENERGY (informações de consumo de energia), MEM (informações de uso de memória RAM, largura de banda), L2 e L3 (informações de uso de memória cache, respectivamente), DATA (oferece as informações necessárias para mensurar o consumo de memória cache em nível L1). Dentre outras ferramentas como *Intel VTUNE*, *nuprof* e *Intel Advisor*.

<sup>1</sup><https://www.nas.nasa.gov/publications/npb.html>

<sup>2</sup>[https://perf.wiki.kernel.org/index.php/Main\\_Page](https://perf.wiki.kernel.org/index.php/Main_Page)

<sup>3</sup><https://linux.die.net/man/1/ipmitool>

- Metas 4 e 5: Compreensão das aplicações científicas e arquiteturas quanto à exigência dos recursos computacionais e eficiência energética. Identificação dos fatores que limitam o desempenho das aplicações utilizando medidas de IA.

Para a realização das metas 4 e 5 foi utilizado uma metodologia chamada *Roofline*, que relaciona desempenho de ponto flutuante a intensidade aritmética para extração de conhecimentos sobre os requisitos computacionais mais relevantes sobre as aplicações, a partir da análise do desempenho prático atingido pela aplicação ao teto de desempenho e mapeando e categorizando o *hardware* em relação ao consumo da aplicação. Esta metodologia foi utilizada também para obtenção de conhecimentos sobre a relação desses requisitos com o desempenho, de forma a caracterizar as arquiteturas quanto a capacidade computacional. Parâmetros de energia foram obtidos com a ferramenta *Likwid*. Estudo da metodologia *Roofline*, entendimento do modelo, sua metodologia de coleta de dados, a compreensão e a interpretação de seus resultados com base em [Williams et al. 2009, Ilic et al. 2014, Yang et al. 2018, Lo et al. 2014]. Além disso, foi realizado o estudo da ferramenta *Intel Advisor*<sup>4</sup>, a qual atualmente é uma das principais ferramentas que atuam por meio da metodologia *Roofline*, além de analisar o desempenho da aplicação em relação ao hardware, dá ênfase para análise de eficiência de vetorização, *multithreading* e principalmente no auxílio em categorizar uma aplicação em relação ao seu requisito computacional.

## 4 Resultados e Discussão

O *Roofline* se mostra uma metodologia útil para os objetivos propostos, oferecendo um modelo visual que relaciona o desempenho do processador ao tráfego de memória, apontando os gargalos da aplicação e possíveis formas de otimização. O seu diferencial é unir todas essas informações em um único ambiente, diferindo dos benchmarks tradicionais, os quais analisam separadamente o desempenho da aplicação e a capacidade computacional do *hardware*. Reunir todas essas informações oferecem *insights* pertinentes sobre o funcionamento do *hardware* e do *software*. Esse modelo vincula a intensidade aritmética (operações por byte de tráfego), a largura de banda e o desempenho de ponto flutuante (operações por segundo) juntos em um gráfico dimensional, onde o eixo x representa a Intensidade Aritmética (IA), o eixo y representa o desempenho de ponto flutuante e a largura de banda representa o teto de memória. Dentro do gráfico, há o pico teórico de desempenho de ponto flutuante e o pico de largura de banda, que podem ser obtidos através de especificações do hardware ou benchmarks. De acordo com a necessidade do usuário são inseridas técnicas de otimizações para a aplicação tais como, reconhecimento dos níveis de cache, vetorização, paralelismo a nível de *thread* e instrução, entre outras. Para avaliar o desempenho da aplicação é necessário obter os parâmetros práticos, e relacioná-los aos parâmetros teóricos. Na Figura 1 é possível observar os parâmetros, descrição, fórmulas e métodos de coleta teóricos e práticos. Esse levantamento de parâmetros e fórmulas não foram triviais de serem obtidos, já que apresentam muitas divergências na literatura. Assim, esse resumo da metodologia e como utilizá-la corretamente é resultado de todo processo de pesquisa bibliográfica e experimental.

Por meio deste estudo foi possível desenvolver uma ferramenta baseada na metodologia *Roofline* (Figura 4), implementado a nível de memória DRAM e fundamentado em avaliação geral da aplicação em relação ao ambiente. Utilizamos a ferramenta GNUPLOT para gerar o gráfico *Roofline*, capaz de plotar os gráficos de funções matemáticas em duas ou três dimensões. Para fazer os tetos, foram utilizadas informações do próprio hardware, baseado nas fórmulas descritas na Figura 1. Além disso, foram adicionadas informações de pico de

<sup>4</sup><https://software.intel.com/en-us/advisor>

	Utiliza	Dados	Descrição	Equação	Método Teórico	Método Prático
Operações de Ponto Flutuante (OPF)	Número de Soquetes (NS)	Quantidade	Operações de Ponto Flutuante	$\text{Peak} = \text{NS} * \text{FP} * \text{NN} * \text{NSIMD} * \text{NPD}$	Obtenção desses dados em sudo lshw e/ou no site oficial	perf stat -M GFLOPs
	Freqüência do Processador (FP)	Ghz				
	Número de Núcleos (NN)	Quantidade				
	Número de Elementos de Precisão Única SIMD (NSIMD)	Quantidade				
Largura de Banda (LB)	Número de Portas Dedicadas (Soma, Mult.) (NPD)	Quantidade	Determina a medida da capacidade de transmissão	$\text{Peak} = \text{NS} * \text{FM} * \text{BC} * \text{NCM}$	Obtenção desses dados em sudo lshw e/ou no site oficial	lkhwd-perfctr -O -g (MEM, L2, L3)
	Número de Soquetes (NS)	Quantidade				
	Freqüência da Memória (FM)	MHz				
Intensidade Aritmética (IA)	Byte por Canal (BC)	Byte	CPU <-> MEM (L1, L2, L3 e DRAM)	OPF / LB	Citado Anteriormente	perf stat -M GFLOPs
	Número de Canais de memória (NCM)	Quantidade			Citado Anteriormente	lkhwd-perfctr -O -g (MEM, L2, L3)
Operações de Ponto Flutuante (OPF)	Operações de Ponto Flutuante (OPF)	GFLOPs	CPU <-> MEM (L1, L2, L3 e DRAM)	OPF / LB	Citado Anteriormente	perf stat -M GFLOPs
	Largura de Banda (LB)	Byte				

Figura 1: Parâmetros, descrição, fórmulas e métodos de coleta teóricos e práticos do *Roofline*

desempenho utilizando diferentes quantidades de núcleos. Para obter informações sobre o desempenho da aplicação foram utilizadas ferramentas distintas, resumidas em 1.

Com a implementação do modelo Roofline utilizando o GNUPLOT, seria necessário gerar um modelo matemático novo e um monitoramento manual a cada execução diferente da aplicação a ser avaliada, tornando-se uma tarefa inviável. Para contornar esse problema foi desenvolvido um programa escrito em Bash. O programa consiste em monitorar e reunir as informações da aplicação, relacioná-las às informações previamente descritas sobre o *hardware*, gerar o modelo matemático necessário para a geração do gráfico *Roofline* baseado nas informações da aplicação e do *hardware*, e traduzi-las para a sintaxe do GNUPLOT.

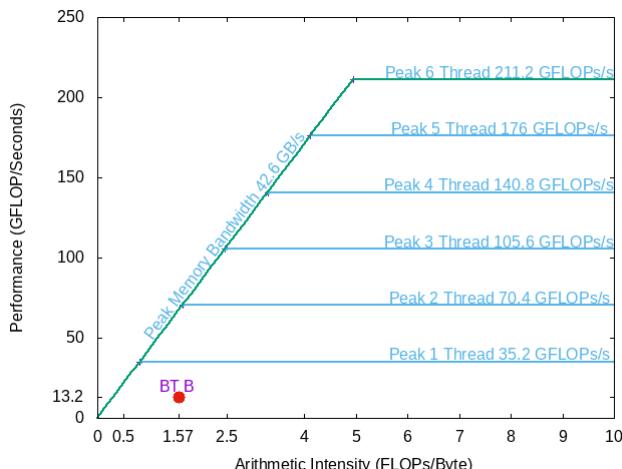
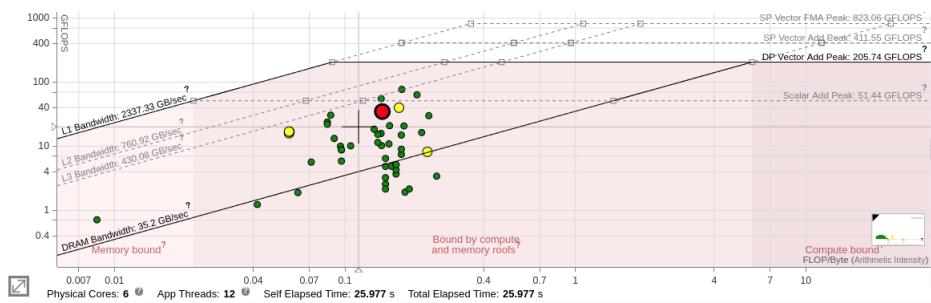


Figura 2: Implementando um Modelo *Roofline*

Implementar técnicas de otimização em um modelo baseado na metodologia *Roofline* é bastante complexo, por isso foi realizado um estudo sobre a ferramenta *Intel Advisor*. O *Advisor* oferece técnicas para análise de eficiência de vetorização, *multithreading* e principalmente o auxílio em categorizar uma aplicação em relação ao seu requisito computacional. Além disso, o *Advisor* oferece uma visão de ambiente em vários níveis de memória *cache*, sendo um aprimoramento do conceito original, que é baseado apenas em memória *DRAM*.

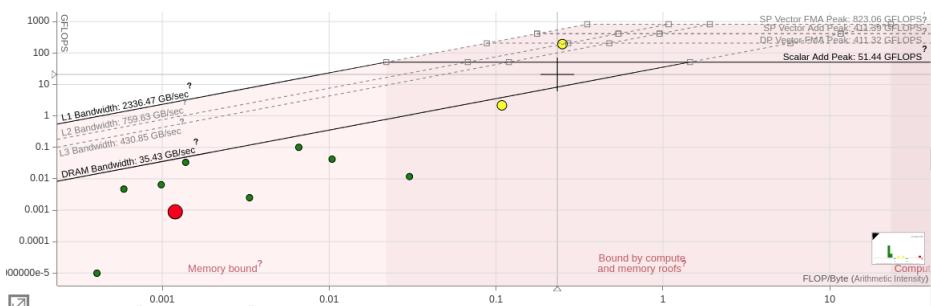
No gráfico do *Advisor*, cada loop/função da aplicação representa um círculo (sendo 3 tipos diferentes de círculos). Isso auxilia na análise do custo computacional de cada um, destacando então os círculos vermelhos como custosos, amarelo como custo moderado e verde como baixo custo. Podemos também avaliar o quanto otimizada se encontra a aplicação, relacionando a distância do ponto ao teto de desempenho (quanto mais longe, mais espaço para ganho de desempenho). Essas duas informações podem ser usadas para tomar a decisão de quais funções apresentam uma capacidade de otimização maior. O teto acima do ponto representa o limitador de desempenho daquele ponto em específico, onde cada teto representa seu próprio limite de *hardware*, caso não seja feita a otimização representada pelo próximo teto.

Figura 3: Resultado da aplicação BT - *NAS*.



(a) BT.

Figura 4: Resultado do algoritmo de Inteligência Artificial - CNN



(a) CNN.

Através desses gráficos é possível avaliar o comportamento de duas aplicações. A Figura 3 é uma aplicação DLA do *NAS*. Observa-se que é uma aplicação equilibrada entre processamento e memória, pois se encontra no centro do gráfico. Um dos maiores limitadores dessa aplicação seria a falta ou a implementação irregular de vetorização. Além disso, a memória L3 está limitando a capacidade da aplicação, e assim, uma arquitetura com uma memória cache L3 grande pode otimizar o desempenho dessa aplicação. Outra alternativa é buscar uma maior utilização da IA, o que pode chegar o gráfico para a direita, ampliando o teto. Ou seja, o desempenho atingível da aplicação. Essa é uma aplicação que não apresenta muito espaço para otimizações, já que seus principais pontos se encontram próximos ao teto. Na Figura 4 o gráfico é do algoritmo de rede neural convolucional (CNN), que se mostra intensivo em memória com alguns requisitos de processamento. Seu maior limitador é a memória *DRAM*. Otimizações de processamento para esse algoritmo se torna irrelevante, pois seria necessário uma IA maior para chegar aos requisitos de processamento, assim como a sua execução em uma máquina que ofereça alta capacidade de processamento pode ser um desperdício computacional. Porém, a execução desse algoritmo em uma máquina que ofereça uma memória alta pode influenciar positivamente na sua execução.

#### 4.1 Publicações

Vinícius Klôh, Matheus Gritz, Vitor Sá, Bruno Schulze, Mariza Ferro. (2020). Characterization of the Scientific Applications Requirements: An Hybrid Approach Using Machine Learning and Roofline. <http://doi.org/10.5281/zenodo.3961639>

Neste momento encontra-se em preparação um artigo a ser submetido ao Workshop de Sistemas Computacionais de Alto Desempenho (WSCAD-WIC 2020).

## 5 Conclusões

Neste trabalho é apresentada a metodologia de estudo para buscar a compreensão dos sistemas computacionais e das aplicações científicas, incluindo seus requisitos computacionais, visando otimizações de desempenho. A metodologia *Roofline* se provou útil para a proposta do projeto, pois é um modelo que fornece informações sobre o sistema e o desempenho da aplicação. Além disso, é uma ferramenta valiosa para auxiliar no desenvolvimento e otimização de aplicações em arquiteturas futuras. Os resultados apresentados oferecem informações relevantes de quão otimizado se encontra os algoritmos executados e o que deve ser feito para a otimização do mesmo, em um modelo que integra o ambiente e a execução em um mesmo modelo gráfico.

Em trabalhos futuros, será implementado um modelo *Roofline* para avaliação de consumo de energia, proposto por [Ghane et al. 2018]. Além disso, será feito o desenvolvimento e validação de um modelo preditivo para o desempenho e o consumo de energia utilizando aprendizado de máquina.

## Referências

- [Eitzinger et al. 2011] Eitzinger, J., Hager, G., Wellein, G., and Meier, M. (2011). Poster: Likwid: lightweight performance tools. pages 29–30. 2
- [Ferro 2015] Ferro, M. (2015). *Avaliao de Sistemas de Computao Massivamente Paralela e Distribuda: Uma metodologia voltada aos requisitos das aplicaes cientficas*. Tese de doutorado, Laboratrio Nacional de Computao Cientfica, Petrpolis - RJ. 2
- [Ghane et al. 2018] Ghane, M., Larkin, J., Shi, L., Chandrasekaran, S., and Cheung, M. (2018). Power and energy-efficiency roofline model for gpus. 6
- [Ilic et al. 2014] Ilic, A., Pratas, F., and Sousa, L. (2014). Cache-aware roofline model: Upgrading the loft. *IEEE Computer Architecture Letters*, 13(1):21–24. 3
- [Lo et al. 2014] Lo, Y., Williams, S., Straalen, B., Ligocki, T., Cordery, M., Wright, N., Hall, M., and Oliker, L. (2014). Roofline model toolkit: A practical tool for architectural and program analysis. pages 129–148. 3
- [Silva et al. 2017] Silva, G., Ferro, M., Oliveira, V., Klöh, V., Yokoyama, A., and Schulze, B. (2017). Estudo de abordagens de monitoramento de desempenho e energia para computação científica. 2
- [Silva et al. 2018] Silva, G., Klöh, V., Yokoyama, A., Ferro, M., and Schulze, B. (2018). Smcis: um sistema para o monitoramento de aplicações científicas em ambientes hpc. pages 277–288. <http://www2.sbc.org.br/wscad/current/anais/anais-wscad-2018.pdf>. 2
- [Williams et al. 2009] Williams, S., Waterman, A., and Patterson, D. (2009). Roofline: an insightful visual performance model for multicore architectures. *Commun. ACM*, 52(4):65–76. 3
- [Yang et al. 2018] Yang, C., Gayatri, R., Kurth, T., Basu, P., Ronaghi, Z., Adetokunbo, A., Friesen, B., Cook, B., Doerfler, D., Oliker, L., Deslippe, J., and Williams, S. (2018). An empirical roofline methodology for quantitatively assessing performance portability. pages 14–23. 3

---

**Laboratório Nacional de Computação Científica  
Coordenação de Ciência da Computação (CCC)**

**Relatório de atividades**

**Programa Institucional de Bolsas de Iniciação Científica  
PIBIC**

**Análise de Desempenho e Características de Paralelização dos  
Algoritmos de Aprendizado de Máquina**

**Vitor Gonçalves Vieira**

**Orientador - Prof. Bruno Schulze**

**Período Abril/2020 a Julho/2020**

---

**Abril 2020**

## Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
<b>2</b>	<b>Objetivos</b>	<b>1</b>
<b>3</b>	<b>Metodologia</b>	<b>2</b>
<b>4</b>	<b>Resultados e Discussão</b>	<b>2</b>
<b>5</b>	<b>Conclusões e trabalhos Futuros</b>	<b>4</b>

## Dados Referentes à Bolsa

- **Instituição:** Laboratório Nacional de Computação Científica
- **Bolsista:** Vitor Gonçalves Vieira
- **Coordenador/Orientador:** Bruno Schulze (CCC-LNCC)
- **Coorientadora:** Mariza Ferro (LNCC)
- **Modalidade:** Bolsa de Iniciação Científica
- **Período de Vigência da Bolsa:** Abril de 2020 até Julho de 2020
- **Projetos Relacionados:** ComCiDis (Computação Científica Distribuída)

## 1 Introdução

A inteligência artificial (IA) é uma das forças mais transformadoras do nosso tempo e que vem provocando alterações na sociedade e constituindo uma excelente oportunidade para aumentar a prosperidade e o crescimento dos países. Na última década foram realizados grandes progressos em técnicas de IA, e da sua sub-área de Aprendizado de Máquina (AM), devido à disponibilidade de enormes quantidades de dados digitais e de arquiteturas de computação de alto desempenho (HPC<sup>1</sup>). Com a popularização no uso da IA por diversas áreas da ciência e da indústria, têm ocorrido também um crescimento acentuado na busca e investimento por hardwares que possam alcançar melhorias significativas de desempenho e economia de energia para execução dessas aplicações (por exemplo, GPUs, FPGAs e ASICs).

Apesar do uso de arquiteturas do tipo GPGPU para algoritmos de RNA (incluindo Deep Learning), já estar bastante estabelecido, poucos estudos comprovam se as GPUs, FPGAs e ASICs são realmente mais eficientes em relação as CPUs para qualquer algoritmo de IA. Mesmo com a relevância desses algoritmos de AM, pouco se sabe a respeito dos seus requisitos computacionais em diferentes arquiteturas computacionais. Ou seja, as diferentes tarefas de AM envolvem diferentes algoritmos, os quais podem ter requisitos computacionais bastante diversos. Devido a alta demanda de HPC faz-se necessário compreender se os algoritmos de AM possuem um desempenho satisfatório frente as diferentes técnicas de paralelização existentes. Assim, motivados pela relevância do tema e dos aspectos desafiadores a serem explorados na próxima seção são apresentados os objetivos deste projeto de pesquisa.

## 2 Objetivos

O objetivo geral deste projeto de pesquisa é a identificação dos requisitos de software e hardware das diferentes aplicações de IA, mais especificamente aplicações de AM. Além disso, avaliar para quais algoritmos de AM são possíveis aplicar técnicas de paralelização que melhorem o seu desempenho. Assim, os objetivos específicos são:

- a. Estudar as diferentes classes de algoritmos de AM e seus respectivos algoritmos;
- b. Monitorar de forma precisa e eficiente a execução dos algoritmos de AM durante as fases de treinamento. Serão avaliados aspectos de desempenho e consumo de energia;

<sup>1</sup>HPC - Do termo em inglês High Performance Computing

- c. Identificar os principais requisitos de hardware das diferentes técnicas de AM (uso de CPU, E/S, memória e comunicação);
- d. Identificar os *hotspots* de algoritmos de AM e como essas partes do código influenciam no desempenho e consumo de energia;
- e. Identificar as principais técnicas de paralelização de algoritmos existentes para serem aplicadas nos algoritmos de AM e quais classes são viáveis de serem paralelizados.

### 3 Metodologia

Para cumprir os objetivos acima descritos foi adotada uma metodologia para o desenvolvimento da pesquisa baseada em metas. Durante esses três meses de trabalho foi possível desenvolver atividades relacionadas às duas primeiras metas.

- **Meta 1:** Levantamento bibliográfico sobre AM e avaliação de desempenho e energia.

Revisão bibliográfica sobre algoritmos de AM, entendendo conceitos de algoritmos supervisionados, não supervisionados, o objetivo de cada um e entendendo suas diferenças [Gron 2017]. Este estudo foi realizado utilizando como base a linguagem de programação *Python*<sup>2</sup> em sua versão 3.8, estudando suas estruturas, paradigmas e funções, e suas bibliotecas como o *Scikit-learn* [Pedregosa et al. 2011] e *Pandas* [Wes McKinney 2010] e visualizando com *Matplotlib* [Hunter 2007].

Além disso, foi dado início às atividades para o monitoramento de desempenho e energia. Para isso está sendo utilizada a plataforma *Linux* para realizar essa medição de forma mais precisa.

- **Meta 2:** Caracterizar diferentes modelos de algoritmos de AM.

Estão sendo implementados os modelos de AM, iniciando pela regressão linear, por ser o modelo mais básico. Com o desenvolvimento da Meta 1, foi possível adquirir o conhecimento sobre como fornecer dados “bons” para os algoritmos (pré-processamento de dados), melhorar seus resultados, ajustar parâmetros, desde a regressão linear, até modelos mais complexos como árvore regressora, que está sendo o foco atual da pesquisa.

### 4 Resultados e Discussão

Pode-se dizer que *Machine Learning* ou Aprendizado de Máquina é a arte de programar um computador, para que ele possa aprender com os dados [Gron 2017]. O primeiro modelo de AM implementado neste trabalho foi a regressão linear. Porém, com o estudo e implementação foi possível perceber que se trata de um modelo bastante simples. Pela simplicidade do modelo linear é difícil obter bons resultados para problemas complexos, pois ele não funciona bem com muitos atributos e poucos dados, além da necessidade dos dados serem contínuos para fazer uma previsão dentro do esperado.

Usando uma base de dados artificial sobre a quantidade de copos de suco produzidos a partir de uma quantidade de limões, foi traçada a reta que melhor se ajusta aos dados fornecidos. É possível perceber que o modelo não se adapta perfeitamente aos dados, causando um grande erro (Figura 1).

<sup>2</sup>[www.python.org/](http://www.python.org/)

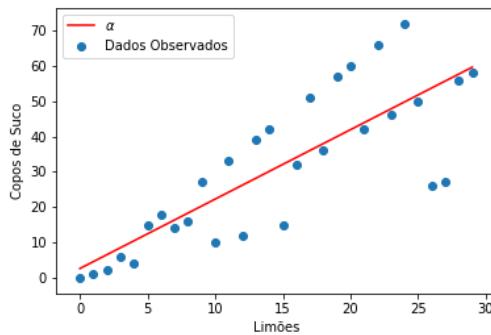


Figura 1: Dados e a reta do modelo linear que melhor se adaptou aos dados fornecidos.

Após a regressão linear foi estudado o modelo de Árvore de Decisão (AD). Os algoritmos de AD são um dos modelos mais práticos e mais usados em inferência indutiva (Figura 2). Este método representa funções como árvores de decisão. Estas árvores são treinadas de acordo com um conjunto de treino (exemplos previamente classificados - aprendizado supervisionado) e posteriormente, outros exemplos de teste são classificados de acordo com essa mesma árvore. Esses algoritmos podem ser divididos entre árvores classificadoras e regressoras.

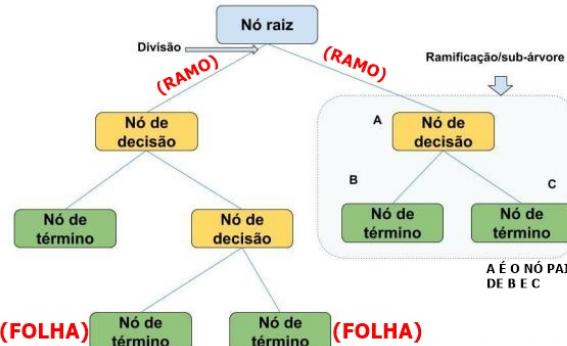


Figura 2: Ilustração do funcionamento de uma árvore

Os algoritmos de árvore mais conhecidos e utilizados devido a sua eficiência são o ID3 [Quinlan 1986], C4.5 [Quinlan 1993] e CART [Moore II 1987], sendo este último o foco deste trabalho, pois além de ser integrado ao *Sckit-Learning*, ele suporta AD regressora. As árvores classificadoras utilizam Ganhos de informação baseado na entropia, ou Gini, para dividir seus ramos, já a regressora utiliza desvio padrão nas suas folhas para predizer resultados numéricos.

Analizando uma árvore classificadora utilizando o conjunto de dados “PlayTennis”<sup>3</sup>, inicialmente foi necessário transformar os valores dos atributos para numéricos pois o CART não suporta atributos nominais, depois foi gerada uma ilustração das decisões da árvore utilizando ferramenta *Graphviz*<sup>4</sup>, como mostrado na Figura 3.

Ao executar alguns testes utilizando a árvore e gerar a sua representação, foi possível perceber que se tratava de um modelo muito poderoso e rápido, mesmo com dados faltantes. Como ela é muito rápida para poucos exemplos, foi utilizada o modelo regressor com uma base de dados sintética (sem objetivo de obter conhecimento ou precisão do modelo) com 100 mil exemplos, para ser melhor medido o consumo de energia. A medição do consumo é

<sup>3</sup><https://www.kaggle.com/fredericobreno/play-tennis>

<sup>4</sup>[www.graphviz.org/](http://www.graphviz.org/)

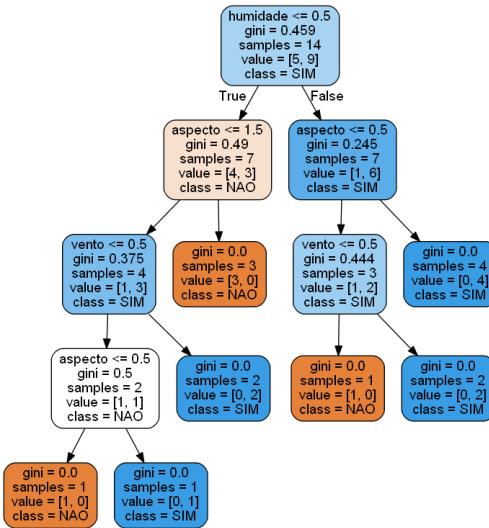


Figura 3: Representação por imagem das decisões da árvore

realizado utilizando a biblioteca *pyRAPL*<sup>5</sup>, os resultados são apresentados na Tabela 1.

Tabela 1: Tabela do consumo de energia das funções da Árvore gerada pelo CART.

Funções	Timestamp	Duration	pkg	DRAM
lerDados	1595451870.5385916	395389.07	3414603.0	465575.0
separarETransformarValores	1595451870.9344769	22600.787	192932.0	22095.0
separarTesteTreino	1595451870.9573667	15178.261	133239.0	22461.0
TreinarArvore	1595451870.9728508	3774700.796	30899152.0	2510369.0
gerarImagemArvore	1595451874.7478733	86608646.488	656030975.0	119277405.0

- lerDados: Importa o conjunto de dados usando *Pandas*.
- SepararETransformarValores: Aplica *labelEncoder*<sup>6</sup> para valores nominais e transforma dados que não podem ser lidos.
- SepararTesteTreino: Separa a porção dos dados em 80% para treino e 20% para teste.
- TreinarArvore: Chama a função da árvore utilizando *Sckit-learn*
- gerarImagemArvore: Com biblioteca *Graphviz* gera a imagem gráfica que representa as decisões da árvore

## 5 Conclusões e trabalhos Futuros

Existem diversos modelos de aprendizado de máquina que podem ser utilizados, porém nenhum é capaz de resolver todos os problema, a divisão entre modelos supervisionados e não supervisionados deixa isso bem claro. Modelos simples como regressão linear servem apenas para dados menos complexos, usando modelos mais robustos como as árvores de decisão, é possível perceber a sua eficiência em mapear resultados mesmo com dados faltantes.

<sup>5</sup><https://github.com/powerapi-ng/pyRAPL>

<sup>6</sup><https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html>

Porém, é importante olhar a parte de consumo energético destes algoritmos, pois consumo de energia e emissão de  $CO_2$  equivalentes tem se mostrado impactante [Strubell et al. 2019]. Analisando o algoritmo foi possível verificar que o maior consumo do processador e Memória RAM ocorrem nas funções “TreinarArvore” e “gerarImagenArvore”, como mostrado na Tabela 1. A primeira teve um alto consumo, pois foi aonde a árvore realizou suas decisões para o treinamento. Porém, o maior gasto ocorreu na última função, a qual é responsável por converter as decisões de árvore em uma representação gráfica, o que demanda muito processamento.

Essas funções estão em uma implementação de alto nível de programação. Assim, o próximo objetivo é implementar e detalhar o código, função por função, da árvore de regressão com o algoritmo CART, para identificar de modo ainda mais específico o consumo de cada função no código, analisando o que pode ser otimizado para evitar esse consumo de energia. Com isso será importante o estudo de perfisadores do *Python* como o cProfile<sup>7</sup>, que poderá contar com auxílio de ferramentas como Perf<sup>8</sup> e vTune<sup>9</sup>. Além disso, a análise do consumo da ferramenta pyRAPL é importante para não haver sobrecarga na medição geral.

## Agradecimentos

Agradeço imensamente a professora Mariza Ferro<sup>10</sup>, que me deu oportunidade de fazer parte do grupo ComCiDis<sup>11</sup>, e aos meus colegas de trabalho que fazem parte dessa grande equipe, e também ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo apoio financeiro.

## Referências

- [Gron 2017] Gron, A. (2017). *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, Inc., 1st edition. 2
- [Hunter 2007] Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95. 2
- [Moore II 1987] Moore II, D. H. (1987). Classification and regression trees, by leo breiman, jerome h. friedman, richard a. olshen, and charles j. stone. brooks/cole publishing, monterey, 1984,358 pages, \$27.95. *Cytometry*, 8(5):534–535. 3
- [Pedregosa et al. 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830. (Decision Tree module scikit-learn 1.10.6). 2
- [Quinlan 1986] Quinlan, J. R. (1986). Induction of decision trees. *MACH. LEARN*, 1:81–106. 3

<sup>7</sup><https://docs.python.org/3/library/profile.html>

<sup>8</sup>[https://perf.wiki.kernel.org/index.php/Main\\_Page](https://perf.wiki.kernel.org/index.php/Main_Page)

<sup>9</sup><https://software.intel.com/content/www/us/en/develop/tools/vtune-profiler.html>

<sup>10</sup><http://lattes.cnpq.br/5895202268828302>

<sup>11</sup><http://comcidis.lncc.br/>

[Quinlan 1993] Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. 3

[Strubell et al. 2019] Strubell, E., Ganesh, A., and McCallum, A. (2019). Energy and policy considerations for deep learning in nlp. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650. 5

[Wes McKinney 2010] Wes McKinney (2010). Data Structures for Statistical Computing in Python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 56 – 61. 2

## **1. Dados Gerais**

**Aplicação das ferramentas Intel Parallel Studio para modernização de código para métodos numéricos de diferenças finitas para solução de equações diferenciais parciais em arquitetura Intel Haswel/Broadwell.**

Bolsista: Gabriel Pinheiro da Costa

Orientadora: Carla Osthoff

Co-orientador: Frederico Luís Cabral

Bolsa de Iniciação Tecnológica pelo CNPq

Esse relatório cobre as atividades realizadas entre agosto de 2019 até julho de 2020

## **2. Objetivos**

O objetivo é o desenvolvimento do conhecimento na área de processamento de alto desempenho para avaliar e otimizar o desempenho de uma aplicação de computação científica em uma máquina de arquitetura paralela haswell/broadwell, com o uso das ferramentas de análise de código. O trabalho também almeja realizar publicações relevantes à comunidade científica. Os estudos desenvolvidos no período apreendido por esse resumo renderam duas publicações em fóruns de Iniciação Científica: *Aplicação das ferramentas Intel Parallel Studio para modernização de código para métodos numéricos de diferenças finitas para solução de equações diferenciais parciais em arquitetura Haswell/Broadwell*, no ERAD-RJ de 2019, e *Otimização do Método HOPMOC 1D com auxilio das ferramentas Intel Parallel Studio*, no WSCAD-WIC de 2019.

## **3. Introdução**

O trabalho almeja realizar um estudo das oportunidades de otimização de código para a solução de equações diferenciais parciais através das ferramentas do Intel Parallel Studio.

Os métodos numéricos destinados à solução dessas equações são contemplados pela possibilidade vasta do uso de ferramentas e técnicas cada vez mais efetivas, que buscam aumentar o desempenho dessas aplicações, reduzindo o tempo de execução e permitindo entradas de dados de maior magnitude.

No contexto de otimização, ganha forte destaque a técnica de paralelização, que permite que os dados sejam processados simultaneamente por mais de uma unidade de processamento, sem comprometer a qualidade do resultado final, e reduzindo o tempo necessário para chegar até esse resultado.

As ferramentas do Intel Parallel Studio trabalham com informações do comportamento das instruções do código durante a compilação e a execução. A partir dessa gama de informações é possível extrair conclusões acerca da aplicação e seu desempenho atual, e portanto nortear um plano de otimização para essa aplicação, buscando um aproveitamento ótimo do hardware utilizado.

O HOPMOC é um método numérico para solução de equações diferenciais parciais hiperbólicas, que se baseia no método Hopscotch, e no método das características, que permite a separação de variáveis através de curvas características. A estratégia do HOPMOC consiste em dividir os pontos que constituem a malha em elementos que serão calculados de maneira explícita e outros de maneira implícita, a fim de eliminar a necessidade de resolver um sistema de equações a cada passo no tempo. Dessa forma o código HOPMOC possui a seguinte estrutura geral: Um loop principal *while* onde cada iteração representa um momento no tempo, e um conjunto de loops *for* no interior do loop principal, responsáveis por realizar os cálculos explícitos e implícitos alternadamente.

```

1   ...
2   while(condição){
3       for(...){
4           //Calcula deslocamento do pulso
5       }
6       ...
7       for(...){
8           //Realiza cálculo explícito
9       }
10      ...
11      for(...){
12          //Realiza cálculo implícito
13      }
14  }
15  ...

```

Dessa forma, o trabalho de paralelização do método consiste em distribuir com eficiência a carga de trabalho presente no loop principal, mais notavelmente a carga de trabalho contida nos loops *for*. A forma como a paralelização desses loops ocorre dará origem a novas versões do HOPMOC. Cada versão do método alcança o paralelismo de uma forma distinta, mais ou menos eficiente. Um dos objetivos mais diretos desse trabalho, portanto, é desenvolver novas formas de paralelizar, novas versões do método HOPMOC, e então realizar um estudo comparativo entre elas com finalidade de destacar as estratégias mais eficientes e também os motivos que permitiram essa eficiência.

#### **4. Material e Métodos ou Metodologia**

Para a realização dos testes das diferentes implementações do método HOPMOC foi usada uma máquina contendo dois processadores Intel Xeon CPU E5 - 2698 v3 @2.30GHz, de arquitetura Haswell, totalizando 32 núcleos, uma memória total de 115097784 kB e um SO CentOS Linux 7 (Core). Todos os testes comparativos foram realizados com base em execuções contendo os mesmos parâmetros de entrada e usando os mesmos recursos de hardware, a fim de que nenhuma dessas variáveis pudesse mascarar o resultado final. Todos os valores referentes às métricas spin time, CPU time e elapsed time apresentados nos próximos capítulos foram coletados através das ferramentas Intel Parallel Studio.

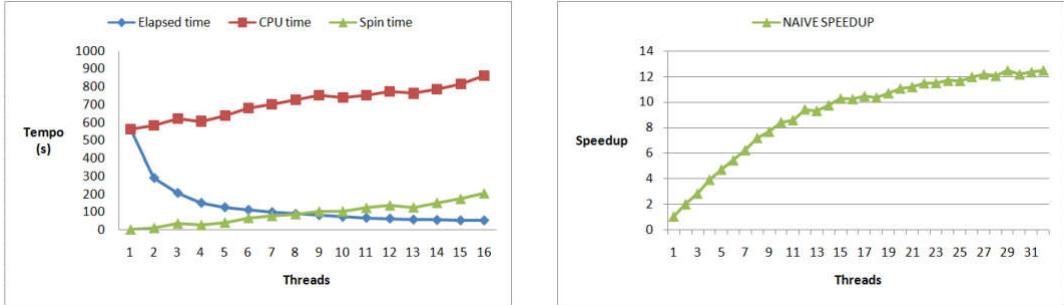
#### **5. Resultados e Discussão**

Até o momento da escrita desse relatório o método numérico HOPMOC conta com quatro versões principais: Naive, EWS-AdjSynch, MPI e Híbrido. As subseções seguintes apresentam uma breve explicação sobre cada uma dessas versões. Entretanto, devido à limitação de espaço não foi possível incluir trechos de pseudo-código e explicações mais detalhadas, mas é possível consultar as referências [2], [3], [4], [5], [6] e [7], para ter acesso a esses recursos.

##### **5.1 Novos Estudos com o HOPMOC Naive**

A versão Naive, a mais simples entre as citadas, baseia seu paralelismo exclusivamente em diretivas da biblioteca OpenMP. É aberta uma região paralela que envolve a carga de trabalho principal do código, e os loops *for* contidos dentro dessa região são paralelizados individualmente através da

diretiva *pragma omp for*. Apesar de não exigir muita complexidade e ser altamente adaptável a mudanças no código, essa versão possui uma série de limitações que impedem o ganho de desempenho à medida que crescem os recursos paralelos.

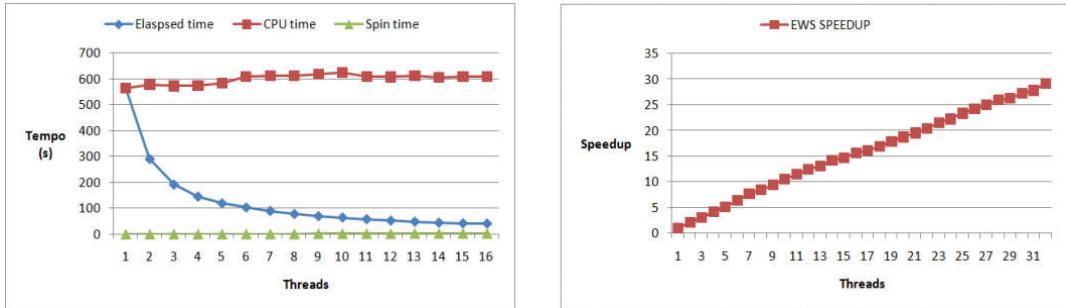


Os gráficos acima apresentam os resultados da versão Naive unidimensional obtidos em execuções elevando o número de threads na @intelknl especificada na seção 4 desse resumo. Algumas métricas como speedup e tempo de execução já foram apresentadas em estudos anteriores, entretanto o diferencial desses resultados está no estudo do comportamento de CPU Time. O CPU Time representa o tempo gasto por todas as threads envolvidas naquela execução, e em um cenário ideal deveria se manter constante. O Spin Time é o tempo gasto em barreiras de sincronização, e são responsáveis por sincronizar todas as threads ativas: as threads que chegam primeiro na barreira devem esperar que todas as demais threads também alcancem esse ponto para só então darem continuidade à execução a partir do mesmo ponto.

Os gráficos da versão Naive revelam uma forte limitação no ganho de desempenho, observada na curva de speedup que apresenta uma tendência de estabilização. O aumento do número de recursos paralelos disponíveis não se reflete em um aumento de desempenho, como se esperava. Essa limitação pode ser explicada pelas curvas de spin time e CPU time: o tempo gasto em barreiras de sincronização passa a ser maior conforme cresce o número de threads, o que leva a um aumento da soma total dos tempos de todas as threads. Esse crescimento da curva de CPU Time comprova a limitação da versão Naive em relação à escalabilidade de recursos paralelos.

## 5.2 Novos Estudos com o HOPMOC EWS-AdjSynch

Uma versão desenvolvida com fins de superar as limitações observadas na estratégia Naive foi a EWS-AdjSynch. Essa versão não mais faz uso de uma paralelização implícita e automática como na estratégia Naive, mas realiza uma divisão explícita e igualitária da malha que será processada entre as threads disponíveis. Assim a cada thread é designado um segmento de malha em que ela deverá trabalhar até o fim da execução. Entretanto, a mudança mais poderosa dessa versão está no seu mecanismo de sincronização. As barreiras explícitas que limitavam o desempenho são retiradas e dão lugar a uma sincronização entre adjacentes: cada thread é responsável por garantir a sincronia apenas entre suas duas threads vizinhas, a imediatamente anterior e a imediatamente seguinte. Com todas as threads realizando esse procedimento é possível garantir a sincronia total da execução e eliminar as barreiras explícitas.

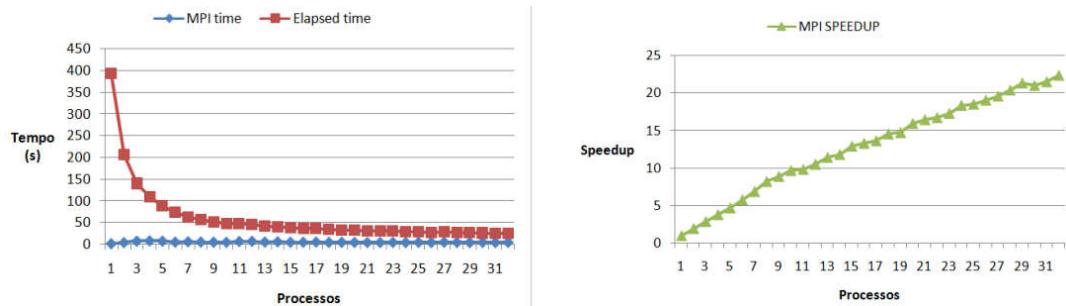


Os gráficos acima apresentam os resultados obtidos em testes com condições idênticas às daquelas presentes no caso Naive: mesma máquina, mesmos valores de entrada, mesma malha e mesmos recursos paralelos. Chama atenção o perfil linear da curva de speedup, que se mantém crescendo na mesma proporção até o máximo de threads disponíveis. Aqui já não é mais possível observar a limitação de desempenho presente na versão Naive: o speedup cresce à medida e em mesma proporção que o número de recursos paralelos, de forma próxima ao ideal.

Novamente a curva de spin time e de CPU time apresentam valores que corroboram a curva de speedup. A curva de spin time mantém um perfil muito constante e também de valores muito baixos: o tempo de espera gasto em barreiras de sincronização agora é desprezível frente à magnitude do tempo total de execução. Ainda sim essa diminuição poderia ser explicada pelo fato do Intel Parallel Studio processar o tempo gasto pelo mecanismo de sincronização adjacente criado para essa versão, não como um mecanismo de sincronização, mas sim como execução integral do código, o que portanto faria com que os tempos de spin time não servissem como prova da eficiência de tal mecanismo. Entretanto, a curva de CPU time nos mostra que a soma dos tempos finais de todas as threads tendeu a se manter constante durante todo o gráfico, um comportamento muito próximo do ideal. Isso revela que o aumento do número de recursos paralelos não implicou num aumento do spin time real e nem num aumento de tempo total, e portanto também corrobora a validade da versão EWS-AdjSync.

### 5.3 Novos Estudos com o HOPMOC MPI

Outra estratégia desenvolvida com fins de melhorar o desempenho da versão Naive foi a MPI. O HOPMOC MPI é muito semelhante ao HOPMOC EWS-AdjSync, com a diferença que, como sugerido no próprio nome, são usados processos ao invés de threads como recursos paralelos, e portanto biblioteca MPI é utilizada para gerir o paralelismo, e não mais a OpenMP. Essa versão também faz uma divisão explícita e igualitária da malha entre os processos disponíveis, com cada processo ficando responsável por uma seção de malha até o fim da execução. A comunicação e a sincronização entre os diferentes processos também é realizada somente entre os adjacentes, e é possibilitada pelas rotinas MPI *MPI\_Isend*, *MPI\_Irecv*, e *MPI\_Wait*.



A imagem acima mostra os resultados da versão MPI para os mesmos testes realizados anteriormente. A curva de speedup mantém um perfil próximo ao do observado nos resultados do HOPMOC EWS-AdjSync, conseguindo traduzir o aumento do número de processos disponíveis em aumento de desempenho. Apesar do MPI não possuir um equivalente ao CPU time da coleta em código OpenMP, o MPI time é o equivalente direto ao Spin time da versão OpenMP, e contabiliza o tempo gasto em rotinas estritamente referentes ao MPI: nesse caso o mecanismo de sincronização. A curva dessa métrica possui um perfil constante e de valores muito baixos se comparados com as outras grandezas envolvidas, de forma que é possível concluir que aqui, também, o mecanismo de sincronização adjacente foi bem sucedido em eliminar os grandes tempos gastos em barreiras de sincronização e permitir o ganho de desempenho do método quando há aumento da oferta de recursos paralelos.

#### 5.4 - A Estratégia Overthread

Uma abordagem muito promissora que pode ser aplicada na versão Híbrida do HOPMOC é a overthread. O HOPMOC Híbrido é um modelo que combina as duas versões de paralelismo usadas até então: através de threads com o OpenMP e através de processos com o MPI. A matriz original é dividida em diversas seções de malha, cada uma designada a um processo, e cada processo por sua vez divide sua fração da malha em seções ainda menores entre as threads disponíveis. A técnica de overthread consiste em definir um número de processos maior do que a quantidade de cores físicas disponíveis, criando um excedente de processos e também de threads.

Apesar de que à primeira vista a idéia pareça impossível de produzir bons resultados (uma vez que o tempo de overhead criado com a alocação e escalonamento de threads e processos derrubaria o desempenho), os valores obtidos em diversos testes apontam para o caminho contrário: após alcançar um número de processos/threads superior aos disponíveis fisicamente, o speedup apresenta tendência de subida e depois volta a estabilizar.

Os estudos para compreensão desse fenômeno ainda estão sendo desenvolvidos, e resultados parciais já foram gerados. Uma das hipóteses levantadas é de que o aumento excessivo do número de threads/processos promove também o aumento da localidade de cache, uma vez que a malha original é dividida em submalhas tão pequenas que cabem quase ou inteiramente na memória cache.

Para investigar as causas desse fenômeno estão sendo executados testes com 100, 1.000 e 10.000 iterações, com 3 abordagens Híbridas diferentes. A primeira dessas abordagens realiza o paralelismo OpenMP dentro de cada processo da mesma forma que a versão Naive. Outra abordagem faz a divisão das threads da mesma maneira que a versão EWS-AdjSync. Por último, uma terceira abordagem também faz uso da divisão explícita da carga de trabalho (EWS), mas não faz uso da sincronização entre adjacentes(AdjSync), mas sim da barreira explícita como no HOPMOC Naive.

### 6. Conclusões

Os estudos de CPU time e spin time revelam a séria limitação criada pela sincronização automática via *pragma omp barrier*, como implementada na versão Naive. Os resultados mostram que a eliminação do tempo gasto com essas barreiras (spin time) nas versões EWS-AdjSync e MPI conseguiu elevar o desempenho a valores muito próximos dos ideais. Enquanto a versão Naive alcançava um speedup máximo de 12 para 32 threads, a versão EWS-AdjSync conseguiu alcançar os 29, enquanto a MPI chegou até 23. A curva de CPU time também apresenta um perfil diferente entre a abordagem Naive e a EWS-AdjSync: enquanto na primeira a curva apresenta tendência de subida à medida que o número de threads aumenta, na segunda a curva se mantém constante, revelando um melhor uso dos recursos disponíveis.

Além disso, a estratégia overthread pode indicar um caminho promissor, ao permitir um acréscimo de desempenho notável ao exceder o número de threads utilizadas em relação à quantidade de cores disponíveis. Os resultados parciais apontam para a constância desse resultado positivo nas versões do método HOPMOC que fazem uso da divisão explícita da carga de trabalho entre as threads(EWS), mesmo quando se varia a quantidade de iterações da execução e até mesmo em máquinas diferentes. Os estudos se concentram agora na explicação do fenômeno.

As publicações geradas tanto no ERAD-RJ como no WSCAD-WIC, bem como a perspectiva promissora da nova estratégia overthread reiteram a relevância do tema e também a qualidade dos estudos realizados.

## 7.Referências

- [1] RAUBER, Thomas; RÜNGER, Gudula. **Parallel programming: For multicore and cluster systems**. Springer Science & Business Media, 2010.
- [2] TEIXEIRA, Thiago Daniel Quimas Simões. Estratégias de otimização para um Método Numérico para resolução de problemas de Convecção-Difusão. 2016. 72 f. Trabalho de Conclusão de Curso(Graduação em Sistema de Informação) - Universidade Estácio de Sá, Petrópolis, 2016.
- [3] Cabral F.L.; Osthoff C.; Costa G.; Brandão D.; Oliveira S.L.G.; Kischinhevsky M. *Tuning up TVD HOPMOC method on Intel MIC Xeon Phi Architectures with Intel Parallel Studio Tools Workshop on Applications for Multicore Architecture (WAMCA 2017)* - International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD 2017) - Campinas - São Paulo - Brasil - Outubro de 2017.
- [4] Cabral, F.L.; Osthoff, C.; Costa, G.P.; Gonzaga de Oliveira, S.L.; Brandão, D.; Kischinhevsky, M. *An OpenMP Implementation of the TVD Hopmoc Method Based on a Synchronization Mechanism Using Locks Between Adjacent Threads on Xeon Phi (TM) Accelerators*. Lecture Notes in Computer Science. Springer International Publishing, 2018, v. 3, p. 701-707. doi: [10.1007/978-3-319-93713-7\\_67](https://doi.org/10.1007/978-3-319-93713-7_67)
- [5] Cabral, F. L., Osthoff, C., Souto, R. P., Costa, G. P., de Oliveira, S. L. G., Brandão, D., & Kischinhevsky, M. (2018, September). Fine-Tuning an OpenMP-Based TVD–Hopmoc Method Using Intel® Parallel Studio XE Tools on Intel® Xeon® Architectures. In *Latin American High Performance Computing Conference* (pp. 194-209). Springer, Cham.
- [6] Costa, G., Cabral, F., & Osthoff, C. (2019). Otimização do Método HOPMOC 1D com auxílio das ferramentas Intel Parallel Studio. In *Anais Estendidos do XX Simpósio em Sistemas Computacionais de Alto Desempenho*, (pp. 41-48). Porto Alegre: SBC. doi:[10.5753/wscad\\_estendido.2019.8697](https://doi.org/10.5753/wscad_estendido.2019.8697)
- [7] da Costa, G., Osthoff, C., & Cabral, F. (2019). Aplicação das ferramentas Intel Parallel Studio para modernização de código para métodos numéricos de diferenças finitas para solução de equações diferenciais parciais em arquitetura Haswell/Broadwell. In *Anais da V Escola Regional de Alto Desempenho do Rio de Janeiro*, (pp. 29-31). Porto Alegre: SBC.

**Título do Projeto:** Gerência de Aplicações Científicas no Portal da Rede Nacional de Bioinformática  
(Bioinfo-Portal)

**Nome do bolsista:** Mayconn Luiz Bispo dos Santos

**Nome do orientador:**

D.Sc. Kary Ann del Carmen Ocaña Gautherot (Tecnologista Pleno – LABINFO/LNCC)

**Nome dos coorientadores:**

D.Sc. Antonio Tadeu Azevedo Gomes (Tecnologista Sênior – SINAPAD/LNCC)

B.Sc. Marcelo Monteiro Galheigo (SINAPAD/LNCC)

**Tipo de bolsa:** PIBIT

**Período do relatório:** 01/10/2019 a 31/07/2020

## OBJETIVOS

O principal objetivo é implementar uma nova versão otimizada da interface Web do Portal-Bioinfo. Entre os objetivos secundários temos: Realizar estudo dos códigos do Bioinfo-Portal, fazer levantamento de pontos críticos, e realizar o processo de otimização de código. Integrar as camadas Interface Web e Dados para levantar a aba de estatísticas do Bioinfo-Portal. Adaptar tecnologias analíticas Web como Google Analytics que ofereçam informação sobre audiência, aquisição, comportamento, e conversões do gateway.

## INTRODUÇÃO

O gateway científico Bioinfo-Portal (<https://bioinfo.lncc.br/>) é uma plataforma multiusuária de bioinformática criada para apoiar a comunidade científica na execução de programas e software em ambientes de computação de alto desempenho (CAD) [1-3]. Bioinfo-Portal foi desenvolvido a pedido da Rede Nacional de Bioinformática (RNBio) e do Ministério de Ciência, Tecnologia, Inovações e Comunicações (MCTIC) e atualmente gerenciado e alocado no Laboratório Nacional de Computação Científica (LNCC), como resultados das pesquisas colaborativas entre o Laboratório de Bioinformática (Labinfo, <https://www.labinfo.lncc.br/>) e o Sistema Nacional de Processamento de Alto Desempenho (SINAPAD, <https://www.lncc.br/sinapad/>). Bioinfo-Portal foi acoplado à arquitetura do supercomputador Santos Dumont (SDumont), o que permite que as execuções de aplicações de bioinformática consideradas como computacional e tempo intensivas se beneficiem de arquiteturas paralelas e distribuídas. Mais detalhes sobre a arquitetura do Bioinfo-Portal, por favor se referir a Ocaña et al 2020 [4].

O presente projeto de Iniciação Científica, dentro do contexto PIBIC/PIBIT no LNCC, apresenta uma versão de interface Web mais recente do Bioinfo-Portal que incluiu acoplar novos programas e software de bioinformática na plataforma, assim como, diversas tecnologias Web no nível programático com o objetivo de torná-lo mais interativo, seguro e eficiente. A nova versão da interface Web do Bioinfo-Portal foi integrada com sucesso e apresenta funcionalidades relacionadas à tomada de decisão, gerência de dados e aplicações, distribuição de tarefas, e um update da versão do PHP mais recente.

## METODOLOGIA

As tecnologias usadas no desenvolvimento da nova versão da interface Web do Bioinfo-Portal incluem LAMP (Linux, Apache, MySql, PHP-7), HTML, CSS, Angular JS, Java Script, e Bootstrap. Uso de tecnologias Web do Google Analytics como as APIs Reporting e GeoCharts foi também usado para a geração de gráficos e estatísticas. API do Google Geo Charts gera o gráfico com o número de acessos ao portal por região e foi também acoplada.

As atividades estão relacionadas à melhora do código fonte da interface Web e foram atualizadas as classes relacionadas a funções e métodos utilizadas no Bioinfo-Portal. Foi realizada a migração para o PHP versão 7.2 para garantir a segurança e eficiência do Bioinfo-Portal. Mais especificamente foram realizadas e finalizadas as seguintes tarefas:

- Implementação da Classe de Conexão da interface Web com o banco de dados do Bioinfo-Portal que inclui funções e métodos que têm como objetivo consultar, retornar e consumir os dados do banco de dados.
- Configuração do PHP para permitir a conexão com servidores de banco de dados mysql e postgresql externos. Para isso foi configurada a propriedade “`httpd can network connect`” do PHP. Com isso, alguns problemas de estruturação de códigos e configurações de servidor foram encontrados e resolvidos, como por exemplo, a permissão de ‘short tags’ do PHP foi reconfigurada.

Testes de desempenho e escalabilidade das aplicações de bioinformática nos ambientes computacionais do SINAPAD foram realizadas para avaliar a funcionalidade do Bioinfo-Portal.

## RESULTADOS E DISCUSSÃO

A interface Web do Portal-Bioinfo está acessível e funcional, com 10 novas aplicações de bioinformática. A atualização dos códigos para HTML 5 e PHP 7 diminuiu o número de plugins externos em várias funções. A implementação da API GeoChart do Google Analytics possibilitou o acesso às estatísticas do Bioinfo-Portal. O Bioinfo-Portal foi acessado 2.812 vezes (2.213 no Brasil) e 766 aplicações foram executadas. Outras informações que estão incluídas na interface do Bioinfo-Portal são o número de tarefas

por mês, por aplicação e por país. No final, foram implementadas com sucesso as metodologias Web e todas as atividades como relatadas na Metodologia.

Apresentamos alguns exemplos de códigos e frond-end que foram obtidos relacionados ao presente projeto. A Figura 1 apresenta o front-end do Bioinfo-Portal mostrando a aplicação de bioinformática RAxML e parâmetros a serem preenchidos pelos usuários na interface Web. A Figura 2 apresenta o número de acessos do Bioinfo-Portal realizado a través das ferramentas do Google Analytics. A Figura 3 apresenta uma amostra do código desenvolvido no API do Google Analytics no Reporting V.4 implementado na nova versão do Bioinfo-Portal. Esses gráficos fazem parte do artigo do Bioinfo-Portal Ocaña et al 2020 [4].

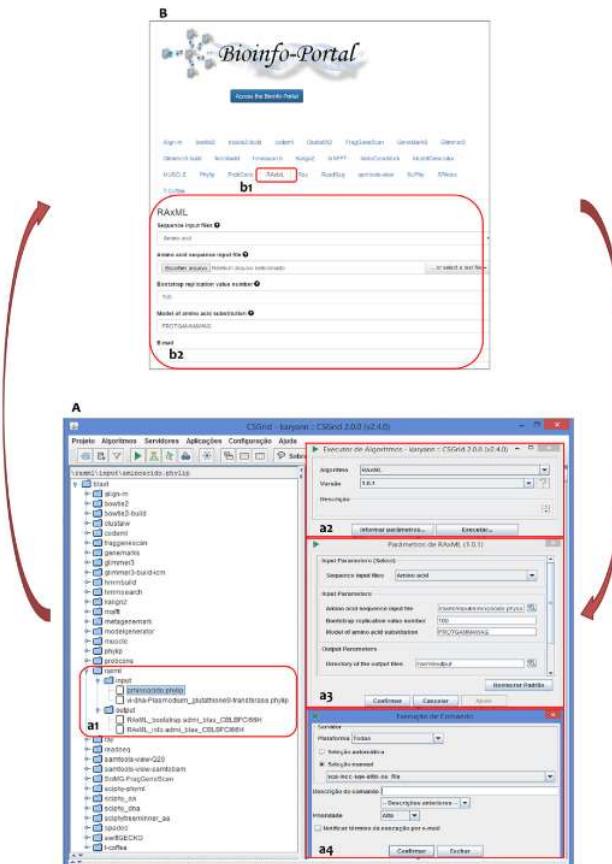


Figura 1. Front-end do RAxML no Bioinfo-Portal



Figura 2. Número de acessos do Bioinfo-Portal obtido com o Google Analytics

```

public function getCountryAccessed(){
    // Create the DateRange object.
    $dateRange = new Google_Service_AnalyticsReporting_DateRange();
    $dateRange->setStartDate("2015-06-15");
    $dateRange->setEndDate("today");

    // Create the Dimensions object.
    $dimensions = new Google_Service_AnalyticsReporting_Dimension();
    $dimensions->setName("ga:country");

    // Create the ReportRequest object.
    $request = new Google_Service_AnalyticsReporting_ReportRequest();
    $request->setViewId($this->VIEW_ID);
    $request->setDateRanges($dateRange);
    $request->setDimensions(array($dimensions));

    $body = new Google_Service_AnalyticsReporting_GetReportsRequest();
    $body->setReportRequests( array( $request ) );
    $reports = $this->analytics->reports->batchGet( $body );
    $countries = array();

    for ( $reportIndex = 0; $reportIndex < count( $reports ); $reportIndex++ ) {
        $report = $reports[ $reportIndex ];
        $header = $report->getColumnHeader();
        $dimensionHeaders = $header->getDimensions();
        $rows = $report->getData()->getRows();

        for ( $RowIndex = 0; $RowIndex < count($rows); $RowIndex++ ) {
            $row = $rows[ $RowIndex ];
            $dimensions = $row->getDimensions();

            for ( $i = 0; $i < count($dimensionHeaders) && $i < count($dimensions); $i++ ) {
                array_push($countries,$dimensions[$i]);
            }
        }
    }
    return $countries;
}

```

Figura 3. Código do API do Google Analytics no Reporting V.4 do Bioinfo-Portal

## CONCLUSÕES

A nova interface Web do Bioinfo-Portal foi concluída com sucesso. O Bioinfo-Portal está funcional com 10 novas aplicações de bioinformática. O Google Analytics foi implementado com sucesso, o que possibilita a geração de gráficos interativos com estatísticas de acessos por países ou estados de países. As camadas estão interagindo de maneira eficiente proporcionando ao Bioinfo-Portal visibilidade nacional e internacional.

Para as próximas etapas de implementação, planejamos desenvolver novos serviços para oferecer suporte ao acesso, processamento e análise de dados. O objetivo é permitir que os usuários visualizem remotamente suas execuções, suportadas pelo acesso total ao controle dos processos, o que permite aumentar o desempenho na segurança, transferência e confidencialidade dos dados. Além da integração do portal com diferentes bancos de dados.

## REFERÊNCIAS BIBLIOGRÁFICAS

1. Gesing S, Krüger J, Grunzke R, Herres-Pawlis S, Hoffmann A. Using Science Gateways for Bridging the Differences between Research Infrastructures. *Journal of Grid Computing*. 2016;14:545–57.
2. Gesing S, Nabrzyski J, Jha S. Gateways to high-performance and distributed computing resources for global health challenges. In: 2014 IEEE Canada International Humanitarian Technology Conference - (IHTC). Montreal, QC: IEEE; 2014. p. 1–5. doi:10.1109/IHTC.2014.7147530.

3. Gomes ATA, Bastos BF, Medeiros V, Moreira VM. Experiences of the Brazilian national high-performance computing network on the rapid prototyping of science gateways: SGW2013 SPECIAL ISSUE. *Concurrency and Computation: Practice and Experience*. 2015;27:271–89.
4. Ocaña K, Galheigo M, Osthoff C, Gadelha L, Porto F, Gomes T, Oliveira D, Vasconcelos A. BioinfoPortal: A scientific gateway for integrating bioinformatics applications on the Brazilian national high-performance computing network. In Press. *Future Generation Computer Systems*, 2020. <https://doi.org/10.1016/j.future.2020.01.030>.

# Título do Projeto: Implementação de estrutura de indexação de dados para otimizar as consultas no SAVIME

Nome do bolsista: André Muniz Demori

Nome do orientador: Fabio Porto

Tipo de bolsa: IC

Período do relatório: 01/08/2019 – 31/07/2020

## Objetivo:

Esse trabalho teve como objetivo otimizar as consultas realizadas no sistema SAVIME que é um SGBD que trabalha com matrizes multidimensionais denominadas (TARs) através de estruturas de indexação de dados que são usadas para evitar uma varredura completa e sequencial no data set construindo uma estrutura hierárquica a ser percorrida quando um elemento ou um conjunto de elementos é consultado no banco de dados.

## Introdução:

Este trabalho se contextualiza na otimização que estruturas de indexação de dados podem trazer nas execuções de consultas do sistema SAVIME. O sistema funciona de forma semelhante a um SGBD, tendo como modelo de dados matrizes multidimensionais (TARs). Seu objetivo é realizar consultas e visualizar dados de simulações numéricas, mas também funcionar como um sistema analítico de gerenciamento de banco de dados de matriz em memória. A estrutura de indexação usada foi a SPST-Index que é uma estrutura projetada para database cracking, porém com algumas funções adicionais, foi transformada nesse trabalho em uma estrutura para realizar consultas em grandes data sets no SAVIME.

## Metodologia:

Este relatório apresenta a implementação da estrutura de indexação SPST-Index (Self Pruning Splay Tree - Index) no SAVIME e como essa estrutura foi modificada para atender às necessidades de consulta do SAVIME.

SPST-Index foi inicialmente desenvolvida para armazenar os índices das colunas em aplicações de Database Cracking. Essa estrutura permite manter os dados mais recentemente acessados no topo da árvore (Figura 1).

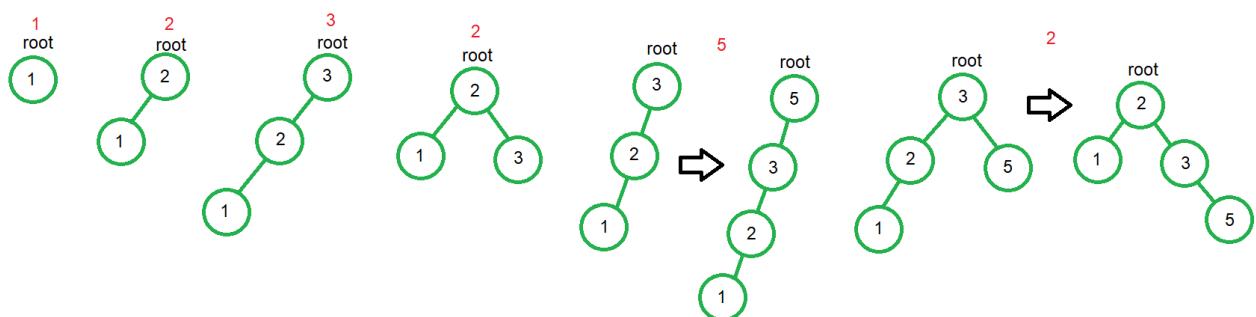


Figura 1 – Inserção na SPST-Index

Nesse trabalho essa estrutura foi transformada para que pudesse realizar consultas em um dada set de atributos. Para isso algumas modificações foram feitas na estrutura. Cada nó da árvore foi alterado para armazenar uma lista com posições (Figura 2). Essas posições são relativas à posição de cada elemento no array que representa o data set. Esse data set tem o mesmo tamanho do bitMask do SAVIME que armazena em suas posições 1 para elementos que fazem parte do conjunto resposta e 0 para os que não fazem (Figura 3). A ideia é otimizar a inserção dos elementos que fazem parte do conjunto resposta da consulta no bitMask. Atualmente o SAVIME realiza uma busca sequencial comparando elemento por elemento. Uma estrutura de indexação deveria retornar os elementos que fazem parte do intervalo da consulta e então preencher o bitMask com essa lista de posições.

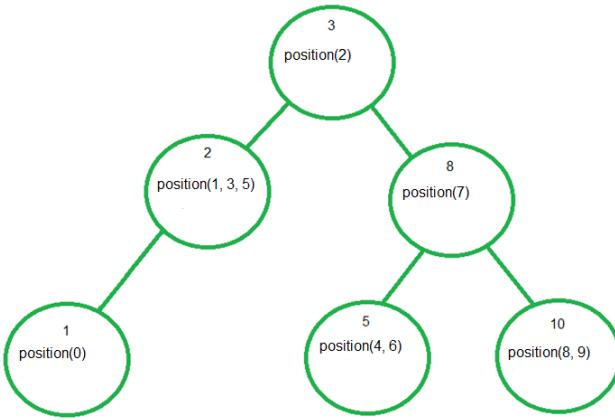


Figura 2 - Nós da árvore e suas respectivas listas com as posições dos elementos no data set.

```

bitmask
[0] = 0
[1] = 0
[2] = 0
[3] = 0
[4] = 0
[5] = 0
[6] = 0
[7] = 1
[8] = 1
[9] = 1

```

Figura 3 - bitMask, onde 7, 8 e 9 fazem parte do conjunto resposta

Para realizar as consultas necessárias no SAVIME com o auxílio da estrutura de indexação utilizando os operadores lógicos ( $>$ ,  $<$ ,  $\geq$ ,  $\leq$ ), foram criadas funções no código da estrutura que possibilitasse esses tipos de operações.

#### Implementação de filtro na SPST-Index: Funções Greater, Less, One\_Child e Two\_Childs:

A função de filtro recebe como parâmetros um valor de referência, a operação desejada ( $>$ ,  $<$ ,  $\geq$ ,  $\leq$ ), a árvore e o um objeto que vai possuir a lista com o conjunto resposta da consulta.

Primeiro o nó com o valor de referência é encontrado na árvore. Então são feitas verificações para analisar se ele tem filhos à esquerda e à direita e se esses elementos são maiores ou menores que o elemento de busca. Portanto, depois de encontrar o elemento passado como parâmetro, esse elemento é passado como parâmetro como a raiz da árvore para a função **Greater** se o operador passado for " $>$ " ou **Less**, se o operador passado for " $<$ ". Se o operador passado for  $\leq$  ou  $\geq$ , ele retorna o valor da raiz e também executa as funções Less ou Greater. Essas funções verificam se existe um nó filho à direita ou à esquerda da raiz e resgatam todos os valores através de funções recursivas.

Durante a consulta, se a raiz possuir filho à direita, significa que existem valores maiores que ele, então a raiz se torna o primeiro elemento à direita, e então é verificado se existem elementos à esquerda e à direita dele. Todos os valores que fizerem parte dessa sub árvores serão maiores que o elemento passado como parâmetro. A função **Two\_childs** é executada para retornar os valores dos filhos de um determinado nó se ele possuir dois filhos. a função **One\_child** é executada se o nó possuir apenas um filho. Essas duas funções são chamadas recursivamente dentro delas mesmas, varrendo todas as sub-árvores.

Conforme as sub árvores são varridas, os elementos que fazem parte do conjunto resposta são inseridos em uma lista que contém todas as posições do data set cujos elementos devem preencher o bitMask com 1.

## **create\_dataset():**

A opção de se utilizar estruturas de indexação será dada no operador create\_dataset nas três situações possíveis:

- Quando se passa os valores do data set por parâmetro diretamente

```
savimec 'create_dataset(«vec:int:2», literal(1, 2, 3, 4, 5, 6, 7, 8, 9, 10), «spst-index»);'
```

- Quando se passa uma expressão

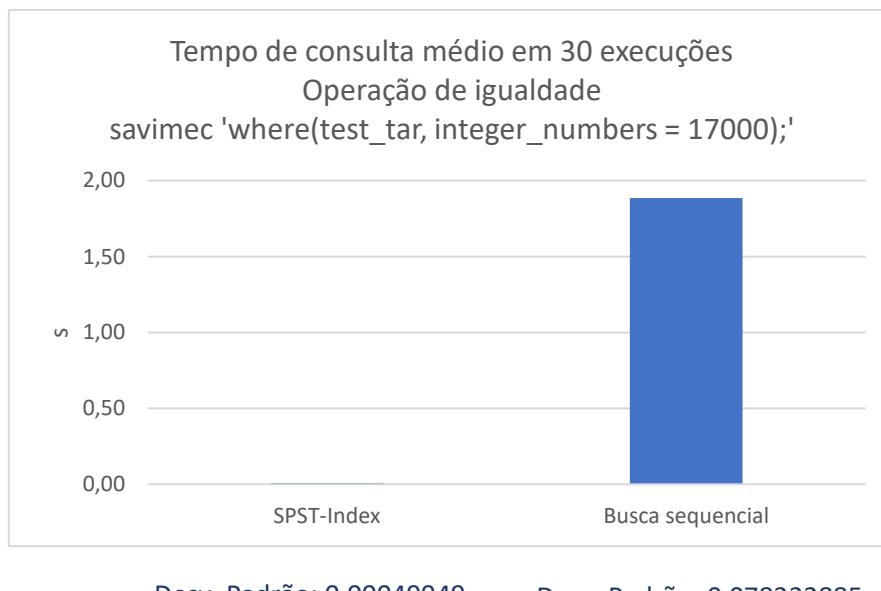
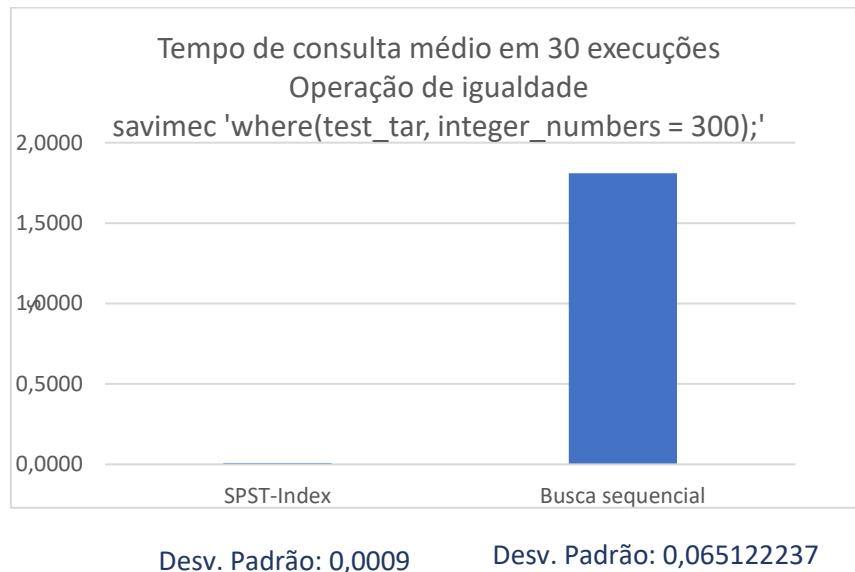
```
savimec 'create_dataset("dpart1:int", "2:2:6:1", "spst-index");'
```

- Quando se passa um arquivo

```
savimec 'create_dataset("base:double", "@'$pwd'/'base", "spst-index");'
```

## **Resultados de testes preliminares de desempenho em comparação com a busca sequencial:**

- Data set com muitas repetições - 31.525.380 números inteiros até 30.000

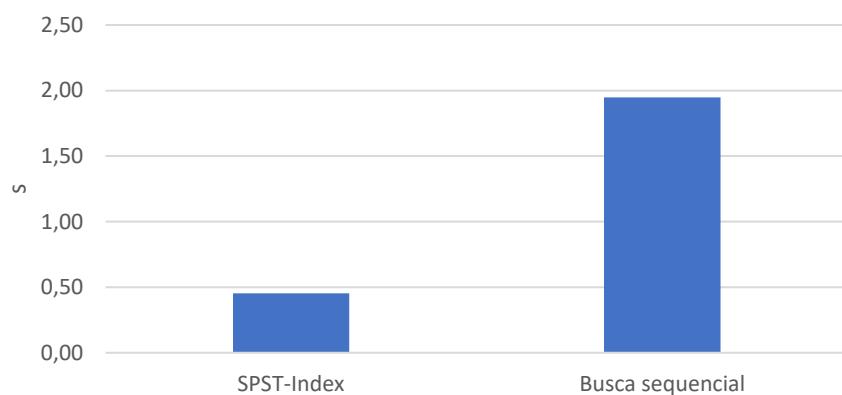


Tempo de consulta médio em 30 execuções  
Operação de igualdade  
savimec'where(test\_tar, integer\_numbers = 28000);'



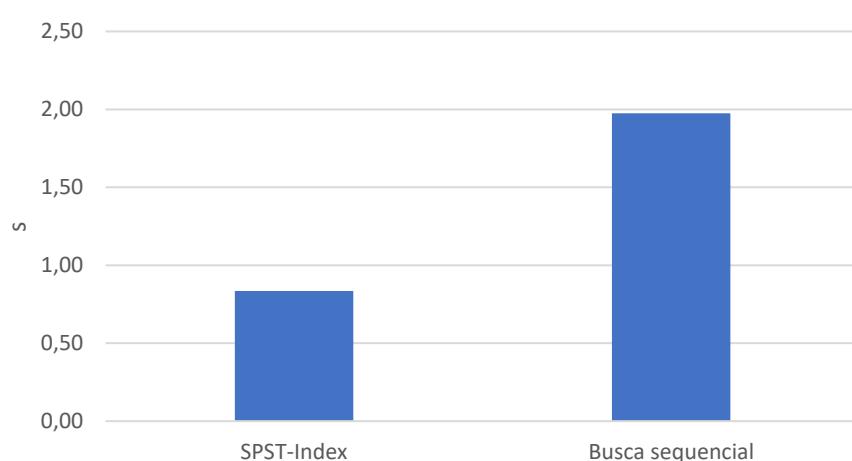
Desv. Padrão: 0,001161076 Desv. Padrão: 0,219440135

Tempo de consulta médio em 30 execuções  
Operação (> maior que)  
savimec 'where(test\_tar, integer\_numbers > 29500);'



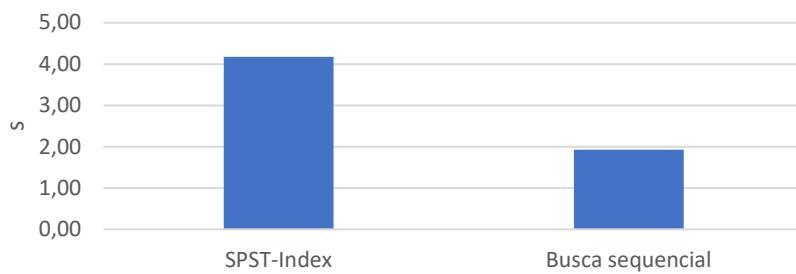
Desv. Padrão: 0,10207833 Desv. Padrão: 0,101384345

Tempo de consulta em 30 execuções  
Operação (< menor que)  
savimec 'where(test\_tar, integer\_numbers < 1000);'



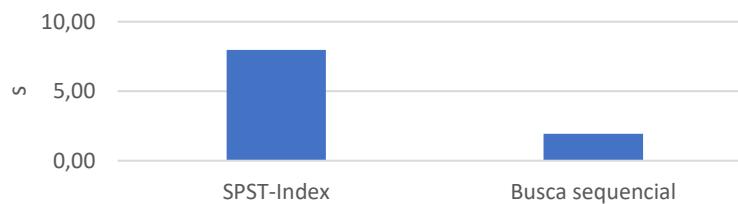
Desv. Padrão: 0,207247861 Desv. Padrão: 0,07386468

Tempo de consulta médio em 30 execuções  
 Operação (> maior que)  
 savimec 'where(test\_tar, integer\_numbers >  
 25000);'



Desv. Padrão: 0,969900497 Desv. Padrão: 0,111557853

Tempo médio em 5 execuções  
 Operação (< menor que)  
 savimec 'where(test\_tar, integer\_number <  
 10000);'



Desv. Padrão: 1,653627043 Desv. Padrão: 0,129053525

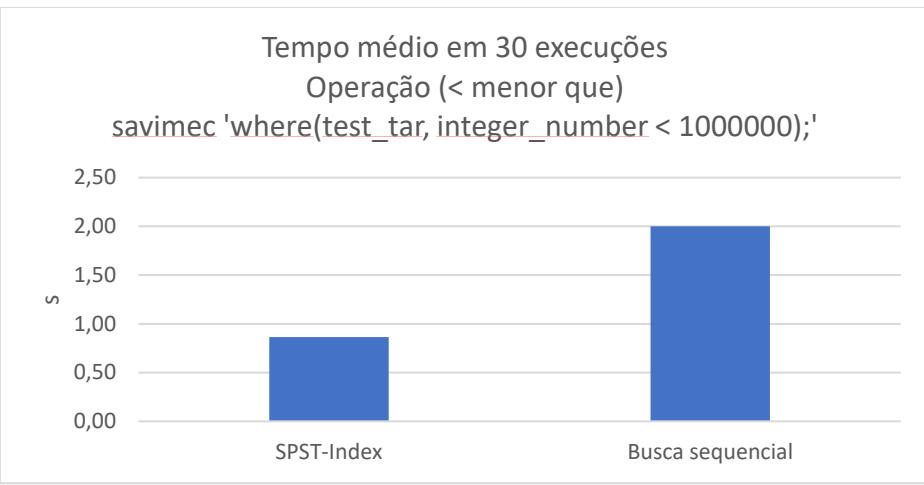
Com um intervalo maior de atributos, a estrutura de indexação começou a se mostrar ineficiente, então foi decidido criar um data set com menos repetições para ver se as cópias da lista de cada nó da árvore para a lista de resultados da consulta estava influenciando.

- Data set com muitas repetições - 31.525.380 números inteiros até 31.525.380

Tempo médio em 30 execuções  
 Operação (< menor que)  
 savimec 'where(test\_tar, integer\_number < 10000);'

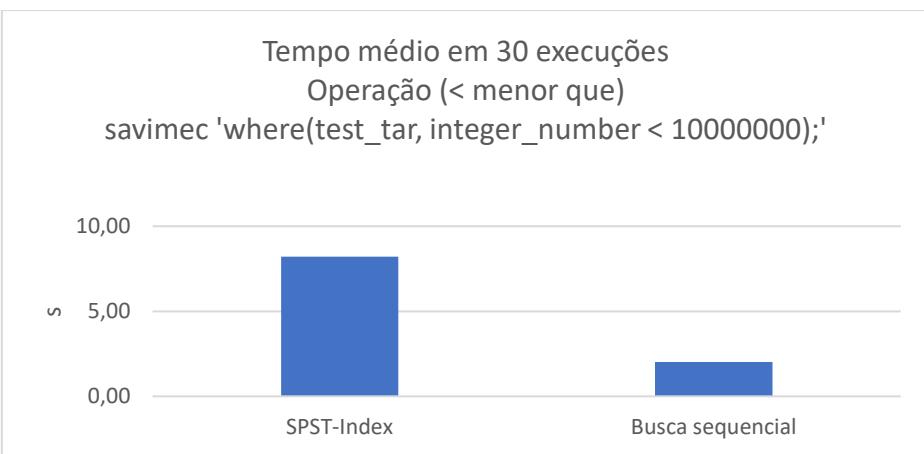


Desv. Padrão: 0,003887671 Desv. Padrão: 0,09624456



Desv. Padrão: 0,15429887      Desv. Padrão: 0,101242421

A princípio a estrutura de indexação mostrou uma melhora de desempenho com data sets com poucas repetições, mas conforme o intervalo foi aumentando a ineficiência voltou a aparecer.



Desv. Padrão: 1,445013896      Desv. Padrão: 0,113217594

## Conclusão:

Posso concluir que o objetivo de implementar uma estrutura de indexação para otimizar a busca por atributos no SAVIME foi concluída em parte. A estrutura escolhida com as implementações feitas apresentou melhora no desempenho da busca apenas para um pequeno intervalo no data set. Se esse intervalo de busca aumentar pode chegar a ter um desempenho pior que uma busca sequencial.

Agora talvez fosse melhor procurar como melhorar a busca nessa árvore ou outro código como uma B+ ou R+ e ajustar esse código ao SAVIME para buscar uma alternativa que diminuisse o tempo de busca com um intervalo de dados maior. Também vale ressaltar que a criação do data set ganhou um tempo considerável utilizando a estrutura de indexação. Mas como o data set só é construído uma única vez e as consultas são feitas várias vezes, me parece ser um overhead necessário e que vale a pena.

## Referências bibliográficas:

Lustosa, H., Porto, F. & Valduriez, P. (2019). SAVIME: A Database Management System for Simulation Data Analysis and Visualization. SBBD 2019, Fortaleza, CE.

Lustosa, H., & Porto, F. (2019). SAVIME: A Multidimensional System for the Analysis and Visualization of Simulation Data. arXiv preprint arXiv:1903.02949.

Holanda, Pedro & Almeida, Eduardo. (2017). SPST-Index: A Self-Pruning Splay Tree Index for Caching Database Cracking.

**Título do Projeto:**

Bioinfo-BD: Integração Analítica das Aplicações no Bioinfo-Portal com Bancos de Dados Relacionais

**Nome do bolsista:**

Guilherme da Silva Vieira

**Nome do orientador:**

D.Sc. Kary Ann del Carmen Ocaña Gautherot (Tecnologista Pleno – LABINFO/LNCC)

**Nome dos coorientadores:**

D.Sc. Fábio André Machado Porto (Tecnologista Sênior – DEXL/LNCC)

D.Sc. Douglas Ericson Marcelino de Oliveira (Pós-Doutor – DEXL/LNCC)

**Tipo de bolsa:** PIBIC

**Período do relatório:** 01/10/2019 a 31/07/2020

## OBJETIVOS

O principal objetivo é levantar os requisitos para a modelagem conceitual e implementação do banco de dados na camada de dados no gateway científico Bioinfo-Portal. Mais especificamente e dentre os objetivos específicos foram propostos: (1) implementar um modelo de Banco de Dados conceitual no apoio ao Portal-Bioinfo e suportado por tecnologias de sistema de gerência de banco de dados (SGBD), acoplado aos recursos computacionais do SINAPAD e da infraestrutura do supercomputador Santos Dumont, (2) implementar funcionalidades de gerência dos dados de proveniência gerados pelas aplicações do portal por meio de consultas no banco de dados e (3) aplicar testes de desempenho para validar a consistência do banco de dados.

## INTRODUÇÃO

O Projeto de Iniciação Científica (IC) se desenvolveu no nível de estágio supervisionado sob coordenação da orientadora Kary Ocaña (LNCC) e com bolsa IC financiada pelo CNPq. O mesmo se enquadra nas pesquisas relacionadas aos projetos institucionais do LNCC, especificamente no desenvolvimento do Portal-Bioinfo (<https://bioinfo.lncc.br/>) que visa a execução em larga escala de aplicações de bioinformática usando recursos computacionais paralelos e distribuídos a fim de diminuir o grande tempo de processamento das execuções.

O gateway científico Bioinfo-Portal é uma plataforma multiusuária de bioinformática criada para apoiar a comunidade científica na execução de programas e software em ambientes de computação de alto desempenho (CAD) [1-3]. Bioinfo-Portal foi desenvolvido a pedido da Rede Nacional de Bioinformática (RNBio) e do Ministério de Ciência, Tecnologia, Inovações e Comunicações (MCTIC) e atualmente gerenciado e alocado no Laboratório Nacional de Computação Científica (LNCC), como resultados

das pesquisas colaborativas entre o Laboratório de Bioinformática (Labinfo, <https://www.labinfo.lncc.br/>) e o Sistema Nacional de Processamento de Alto Desempenho (SINAPAD, <https://www.lncc.br/sinapad/>). Bioinfo-Portal foi acoplado à arquitetura do supercomputador Santos Dumont (SDumont, <https://sdumont.lncc.br/>), o que permite que as execuções de aplicações de bioinformática consideradas como computacional e tempo intensivas se beneficiem de arquiteturas paralelas e distribuídas. Mais detalhes sobre a arquitetura do Bioinfo-Portal, por favor se referir a Ocaña et al 2020 [4].

Uma questão importante em gateways científicos é orquestrar tarefas em um conjunto distribuído de recursos com custos minimizados, garantindo integridade e consistência nas informações. Uma palavra chave é proveniência que está relacionada às informações de um experimento científico e a metadados como aqueles específicos do experimento, ao desempenho, ou o sistema usado para a execução e análise dos dados [5-7]. Banco de dados é uma área consolidada fundamentada na teoria do Modelo de Bancos de Dados e nos Sistemas de Gerência de Bancos de Dados (SGBD). Banco de dados são usados em diferentes atividades e áreas como: (i) sítios web como Google, Facebook, Amazon; (ii) empresas para manter informações sobre negócios; (iii) pesquisas científicas como Astronomia (Sloan Digital Sky Survey, Dark Energy Survey, Large Synoptic Survey Telescope), Biodiversidade (Global Biodiversity Information Facility) e Genômica: (GenBank, DNA Databank of Japan, European Molecular Biology Laboratory), entre outros.

A presente proposta visa implementar funcionalidades de gerência e armazenamento de dados científicos, de proveniência e de desempenho do Bioinfo-Portal em ambientes de PAD pelo qual o problema será atacado em três frentes: (i) modelar um banco de dados para o Bioinfo-Portal e implementar um sistema de gerência de banco de dadao (SGBD) acoplado ao SDumont e ao SINAPAD; (ii) Aplicar testes de desempenho e escalabilidade em larga escala para validar a consistência do banco de dados e (iii) acoplar a arquitetura do item (i) a uma interface Web inteligente que consuma informações do banco de dados. Toda a gerência será guiada por dados de proveniência, em especial dados de domínio específico da bioinformática a serem consultados em tempo real de uma forma organizada via SQL. Esses dados “escondem” padrões interessantes e úteis para a descoberta de padrões de biológicos e computacionais que podem alavancar investigações biomédicas e de otimização dos processos no

## METODOLOGIA

Modelado o modelo conceitual baseado na arquitetura do sistema do Bioinfo-Portal, especificamente da camada de Banco de Dados, foi possível dar início ao desenvolvimento do mapeamento dos dados para a implementação do modelo lógico. O banco de dados foi modelado com o programa dbdesigner. O projeto físico foi iniciado utilizando o PostgreSQL v11.2 como Sistema de Gerência de Banco de Dados (SGBD) relacional Open Source e pgAdmin v4.3 como plataforma de desenvolvimento e gerência Open Source, muito usada e com mais recursos para o PostgreSQL. Utilizando a

ferramenta online Lucidchart foi possível montar o modelo lógico para banco de dados relacional, apresentando sete tabelas e com suas cardinalidades definidas.

## RESULTADOS E DISCUSSÃO

Sobre o desenvolvimento do banco de dados, primeiramente foi necessário o levantamento bibliográfico sobre a gerência e modelos de banco de dados, com foco no estudo e implementação da entidade-relacionamento. A Figura 1 mostra o modelo do banco de dados desenvolvido e as respectivas entidades, que foram acopladas e desenvolvidas no presente Projeto de IC. A entidade Arquivo se refere a entrada que será consumida por uma aplicação e relacionada a entidade Execução que consome/gera dados de proveniência do portal.

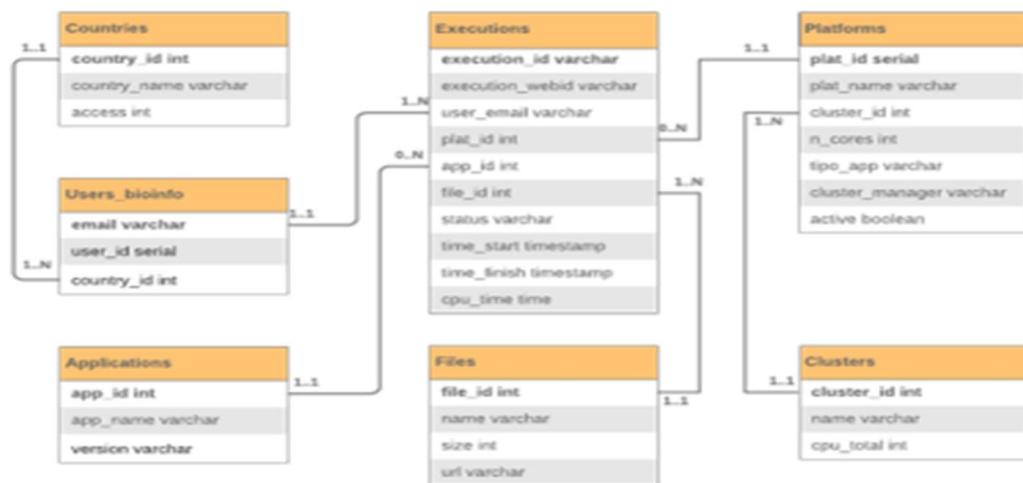


Figura 1. Modelo entidade-relacionamento desenvolvido para o Portal-Bioinfo.

### Estudo de Caso:

- Mapeamento do modelo conceitual ER: Arquivo e Execução são as únicas entidades existentes no Portal-Bioinfo (Figura 1)
- *Arquivo (tipo, nomeArquivo, caminho, tamanho); Execução (estadoFinal, tempo de execução); Usado em (nomeArquivo, estadoFinal, tipo de uso)*
- *Arquivo* é responsável por armazenar as informações do dado de entrada, tais como: tipo de entrada, nome e tamanho do arquivo e o local de armazenamento do mesmo
- *Execução* armazena informações de tempo de execução (início e fim) e estado da execução (em espera, executando, finalizado com erro e finalizado sem erro)
- *Usado* é o relacionamento entre as entidades Arquivo e Execução

O banco de dados foi implementado na arquitetura Bioinfo-Portal, definido com sete tabelas principais: Executions, Countries, Users\_bioinfo, Applications, Files, Platforms e Clusters. A seguir apresentamos algumas consultas representando a criação de tabelas, inserção de dados e a consulta dos dados extraídos do banco de dados modelado.

### Consulta de criação da tabela Executions:

```

CREATE TABLE executions(
    execution_id varchar,
    execution_webid varchar,
    user_email varchar REFERENCES users_bioinfo(email),
    plat_id int REFERENCES platforms(plat_id),
    app_id int REFERENCES applications(app_id),
    file_id int REFERENCES files(file_id),
    status varchar,
    time_start timestamp,
    time_finish timestamp,
    cpu_time time,
    PRIMARY KEY (execution_id)
);

```

Foram implementadas as funções para o preenchimento das tabelas e a expansão e backup do banco.

#### **Consulta de criação da função de inserção na tabela Executions:**

```

CREATE OR REPLACE FUNCTION insert_execution(execution_id varchar,
execution_webid varchar, user_email varchar, plat_id int, app_id int,
file_id int, status varchar, time_start timestamp, time_finish timestamp,
cpu_time time)
RETURNS text AS $$

BEGIN
INSERT INTO executions (execution_id, execution_webid, user_email, plat_id,
app_id, file_id, status, time_start, time_finish, cpu_time) VALUES
(execution_id, execution_webid, user_email, plat_id, app_id, file_id,
status, time_start, time_finish, cpu_time);
RETURN 'Execução cadastrada com sucesso';
END;
$$ LANGUAGE plpgsql;

```

Ao final, foram implementadas com sucesso as funções para a gerência e análise de dados do Bioinfo-Portal, apresentadas abaixo:

#### **Consulta para obter o número de acessos e execuções totais:**

```

SELECT COUNT (access) FROM countries;
&
SELECT COUNT (app_id) FROM executions;
Consulta para listar as execuções de um usuário:
SELECT execution_id, time_finish, time_start
FROM executions
WHERE user_email = '@email.com';
Consulta para listar as execuções com falhas:
SELECT execution_id, time_finish, time_start
FROM executions
WHERE status = 'Falha';
Consulta para obter o número de usuários no Brasil:
SELECT COUNT (user_id)
FROM users_bionfo
WHERE country_id = 1';

```

## **CONCLUSÕES**

Apresentamos resultados sobre o desenvolvimento e implementação de um banco de dados acoplado à infraestrutura do Portal-Bioinfo, que permitirá armazenar dados de proveniência, por exemplo, das execuções de todas as aplicações do portal. Nesse projeto salientamos que esse modelo de entidade-relacionamento para o Portal-Bioinfo foi implementado, o que até então não existia. Dessa maneira, o desenvolvimento e implementação de um modelo conceitual ER para o Portal-Bioinfo permitirá a garantir a consistência na gerência e acesso aos dados científicos, de proveniência e de desempenho. Esse modelo irá sustentar o acesso aos dados para estudos realizados em paralelo no Bioinfo-Portal, análises de predição via aprendizado de máquina e o desenvolvimento da interface Web inteligente.

## **REFERÊNCIAS BIBLIOGRÁFICAS**

1. Gesing S, Krüger J, Grunzke R, Herres-Pawlis S, Hoffmann A. Using Science Gateways for Bridging the Differences between Research Infrastructures. *Journal of Grid Computing*. 2016;14:545–57.
2. Gesing S, Nabrzyski J, Jha S. Gateways to high-performance and distributed computing resources for global health challenges. In: 2014 IEEE Canada International Humanitarian Technology Conference - (IHTC). Montreal, QC: IEEE; 2014. p. 1–5. doi:10.1109/IHTC.2014.7147530.
3. Gomes ATA, Bastos BF, Medeiros V, Moreira VM. Experiences of the Brazilian national high-performance computing network on the rapid prototyping of science gateways: SGW2013 SPECIAL ISSUE. *Concurrency and Computation: Practice and Experience*. 2015;27:271–89.
4. Ocaña K, Galheigo M, Osthoff C, Gadelha L, Porto F, Gomes T, Oliveira D, Vasconcelos A. BioinfoPortal: A scientific gateway for integrating bioinformatics applications on the Brazilian national high-performance computing network. In Press. *Future Generation Computer Systems*, 2020. <https://doi.org/10.1016/j.future.2020.01.030>.
5. Moreau, L., Freire, J., Futrelle, J., McGrath, R. E., Myers, J., Paulson, P. (2008), “The Open Provenance Model: An Overview”, In: Freire, J., Koop, D., Moreau, L. (eds), *Provenance and Annotation of Data and Processes*, Springer Berlin Heidelberg, p. 323–326.
6. Moreau, L., Missier, P. and Belhajjame, B. (2013) “The PROV Data Model and Abstract Syntax Notation”.
7. Garijo, D. and Gil, Y. (2014) “The OPMW-PROV Ontology”, <http://www.opmw.org/model/OPMW/>.

**Título do Projeto:**

Avaliação de Métodos de Aprendizado de Máquina em Aplicações de Bioinformática

**Nome do bolsista:**

Isabela Canuto Ramos

**Nome do orientador:**

D.Sc. Kary Ann del Carmen Ocaña Gautherot (Tecnologista Pleno – LABINFO/LNCC)

**Nome dos coorientadores:**

D.Sc. Fábio André Machado Porto (Tecnologista Sênior – DEXL/LNCC)

D.Sc. Douglas Ericson Marcelino de Oliveira (Pós-Doutor – DEXL/LNCC)

**Tipo de bolsa:** PIBIC

**Período do relatório:** 01/10/2019 a 31/07/2020

## **OBJETIVOS**

O principal objetivo é analisar dados científicos e de desempenho de programas de bioinformática executados em ambientes de computação de alto desempenho (CAD) por meio do uso de técnicas aprendizado de máquina (AM) usadas para criar modelos que apoiem a decisão automatizada do recurso computacional mais eficiente para determinado experimento. Mais especificamente e dentre os objetivos específicos foram propostos: (1) Realizar uma análise comparativa das técnicas de AM para criar modelos de predição e classificação no apoio à análise a dados científicos e de desempenho de experimentos de bioinformática e (2) Realizar análises estatísticas para validar os modelos construídos.

## **INTRODUÇÃO**

O Projeto de Iniciação Científica (IC) se desenvolveu no nível de estágio supervisionado sob coordenação da orientadora Kary Ocaña (LNCC) e com bolsa IC financiada pelo CNPq. O mesmo se enquadra nas pesquisas relacionadas aos projetos institucionais do LNCC, especificamente no desenvolvimento do Portal-Bioinfo (<https://bioinfo.lncc.br/>) que visa a execução em larga escala de aplicações de bioinformática usando recursos computacionais paralelos e distribuídos a fim de diminuir o grande tempo de processamento das execuções.

O gateway científico Bioinfo-Portal é uma plataforma multiusuária de bioinformática criada para apoiar a comunidade científica na execução de programas e software em ambientes de computação de alto desempenho (CAD) [1-3]. Bioinfo-Portal foi desenvolvido a pedido da Rede Nacional de Bioinformática (RNBio) e do Ministério de Ciência, Tecnologia, Inovações e Comunicações (MCTIC) e atualmente gerenciado e

alocado no Laboratório Nacional de Computação Científica (LNCC), como resultados das pesquisas colaborativas entre o Laboratório de Bioinformática (Labinfo, <https://www.labinfo.lncc.br/>) e o Sistema Nacional de Processamento de Alto Desempenho (SINAPAD, <https://www.lncc.br/sinapad/>). Bioinfo-Portal foi acoplado à arquitetura do supercomputador Santos Dumont (SDumont, <https://sdumont.lncc.br/>), o que permite que as execuções de aplicações de bioinformática consideradas como computacional e tempo intensivas se beneficiem de arquiteturas paralelas e distribuídas. Mais detalhes sobre a arquitetura do Bioinfo-Portal, por favor se referir a Ocaña et al 2020 [4].

Experimentos científicos de bioinformática em larga escala são complexos, pois envolvem dados científicos volumosos, heterogêneos e geograficamente dispersos pelo que ambientes de CAD são requeridos. Dados de científicos de proveniência [5], [6] se referem à toda a informação contida nos metadados associados ao ciclo de vida de um experimento científico [7] e é um conceito chave a ser considerado. Esse dados de proveniência envolvem os metadados de uma área determinada, como dados de execução do experimento. Todas as informações de desempenho e de proveniência das aplicações de bioinformática são centralizadas em um banco de dados e foram acessadas por meio de consultas.

Técnicas de predição baseadas em AM [8–10] utilizam de um conjunto de dados, chamados de treinamento, para construção de um modelo que se aproxime do conjunto de dados fornecido. Uma vez treinados, os modelos permitem realizar previsões envolvendo classificações de novas instâncias de dados de entrada, assim como a estimativa de variáveis dependentes. Neste contexto, várias técnicas têm sido aplicadas: como regressões lineares e lógicas, redes neurais, árvores de decisão, dentre outras. No trabalho [11], foi utilizada a técnica de classificação para obtenção da configuração de parâmetros do SPARK que levam ao melhor tempo de execução de um workflow da área da astronomia.

Dessa maneira, será avaliado o desempenho das execuções de programas no Bioinfo-Portal por meio de técnicas de AM para gerar modelos que visem tomadas de decisão eficientes que incluem parâmetros como o melhor tipo de cluster, o número efetivo de núcleos para processamento, e outras informações. Esses modelos serão usados para prever a melhor configuração para a execução de um experimento e pode apoiar a melhora no desempenho de execuções, resultando no uso efetivo das plataformas de CAD, incluindo o supercomputador SDumont.

## METODOLOGIA

As pesquisas relacionadas ao presente Projeto IC estão relacionadas ao desenvolvimento e implementação de uma nova versão do Portal-Bioinfo (Portal de Bioinformática do LNCC), mais segura e eficiente, adaptada e acoplada para ser executada no SDumont. Nesse trabalho de IC em específico, essa nova versão aborda técnicas para obter alocações eficientes (o melhor possível) das máquinas para a execução de experimentos científicos

no supercomputador. Esse objetivo principal se baseia em outros objetivos específicos do projeto como são explorar técnicas de AM e MD e implementar um módulo para análise de dados do Portal-Bioinfo que possibilite a criação de modelos de predição e classificação.

O projeto apresenta as seguintes atividades desenvolvidas durante o período da bolsa:

- Mapeamento do modelo do banco de dados e a estrutura do Bioinfo-Portal e obtenção dos dados de proveniência e desempenho por meio de consultas à base de dados MySQL e PostgreSQL.
- Organização e formatação dos dados obtidos no formato (\*.csv) e aplicar técnicas de decisão de árvore para a construção de modelos preditivos usando os programas Orange e sci kit learn.
- Criação dos modelos de predição para esses programas.
- Validar as previsões que otimizem os resultados de desempenho (eficiência maior de 75%) e usar essas configurações nos scripts de submissão dos clusters, o que permitirá melhor alocação e uso da infraestrutura computacional.

## RESULTADOS E DISCUSSÃO

Os experimentos realizados com os modelos de predição obtidos com as ferramentas Orange data mining e scikit learn apresentaram resultados satisfatórios. Foram geradas árvores de decisão, que utilizaram dados de tarefas executadas no SDumont, partindo do critério “class” (usado como target na classificação), que foi dividido em três categorias (baixo, alto, medio). Sendo assim, foi determinado como ponto de corte um cut-off de 75% para determinar o uso otimizado (parâmetros a serem levados em consideração) no uso dos clusters do SDumont.

Os resultados obtidos foram parte do trabalho apresentado na forma de Artigo Completo em Workshop Internacional:

- Ocaña, Kary & Osthoff, Carla & Coelho, Micaella & Galheigo, Marcelo & Canuto, Isabela & Oliveira, Douglas & de Oliveira, Daniel. (2020). Performance Evaluation of Parallel Inference of Large Phylogenetic Trees in Santos Dumont Supercomputer: A Practical Approach. 10.1007/978-3-030-41005-6\_31. Latin American High Performance Computing Conference. 2019. CARLA 2019. 25-27 September; Turrialba, Costa Rica.

## CONCLUSÕES

Modelos e árvores de regressão e classificação foram explorados com dados de desempenho de execuções de programas de bioinformática, visando obter informações sobre a configuração que leve a alocar de maneira mais eficiente recursos computacionais para determinado experimento executado no Portal-Bioinfo. O presente trabalho abriu

vários frentes e desafios futuros dentre eles explorar e realizar uma análise comparativa de outras metodologias de AM em larga escala e o desenvolvimento de ferramentas computacionais tipo APIs que integrem esses algoritmos e alimentem os scripts de submissão na camada de execução do Bioinfo-Portal.

## REFERÊNCIAS BIBLIOGRÁFICAS

1. Gesing S, Krüger J, Grunzke R, Herres-Pawlis S, Hoffmann A. Using Science Gateways for Bridging the Differences between Research Infrastructures. *Journal of Grid Computing*. 2016;14:545–57.
2. Gesing S, Nabrzyski J, Jha S. Gateways to high-performance and distributed computing resources for global health challenges. In: 2014 IEEE Canada International Humanitarian Technology Conference - (IHTC). Montreal, QC: IEEE; 2014. p. 1–5. doi:10.1109/IHTC.2014.7147530.
3. Gomes ATA, Bastos BF, Medeiros V, Moreira VM. Experiences of the Brazilian national high-performance computing network on the rapid prototyping of science gateways: SGW2013 SPECIAL ISSUE. *Concurrency and Computation: Practice and Experience*. 2015;27:271–89.
4. Ocaña K, Galheigo M, Osthoff C, Gadelha L, Porto F, Gomes T, Oliveira D, Vasconcelos A. BioinfoPortal: A scientific gateway for integrating bioinformatics applications on the Brazilian national high-performance computing network. In Press. *Future Generation Computer Systems*, 2020. <https://doi.org/10.1016/j.future.2020.01.030>.
5. S. B. Davidson and J. Freire, “Provenance and Scientific Workflows: Challenges and Opportunities,” in ACM SIGMOD, New York, NY, USA, 2008, pp. 1345–1350.
6. I. Foster, C. Kesselman, and S. Tuecke, “The Anatomy of the Grid: Enabling Scalable Virtual Organizations,” *Lecture Notes in Computer Science*, vol. 2150, 2001.
7. M. Mattoso et al., “Towards supporting the life cycle of large scale scientific experiments,” *International Journal of Business Process Integration and Management*, vol. 5, no. 1, pp. 79–92, 2010.
8. T. M. Mitchell, Machine learning, International ed., [Reprint.]. New York, NY: McGraw-Hill, 1997.
9. P. Flach, Machine learning: the art and science of algorithms that make sense of data. Cambridge Univ. Press, 2012.
10. Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin, Learning from data: a short course. S.l.: AMLbook.com, 2012.
11. Douglas Ericson Marcelino de Oliveira, “Otimização da execução de workflows intensivos de dados em frameworks mapreduce,” Tese de Doutorado, Universidade Federal Fluminense, Niterói, RJ - Brasil, 2017.

Número do Projeto: 800657/2018-7

Processo da Bolsa: 155130/2019-6

# Segmentação de Imagens Utilizando Algoritmos Bio-Inspirados e Redes Neurais com Aplicações para Imagens Medicas e Biológicas

Aluno: Lucas Pampolin Laheras

Orientador: Gilson Antonio Giraldi ([gilson@lncc.br](mailto:gilson@lncc.br))

Co-Orientador: Paulo Sérgio Silva Rodrigues ([psergio@fei.edu.br](mailto:psergio@fei.edu.br))

Vigência: De 01/08/2019 a 31/07/2020

Tipo de Bolsa: Iniciação Científica - IC

## 1 Objetivos

No período de execução desse projeto tivemos dois objetivos:

- Criar uma plataforma de segmentação semi-automática de imagens de microscopia confocal utilizando um pipeline que utilize o algoritmo inspirado *firefly*;
- Adaptação do método de aumento de dados descrito em Peixinho *et. al* [8] para melhorar o treinamento da rede U-Net [9] para segmentação.

## 2 Introdução

Com o desenvolvimento de tecnologia digital, o armazenamento e gerenciamento de informações médicas e biológicas coletadas manualmente tem sido cada vez menos comuns, perdendo o lugar para as tecnologias relacionadas à digitalização automática, tanto em hardware quanto em software. Por conta disso, o desenvolvimento das áreas de processamento de imagens, inteligência artificial, aprendizagem profunda e ciência de dados é estratégico, particularmente para a área médica, uma vez que atualmente tanto hardware quanto software têm influenciado em diagnósticos, tratamentos e planejamento cirúrgico [2].

Por outro lado, os avanços na microscopia confocal por varredura a laser, associado ao uso de marcadores fluorescentes, têm possibilitado a geração de dados sobre dos níveis de expressão gênica e localização proteica tanto em células fixadas quanto *in vivo* [11].

Recentemente, a literatura de processamento digital de imagens e visão computacional tem demonstrado a crescente efetividade da aplicação de algoritmos bio-inspirados para segmentação de imagem [12]. Por outro lado, métodos em aprendizagem profunda, baseados em rede neuronal convolucional (*Convolutional Neural Network - CNN*) foram também utilizados para segmentação de imagens, obtendo resultados muito promissores [9]. Vale ressaltar que a área de aprendizagem profunda vem ganhado espaço em inúmeras aplicações em reconhecimento de padrões, jogos, veículos autônomos, além de outras [7].

Porém, o treinamento de uma CNN para segmentação, em geral, necessita de grandes bancos de imagens que na maioria das vezes não estão disponíveis. Para acumular dados suficientes e melhorar o desempenho de modelos de aprendizado profundo em classificação de imagens, várias estratégias de aumento de dados foram propostas [10, 8]. Estas estratégias podem ser adaptadas para o contexto de segmentação de imagens, que é o tema central nesta pesquisa.

### 3 Material e Métodos

Nessa Seção será descrito sucintamente os algoritmos que foram utilizados.

O algoritmo *firefly* [13], é uma meta-heurística inspirada pelo comportamento dos vagalumes, que são atraídos um pelo outro de acordo com sua luminescência natural. O algoritmo é utilizado para limiarização de três níveis na imagem de entrada, auxiliando na segmentação de imagens de baixo contraste.

Outra técnica utilizada no trabalho é o algoritmo level set onde o conceito básico do algoritmo é delimitar a fronteira de uma região planar utilizando o conjunto de pontos no nível zero de uma função  $\alpha : \mathbb{R}^2 \times \mathbb{R}^+ \rightarrow \mathbb{R}$  dado pela seguinte expressão:

$$S(t) = \{(x, y) \in \mathbb{R}^2 \mid \alpha(x, y, t) = 0\} \quad (1)$$

Dada uma curva de nível zero  $S(t)$  no instante  $t$ , consideram-se três possíveis valores para a função. Quando o valor é menor que 0 a coordenada  $(x, y)$  está dentro da curva  $S$ , quando o valor é maior que 0 a coordenada  $(x, y)$  está fora da curva e quando o valor for igual a 0 a coordenada  $(x, y)$  pertence a  $S$ . O próximo passo é encontrar a formulação euleriana da evolução da frente, gerando uma equação diferencial parcial que descreve a evolução da curva de nível no tempo e no espaço [6].

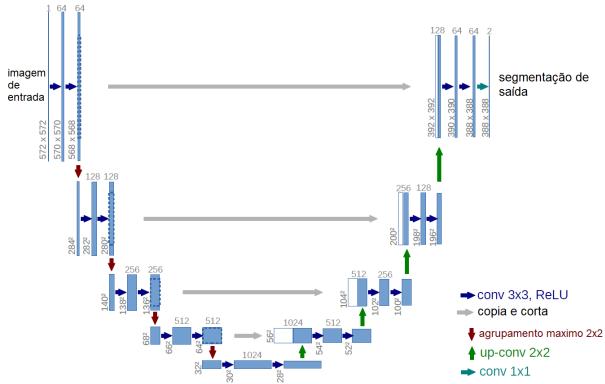


Fig. 1: Arquitetura da U-Net. Modificado de Ronneberger *et. al* [9]

Outra técnica utilizada foi a U-Net [9] que é um tipo de rede neural totalmente convolucional (*Fully Convolutional Network*) [5]. Como observado na Figura 1, a arquitetura da U-Net consiste em duas etapas: contração e expansão. Na contração é aplicada repetidamente duas vezes a convolução 3x3 seguida pela ReLU. Em seguida, é aplicado a operação *max pool* 2x2 iterando com passo 2 para redução de dimensionalidade. Em cada camada de contração é dobrado o número de mapas de características. Na expansão é aplicado uma operação de

*upsampling*, seguida por uma convolução 2x2 chamada de *up-convolution* que reduz pela metade o número de mapas de características. Em seguida ocorre a concatenação com a imagem correspondente e duas convoluções 3x3, ambas seguidas pela ReLU.

Por fim, utilizamos uma rede auto-codificadora que é um algoritmo de aprendizado de máquina não supervisionado que reproduz na saída os valores do dado de entrada. Esse método é composto por duas redes neurais (codificador e decodificador) sendo que a saída do codificador é a entada do decodificador. Após o treinamento da rede, essa saída é interpretada como um vetor de características que representa o dado de entrada. Auto-codificadores são usados para reduzir a dimensionalidade dos dados [8].

### 3.1 Metodologia

A metodologia consistiu em duas etapas: plataforma semi-automatica de Segmentação e metodologia para aumento de dados. Para a criação de plataforma semi-automática foi desenvolvido um software utilizando um módulo padrão do *Python* para interface gráfica multiplataforma chamado *TKinter* [1]. Foi implementado também um pipeline específico para esse tipo de imagem de microscopia confocal, que foi descrito em [4]. Neste caso foi utilizado um banco de imagens de células do ovário de *Drosophila melanogaster* (mosca da fruta) produzidas no Grupo de Biologia do Desenvolvimento e Sistemas Dinâmicos da Universidade Federal do Rio de Janeiro (UFRJ).

Para o aumento de dados foi utilizado um banco de imagens de ultra-sonografia de mama, composta por 250 imagens que contém tanto as imagens segmentadas quanto o padrão-ouro da classificação entre tumor maligno e benigno.

A técnica para aumento de dados foi adaptada de Peixinho *et. al* [8]. A técnica consiste em após treinar o auto-codificador, usá-lo para codificar o conjunto de imagens de entrada, transformando-as em vetores  $n$ -dimensionais e projetar cada imagem em coordenadas bidimensionais utilizando a técnica *Local Affine Multidimensional Projection* (LAMP) [3]. Dados os pontos da projeção é aplicado a triangulação de Delauney. Após são criados novos pontos usando a distribuição normal gaussiana no plano. Os novos pontos serão considerados novas imagens codificadas, no espaço  $n$ -dimensional, somente se estiverem dentro de um triângulo que tiver todos os pontos com a mesma classificação. Depois dessa seleção, todos os novos pontos serão decodificados e transformados em imagens que antes não existiam e a segmentação será gerada pelo mesmo pipeline utilizado na interface gráfica, explicada no parágrafo anterior.

## 4 Resultados

O *pipeline* de segmentação descrito na Seção 3.1 foi implementado em uma plataforma interativa (Figura 2) para que os biólogos e médicos das equipes envolvidas possam determinar o padrão ouro das imagens que farão parte de um novo banco de dados gerado pelos especialistas.

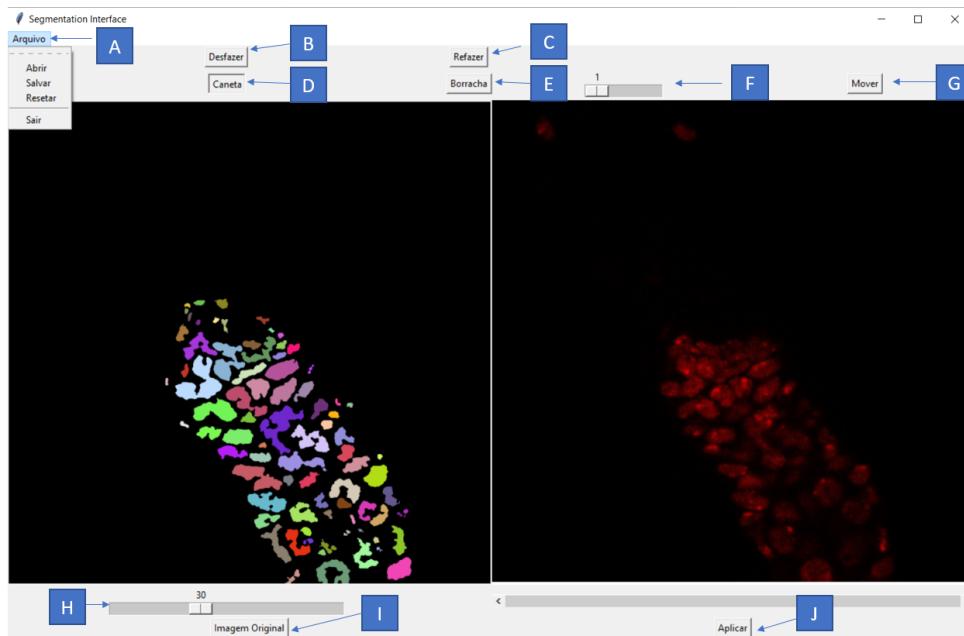
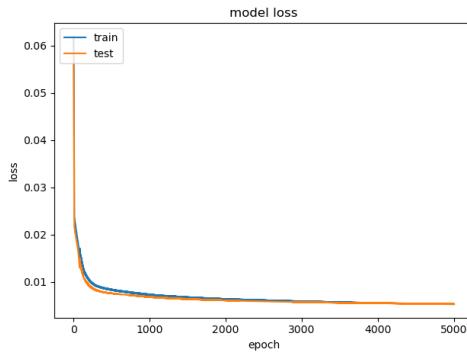


Fig. 2: Layout da interface gráfica desenvolvida.

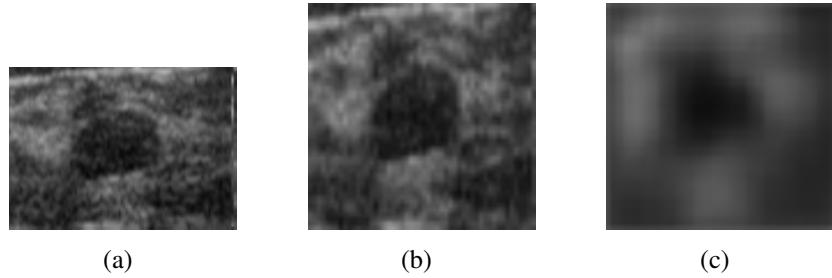
A Figura 2 mostra alguns elementos da interface, bem como o resultado obtido selecionando a imagem da direita para a segmentação. Se o resultado não for satisfatório, o usuário pode usar o mouse e recursos da interface para corrigir as imperfeições manualmente. A documentação de uso e instalação está disponível em *CellDetectionImage*<sup>1</sup> no *GitHub*. A plataforma, esta descrita em [4], e se encontra em utilização por biólogos da UFRJ para geração de um banco de imagens de microscopia confocal, com intuito de treinamento de uma rede neural para segmentação destas imagens.

Com relação a metodologia para aumento de dados, o primeiro passo foi o treinamento de uma rede auto-codificadora. Posteriormente, os resultados do codificador são projetados no espaço bidimensional. Os resultados gerados após o treinamento do auto-codificador mostraram-se satisfatórios usando a função de perda do erro quadrático médio, com erro de 00,529%. O codificador com três camadas de convolução utilizando a ativação ReLU seguidas pelo *pooling* máximo 2x2 com iteração 2 e o decodificador com a mesma configuração de convolução e se-

<sup>1</sup> CellDetectionImage: <https://github.com/LucasLaheras/CellDetectionImage>



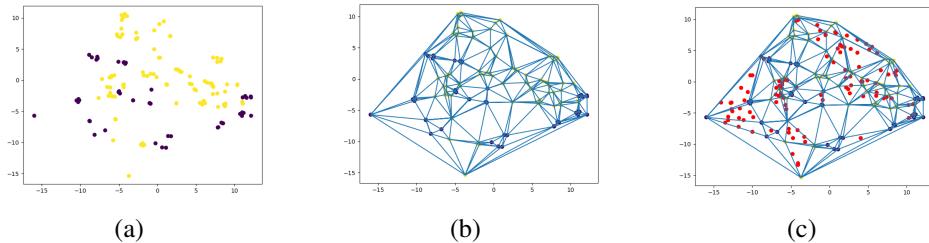
**Fig. 3:** Histórico de perda durante o treinamento do auto-codificador, onde em azul representa a perda das imagens de treinamento e em laranja as imagens de teste



**Fig. 4:** (a) Imagem original 99x71. (b) Imagem remodelada para 52x52. (c) Saída do auto-codificador.

guido pelo *up-sampling* 2x2, onde a entrada é a saída do codificador. Apesar de ser utilizada 5000 épocas para o treinamento os resultados acabaram convergindo perto da época 600, como pode-se observar na Figura 3.

A Figura 4.a mostra uma imagem original da base de dados, a qual é redimensionada para 52 x 52 pixels por conta da padronização da rede (Figura 4.b). O resultado obtido na saída do auto-codificador, para a imagem de entrada da Figura 4.b, está representada na Figura 4.c. Apesar da perda de detalhes da imagem original, observamos que o codificador gerou um vetor de características eficiente para nossa proposta.



**Fig. 5:** (a) Representação bidimensional das imagens. (b) Aplicação da triangulação de Delaunay. (c) Criação dos novos pontos em vermelho, que posteriormente serão convertidos em imagens.

Após o treinamento o auto-codificador é utilizado para a geração de novas imagens, como descrito na Seção 3.1. A Figura 5.a ilustra a representação de cada imagem no plano bidimensional após passar pelo codificador. A Figura 5.b mostra a triangulação de Delauney no plano, que é utilizada para selecionar os novos pontos de acordo com um esquema baseado na classificação das imagens originais em benignas ou malignas. Os pontos em vermelho na Figura 5.c serão usados para gerar por interpolação novos vetores de características no espaço do codificador. Em seguida, este vetores são processados pelo decodificador para gerar novas imagens. A Figura 6.a mostra uma nova imagem gerada neste processo e a Figura 6.b mostra sua segmentação via o software CellDetectionImage desenvolvido neste projeto. Apesar do efeito de suavização já comentado anteriormente, as novas imagens geradas são consideradas satisfatórias para o aumento de dados no treinamento da U-Net.

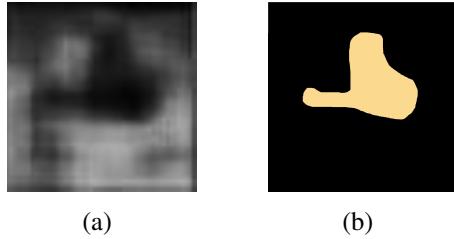


Fig. 6: (a) Imagem gerada pelo aumento de dados. (b) Segmentação gerada pelo software CellDetectionImage.

Atualmente, estamos avaliando o efeito do aumento de dados no treinamento da U-Net. Esperamos concluir esta etapa final em breve. Para estes testes foram utilizados três conjuntos de imagens para treinamento: (A) 100 imagens da base original; (B) 100 imagens da base com utilização do aumento de dados padrão, rotacionando a imagem em 90, 180 e 270 graus, totalizando 400 imagens (C) as imagens do conjunto (B) com as imagens geradas com a nova metodologia de aumento de dados. Os resultados do treinamento estão ilustrados no gráfico da Figura 7.a, 7.b e 7.c. Foi observado que a rede treinada com o conjunto (A) consegue uma acurácia de 94,82% e perda de 11,94%, o conjunto (B) teve acurácia de 95,38% e perda de 11,11%, e por fim, o conjunto (C) atingiu acurácia de 96,01% e perda de 9,54%. Apesar da semelhança entre as acuráncias, os testes preliminares mostram que o treinamento usando o conjunto (A) gerou uma rede hiper-especializada no conjunto de treinamento o que não é bom para a aplicação da rede.

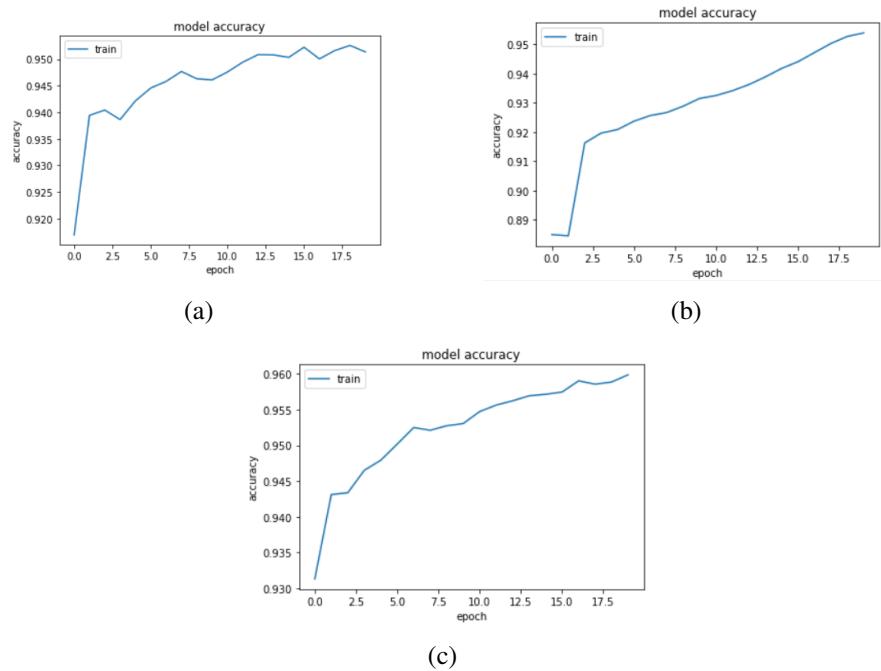


Fig. 7: Gráfico da acurácia da U-Net (a) usando o conjunto sem aumento de dados. (b) Usando o conjunto com aumento de dados padrão. (c) Usando o conjunto com aumento de dados implementado e o padrão.

## 5 Conclusão

Foi implementada uma interface gráfica semi-automática (Figura 2), descrita em [4], a qual está sendo sendo utilizada por especialistas da Universidade Federal do Rio de Janeiro (UFRJ).

O aumento de dados está sendo testado, mostrando resultados promissores. Foi observado que na base sem o aumento de dados está ocorrendo *overfitting* e ambos os métodos de aumento de dados melhoram a rede, com vantagem de 0,63% maior de acurácia usando ambos os métodos. Estamos realizando novos testes para melhorar o auto-codificador na tentativa de obter melhores resultados.

## Referências

- [1] tkinter - python interface to tcl/tk.
- [2] Kunio Doi. Computer-aided diagnosis in medical imaging: historical review, current status and future potential. *Computerized medical imaging and graphics*, 31(4-5):198–211, 2007.
- [3] Paulo Joia, Fernando Paulovich, Danilo Coimbra, Jose Cuminato, and Luis Nonato. Local affine multidimensional projection. *IEEE transactions on visualization and computer graphics*, 17:2563–71, 12 2011.
- [4] Lucas Laheras, Gilson Giraldi, Paulo Rodrigues, and Francisco Lopes. Plataforma semi-automática para segmentação de imagens biológicas com auxílio de algoritmos bio-inspirados. *SciBiz Aceito*, 2020.
- [5] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.
- [6] Ravi Malladi, James A Sethian, and Baba C Vemuri. Shape modeling with front propagation: A level set approach. 1994.
- [7] Philosophia Naturalis. 30 amazing applications of deep learning, Jun 2018.
- [8] Alan Zanoni Peixinho, Bárbara Caroline Benato, Luis Gustavo Nonato, and Alexandre Xavier Falcão. Delaunay triangulation data augmentation guided by visual analytics for deep learning. In *2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 384–391. IEEE, 2018.
- [9] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
- [10] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60, 2019.
- [11] David J Stephens and Victoria J Allan. Light microscopy techniques for live cell imaging. *science*, 300(5616):82–86, 2003.
- [12] G. A. Wachs Lopes, F. S. Beltrame, R. M. Santos, and P. S. Rodrigues. Comparison of bio-inspired algorithms from the point of view of medical image segmentation. In *2018 IEEE International Work Conference on Bioinspired Intelligence (IWobi)*, pages 1–7, July 2018.
- [13] Xin-She Yang. Firefly algorithms for multimodal optimization. In *SAGA*, 2009.

MARCOS VINÍCIOS TERNAVISK DE LIMA

## ALGORITMOS QUÂNTICOS E CRIPTOGRAFIA

Relatório parcial do projeto de Iniciação Científica do Laboratório Nacional de Computação Científica.

Orientador: Prof. Dr. Renato Portugal

Petrópolis  
2020

## SUMÁRIO

<b>1 INTRODUÇÃO . . . . .</b>	<b>3</b>
<b>2 ATIVIDADES DESENVOLVIDAS . . . . .</b>	<b>5</b>
<b>2.1 Sistemática da Orientação . . . . .</b>	<b>5</b>
<b>2.2 Objetivos . . . . .</b>	<b>5</b>
<b>2.2.1 Objetivos Alcançados . . . . .</b>	<b>5</b>
<b>2.3 Atividades Acadêmico-Científicas . . . . .</b>	<b>5</b>
<b>3 RELATÓRIO CIENTÍFICO . . . . .</b>	<b>7</b>
<b>3.1 Revisão da Literatura . . . . .</b>	<b>7</b>
<b>3.1.1 Matriz Unitária . . . . .</b>	<b>7</b>
<b>3.1.2 Produto de Kronecker . . . . .</b>	<b>8</b>
<b>3.1.3 Decomposição em Valores Singulares (<i>Singular Value Decomposition - SVD</i>) . . . . .</b>	<b>9</b>
<b>3.1.3.1 Aproximação por matriz de menor rank . . . . .</b>	<b>9</b>
<b>3.1.3.2 Matriz ortogonal mais próxima . . . . .</b>	<b>10</b>
<b>3.1.4 Produto de Kronecker mais Próximo (<i>Nearest Kronecker Product - NKP</i>) . . . . .</b>	<b>10</b>
<b>3.1.5 Produto de Kronecker na Computação Quântica . . . . .</b>	<b>11</b>
<b>3.2 Metodologia do Trabalho . . . . .</b>	<b>13</b>
<b>3.2.1 Implementação do Algoritmo NKP . . . . .</b>	<b>14</b>
<b>3.2.1.1 Método 1 . . . . .</b>	<b>14</b>
<b>3.2.1.2 Método 2 . . . . .</b>	<b>15</b>
<b>3.2.2 Implementação do Algoritmo para o cálculo da unitária mais próxima . . . . .</b>	<b>16</b>
<b>3.3 Resultados e Discussão . . . . .</b>	<b>16</b>
<b>3.4 Conclusão . . . . .</b>	<b>20</b>
<b>REFERÊNCIAS . . . . .</b>	<b>21</b>

## 1 INTRODUÇÃO

O presente trabalho tem como base o estudo da Computação Quântica. No modelo quântico de circuitos, a Computação Quântica é realizada por circuitos quânticos, que operam sobre qubits através de operações no espaço de Hilbert, também chamado de espaço de estados do sistema. O espaço de estados de um sistema quântico é descrito em termos de espaços vetoriais, que são objeto de estudo da Álgebra Linear.

A Álgebra Linear é a base matemática utilizada na Computação Quântica, permitindo entender conceitos básicos, desde como descrever os estados de qubits até operações mais complexas, pois os qubits, assim como as portas lógicas quânticas, são representados matematicamente por matrizes.

O espaço de estados de um sistema físico composto é o produto tensorial do espaço de estados de componentes individuais de sistemas físicos. Matematicamente, o produto tensorial de dois estados é equivalente ao produto de Kronecker entre seus vetores correspondentes. Contudo, é importante destacar que nem todos os estados de um sistema combinado podem ser separados no produto tensorial de componentes individuais.

Um sistema quântico que opera sobre  $n$  qubits é representado por uma matriz unitária de dimensões  $2^n \times 2^n$  e consiste em uma coleção finita de portas quânticas, que são os elementos básicos de um computador quântico. Cada porta quântica implementa uma transformação unitária sobre qubits. Portanto, pode-se dizer que um programa é uma sequência de operações unitárias aplicadas ao estado inicial. A condição de unitariedade garante que o operador não viole nenhum princípio geral da teoria quântica.

A construção de operadores unitários para implementação de novos algoritmos quânticos e a busca por uma sequência de portas quânticas de custo mínimo para a implementação destes operadores unitários é de fundamental importância para Computação Quântica, pois a quantidade de recursos disponíveis nos computadores quânticos atuais ainda é bastante limitada. Para aumentar os benefícios da computação quântica, é essencial obter uma decomposição eficiente dos algoritmos quânticos em portas universais.

Em particular, neste trabalho utilizaremos a decomposição em produto de Kronecker a fim de melhor compreendermos este problema. Através de uma implementação prática, buscamos absorver tanto ideias da Computação Quântica como desenvolver capacitação em Álgebra Linear.

Como será apresentado mais a frente, o Produto de Kronecker é um tipo especial

de produto de matrizes, ou seja, é a multiplicação de cada elemento de uma matriz por outra matriz.

## 2 ATIVIDADES DESENVOLVIDAS

### 2.1 Sistemática da Orientação

A metodologia de orientação baseou-se em estudo dirigido, resolução de exercícios, simulação e implementação de algoritmos em Python e participação nos seminários do grupo de Computação Quântica do LNCC.

### 2.2 Objetivos

O objetivo geral do trabalho foi complementar a minha formação acadêmica na área de Computação Quântica.

Os objetivos específicos buscaram o desenvolvimento dos fundamentos matemáticos dessa teoria, em especial tópicos de Álgebra Linear, o estudo das principais portas lógicas quânticas e simulações em Python.

#### 2.2.1 Objetivos Alcançados

Consegui aprimorar e aprender conceitos elementares como soma, subtração e multiplicação de matrizes; matriz transposta; matriz inversa; cálculo de determinantes; sistemas lineares e conceitos mais específicos de Álgebra Linear, tais como: espaços e subespaços vetoriais; combinações lineares; base e dimensão; produto interno, além de conjuntos ortogonais e ortonormais.

Tive dificuldades para entender os conceitos de espaços e subespaços vetoriais e a minha estratégia para superar tais dificuldades foi copiar os teoremas e definições para o caderno, fazer exercícios propostos e assistir vídeo aulas no youtube.

Em Computação Quântica, a minha estratégia para aprender alguns conceitos básicos foi a utilização do simulador quântico da IBM. Com ele, pude identificar a porta X, que faz o papel da porta Not; a porta H - Hadamard, que coloca o qubit em superposição; a porta CNOT - Controlled Not, que coloca dois qubits em estado de emaranhamento quântico; e a operação que faz a medição do qubit.

### 2.3 Atividades Acadêmico-Científicas

O trabalho foi direcionado no sentido de formar uma base sólida no tema proposto, dividido nas seguintes etapas:

- Estudo teórico de conceitos da Álgebra Linear, através das referências (RIOS; FIGUEIREDO; CUNHA, 2010; FIGUEIREDO; CUNHA, 2010; BEDOYA; CAMELIER, 2010), com a resolução de exercícios selecionados.
- Estudo introdutório da Computação Quântica baseado no capítulo 1 da referência (PORTUGAL et al., 2012) e na parte I da referência (NIELSEN; CHUANG, 2011).
- Estudo das portas e circuitos lógicos quânticos baseado no capítulo 2 da referência (PORTUGAL et al., 2012) e no capítulo 4 da referência (NIELSEN; CHUANG, 2011).
- Estudo de conceitos básicos de programação em Python, como estruturas de decisão if/elif/else, comparações múltiplas, tipos de variáveis, uso do pacote Numpy.
- Implementação em Python de um problema da Álgebra Linear, em que foram explorados vários conceitos da Computação Quântica.

Além disso, a fim de ampliar o entendimento na área da Computação Quântica, participei de seminários online, organizados pelo professor Renato Portugal, ministrados tanto por pesquisadores do LNCC quanto por pesquisadores internacionais.

Isso me ajudou a fixar o conhecimento de Álgebra Linear e Computação Quântica, pois o que foi visto na teoria pôde ser melhor compreendido na prática.

Também participei dos cursos de verão oferecidos na FAETERJ, campi Petrópolis, ministrados pelo professor Nilo Koscheck em Janeiro de 2020 sobre Eletrônica e Opentrônica Básica.

### 3 RELATÓRIO CIENTÍFICO

#### 3.1 Revisão da Literatura

Gerar um circuito a partir de uma matriz arbitrária é um problema que escala exponencialmente com o número de qubits do sistema em geral, e o problema de encontrar o menor circuito possível para um operador em particular permanece desafiador. No entanto, várias técnicas foram desenvolvidas para esse fim, usando diferentes métodos de decomposição (BARENCO et al., 1995; CYBENKO, 2001; LI; ROBERTS; YIN, 2013).

A proposta deste trabalho consiste em implementar a decomposição de matrizes unitárias através do produto de Kronecker como uma forma de abordar e compreender questões importantes da Computação Quântica.

A seguir, apresentaremos alguns dos principais conceitos utilizados ao longo do trabalho: matrizes unitárias, produto de Kronecker, decomposição SVD, produto de Kronecker mais próximo e o produto de Kronecker na Computação Quântica.

##### 3.1.1 Matriz Unitária

Para garantir o entendimento do conceito de matrizes unitárias, deve-se lembrar os conceitos de conjugado e transposição de matrizes.

O conjugado vem do estudo dos números complexos. Portanto, sendo  $z = a+bi$  um número complexo, com  $\{a, b\} \in \mathbb{R}$ , temos que o conjugado de  $z$  é denotado por  $z^* = a-bi$ . Logo, para uma matriz que possui todos os seus elementos reais, temos que a sua matriz conjugada será igual a matriz original.

**Exemplo 3.1.1.** Dada a matriz  $A = \begin{bmatrix} 0 & 2 \\ 3 & 5 \end{bmatrix}$ , a matriz conjugada será  $A^* = \begin{bmatrix} 0 & 2 \\ 3 & 5 \end{bmatrix}$ .

**Exemplo 3.1.2.** Dada a matriz  $B = \begin{bmatrix} i & 0 \\ 0 & 1 \end{bmatrix}$ , a matriz conjugada será  $B^* = \begin{bmatrix} -i & 0 \\ 0 & 1 \end{bmatrix}$ .

A matriz transposta da matriz  $A$ , de ordem  $m \times n$ , é a matriz  $A^T$ , de ordem  $n \times m$ , que se obtém escrevendo ordenadamente as linhas de  $A$  como colunas.

**Exemplo 3.1.3.** Sendo  $C = \begin{bmatrix} 2 & 3 & 4 \\ 5 & 6 & 7 \end{bmatrix}$ , temos que  $C^T = \begin{bmatrix} 2 & 5 \\ 3 & 6 \\ 4 & 7 \end{bmatrix}$

Conhecendo agora o conceito de conjugado e de transposição de matrizes, pode-se partir para a definição de matriz unitária.

**Definição 3.1.1.** Uma matriz  $U$ , com  $k$  linhas e  $k$  colunas e coeficientes complexos é unitária se o produto de  $U$  com a matriz conjugada da transposta, representada como  $U^\dagger$ , for igual à matriz identidade, ou seja,

$$UU^\dagger = U^\dagger U = I = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

**Exemplo 3.1.4.** Dada a matriz  $U = \begin{bmatrix} 0 & i \\ i & 0 \end{bmatrix}$ , temos que  $U^\dagger = \begin{bmatrix} 0 & -i \\ -i & 0 \end{bmatrix}$ , logo:

$$\begin{aligned} UU^\dagger &= \begin{bmatrix} 0 & i \\ i & 0 \end{bmatrix} \begin{bmatrix} 0 & -i \\ -i & 0 \end{bmatrix} = \begin{bmatrix} 0.0 + i.(-i) & 0.(-i) + i.0 \\ i.0 + 0.(-i) & i.(-i) + 0.0 \end{bmatrix} \therefore \\ UU^\dagger &= \begin{bmatrix} -i^2 & 0 \\ 0 & -i^2 \end{bmatrix} = \begin{bmatrix} -(-1) & 0 \\ 0 & -(-1) \end{bmatrix} \therefore \\ UU^\dagger &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I \end{aligned}$$

Como  $UU^\dagger = I$ , a matriz  $U$  é **unitária**.

### 3.1.2 Produto de Kronecker

Se  $A$  é uma matriz com  $m$  linhas e  $n$  colunas e  $B$  é uma matriz com  $p$  linhas e  $q$  colunas, então o produto de Kronecker de  $A$  e  $B$  (denotado por  $A \otimes B$ ) é uma matriz  $C$  com  $mp$  linhas e  $nq$  colunas, definido como:

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \dots & a_{1n}B \\ a_{21}B & a_{22}B & \dots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \dots & a_{mn}B \end{bmatrix}$$

**Exemplo 3.1.5.** Sejam as matrizes  $C = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$  e  $D = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ . O produto de Kronecker entre as matrizes  $C$  e  $D$  é dado por:

$$\begin{aligned}
 C \otimes D &= \begin{bmatrix} c_{11}D & c_{12}D \\ c_{21}D & c_{22}D \end{bmatrix} \\
 C \otimes D &= \begin{bmatrix} 1 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & 2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ 3 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & 4 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \end{bmatrix} \\
 C \otimes D &= \begin{bmatrix} 1 & 0 & 2 & 0 \\ 0 & 1 & 0 & 2 \\ 3 & 0 & 4 & 0 \\ 0 & 3 & 0 & 4 \end{bmatrix}
 \end{aligned}$$

### 3.1.3 Decomposição em Valores Singulares (*Singular Value Decomposition - SVD*)

A decomposição em valores singulares de uma matriz  $M_{m \times n}$ , real ou complexa, é uma fatoração da forma  $U\Sigma V^\dagger$ , onde  $U$  é uma matriz unitária  $m \times m$ , real ou complexa,  $\Sigma$  é uma matriz diagonal retangular  $m \times n$  contendo números reais não negativos na diagonal, e  $V$  é uma matriz unitária  $n \times n$ , real ou complexa.

As entradas diagonais  $\sigma_i$  de  $\Sigma$  são os valores singulares de  $M$ . Os valores singulares não-nulos de  $M$  são as raízes quadradas dos autovalores não nulos de  $M^\dagger M$  ou  $MM^\dagger$ . O número de valores singulares não nulos é igual ao rank de  $M$ . As colunas de  $U$  e  $V$  são os vetores singulares à direita (autovetores de  $MM^\dagger$ ) e vetores singulares à esquerda (autovetores de  $M^\dagger M$ ) de  $M$ , respectivamente.

A decomposição SVD apresenta inúmeras aplicações, dentre as quais destacamos a aproximação por matriz de menor rank e o cálculo da matriz ortogonal mais próxima.

#### 3.1.3.1 Aproximação por matriz de menor rank

Consiste em um problema de minimização em que se deseja aproximar a matriz  $M$  por uma matriz  $\tilde{M}$  de rank  $r$  mais baixo. No caso em que a aproximação é baseada na minimização da norma de Frobenius, a solução do problema é dada pela decomposição SVD, conforme provado pelo Teorema Eckart–Young (ECKART; YOUNG, 1936).

Tem-se  $\tilde{M} = U\tilde{\Sigma}V^\dagger$ , onde  $\tilde{\Sigma}$  diferencia-se de  $\Sigma$  por conter apenas os  $r$  maiores valores singulares.

### 3.1.3.2 Matriz ortogonal mais próxima

É possível usar a decomposição SVD de uma matriz quadrada  $A$  para determinar a matriz ortogonal  $O$  mais próxima. Tomando-se a distância  $O - A$  dada pela norma de Frobenius, a solução é dada pelo produto  $UV^\dagger$  (SHONEMANN, 1966; ZHANG, 2000).

### 3.1.4 Produto de Kronecker mais Próximo (*Nearest Kronecker Product - NKP*)

O problema do 'Produto de Kronecker mais Próximo' consiste em encontrar  $B \in \mathbb{C}^{m \times n}$  e  $C \in \mathbb{C}^{s \times t}$  que minimizem  $\phi_A(B, C) = \|A - B \otimes C\|_F$ . Van Loan e Pitsianis (PITSIANIS; LOAN, 1993) propuseram uma decomposição com base no algoritmo SVD (conforme seção 3.1.3.1) a partir de uma versão permutada de  $A$ . Descreveremos sua solução para o caso em que utilizamos a norma de Frobenius  $\|\cdot\|_F$ .

Consideremos o caso geral. Seja a matriz  $A_{ms \times nt} \in \mathbb{C}$ . Não é sempre verdade que existam matrizes  $B_{m \times n}$  e  $C_{s \times t}$  tais que  $A = B \otimes C$ , exceto se  $m = n = 1$  ou  $s = t = 1$ .

Usando o fato que

$$\|A\|_F = \sqrt{\sum_{j=1}^m \sum_{k=1}^n |(A)_{j,k}|^2}$$

é independente da ordem da soma sobre os elementos de  $A$ , tem-se

$$\|A\|_F = \|R(A)\|_F$$

onde  $R(A)$  é uma matriz  $m \times n$ , em que cada elemento de  $A$  aparece uma única vez em  $R(A)$ . Especificamente, consideramos  $R = R_{m \times n, s \times t}$  como sendo o operador de reformatação

$$R_{m \times n, s \times t}(A \otimes B) = (\text{vec}(A))(\text{vec}(B))^T$$

para todas matrizes  $A_{m \times n}$  e  $B_{s \times t}$ . Claramente,  $A \otimes B$  e  $(\text{vec}(A))(\text{vec}(B))^T$  possuem os mesmos elementos em configurações diferentes.

Sendo assim, para um exemplo em que  $A$  é uma matriz  $6 \times 4$ ,  $B$  é uma matriz  $3 \times 2$  e  $C$  é uma matriz  $2 \times 2$ , o problema pode ser reformulado da seguinte forma:

$$\begin{aligned}
\phi(A, B) &= \left\| \begin{bmatrix} a_{11} & a_{12} & | & a_{13} & a_{14} \\ a_{21} & a_{22} & | & a_{23} & a_{24} \\ \hline a_{31} & a_{32} & | & a_{33} & a_{34} \\ a_{41} & a_{42} & | & a_{43} & a_{44} \\ \hline a_{51} & a_{52} & | & a_{53} & a_{54} \\ a_{61} & a_{62} & | & a_{63} & a_{64} \end{bmatrix} - \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{bmatrix} \otimes \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \right\|_F \\
&= \left\| \begin{bmatrix} a_{11} & a_{21} & a_{12} & a_{22} \\ a_{31} & a_{41} & a_{32} & a_{42} \\ \hline a_{51} & a_{61} & a_{52} & a_{62} \\ a_{13} & a_{23} & a_{14} & a_{24} \\ \hline a_{33} & a_{43} & a_{34} & a_{44} \\ a_{53} & a_{63} & a_{54} & a_{64} \end{bmatrix} - \begin{bmatrix} b_{11} \\ b_{21} \\ b_{31} \\ b_{12} \\ b_{22} \\ b_{32} \end{bmatrix} \begin{bmatrix} c_{11} & c_{21} & c_{12} & c_{22} \end{bmatrix} \right\|_F \\
&= \left\| \tilde{A} - \text{vec}(B) \cdot \text{vec}(C)^T \right\|_F
\end{aligned}$$

Como a matriz de rank-1 mais próxima de  $\tilde{A} \in \mathbb{C}^{m_1 n_1 \times m_2 n_2}$  é dada pelo maior autovalor e respectivos autovetores, obtém-se a solução a seguir:

Dada a matriz  $A \in \mathbb{C}^{m_1 n_1 \times m_2 n_2}$ , calcular a decomposição SVD de

$$\tilde{A} = U\Sigma V^\dagger = \sum_{k=1}^{r_{KP}} \sigma_k u_k v_k^\dagger.$$

Definir  $B_{opt} \in \mathbb{C}^{m_1 \times n_1}$  como  $\text{vec}(B) = \sqrt{\sigma_1} u_1$ .

Definir  $C_{opt} \in \mathbb{C}^{m_2 \times n_2}$  como  $\text{vec}(C) = \sqrt{\sigma_1} v_1$ .

### 3.1.5 Produto de Kronecker na Computação Quântica

Atualmente, o modelo de circuitos é a abstração mais usada na Computação Quântica, representado por uma sequência de portas lógicas quânticas. A evolução de um sistema quântico normalmente requer que várias portas quânticas sejam aplicadas, sequencialmente, em paralelo ou condicionalmente, a vários subconjuntos de qubits. O produto de Kronecker é usado para descrever portas quânticas que atuam em paralelo, enquanto o produto matricial é usado para descrever portas quânticas que atuam sequencialmente, um após o outro, da esquerda para a direita, em um circuito quântico. Por fim, as portas controladas aplicam uma porta quântica a um subconjunto alvo de qubits dependendo dos valores em um conjunto de qubits de controle.

De maneira similar aos computadores eletrônicos clássicos, uma coleção de portas quânticas é usada para construir circuitos quânticos e projetar algoritmos quânticos. (BARENCO et al., 1995) mostram que a porta controlada CNOT e portas quânticas de um único qubit são universais para a Computação Quântica.

Portas quânticas de um único qubit incluem Pauli-X, Pauli-Y, Pauli-Z e Hadamard. Estas transformações unitárias de um único bit são expressas na forma de matrizes  $2 \times 2$ :

$$\begin{array}{ll} \text{---} \boxed{X} \text{---} & X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \\ \text{---} \boxed{Y} \text{---} & Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \end{array} \quad \begin{array}{ll} \text{---} \boxed{Z} \text{---} & Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \\ \text{---} \boxed{H} \text{---} & H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \end{array}$$

A porta CNOT opera sobre dois qubits e é representada por:

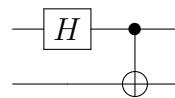
$$\begin{array}{c} \text{---} \bullet \text{---} \\ \text{---} \oplus \text{---} \end{array} \quad \text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

O produto tensorial (ou produto de Kronecker) é uma operação fundamental na construção de sistemas quânticos. Espaços vetoriais associados aos sistemas quânticos combinam-se através do produto tensorial. No entanto, deve-se observar que no mundo quântico há muito mais estados possíveis do que apenas estados que podem ser combinados a partir de estados menores. Quando um estado não pode ser escrito como um produto tensorial de dois subestados, diz-se que está *emaranhado*. Se dois estados não estão emaranhados, dizemos que são um estado de produto ou *separável*. Estados quânticos também podem estar parcialmente emaranhados, o que implica que não estão emaranhados ao máximo, mas possuem alguma correlação. O poder da Computação Quântica é mais frequentemente atribuído ao emaranhamento (JOZSA; LINDEN, 2003).

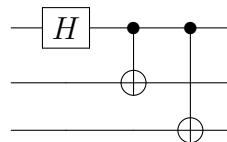
Os estados de Bell são exemplos de estados maximamente emaranhados com 2 qubits:

$$\begin{aligned} |\beta_{00}\rangle &= \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \\ |\beta_{01}\rangle &= \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \\ |\beta_{10}\rangle &= \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \\ |\beta_{11}\rangle &= \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) \end{aligned}$$

Os estados de Bell são representados pelo circuito:



A generalização dos estados de Bell para 3 qubits são os estados GHZ, representados pelo circuito:



Enquanto as portas de Hadamard são responsáveis por gerar o paralelismo quântico (gerando estados com coeficientes não-nulos), as portas CNOT são responsáveis por gerar o emaranhamento. Embora portas CNOT sejam usados na maioria esmagadora dos trabalhos envolvendo circuitos quânticos, suas implementações são ordens de magnitude mais suscetíveis a erros do que as implementações de portas de um único qubit.

Uma grande parte do trabalho com circuitos quânticos consiste na decomposição de operadores unitários em uma série de operações de um único qubit e portas CNOT. Geralmente, o objetivo é encontrar o circuito quântico mínimo capaz de resolver um determinado problema. Neste trabalho, estudaremos a decomposição em produtos de Kronecker.

### 3.2 Metodologia do Trabalho

Descreveremos a seguir a implementação do algoritmo para a decomposição de matrizes unitárias em produtos de Kronecker. Para isso, utilizaremos os conceitos apresentados na seção anterior.

Considerando sistemas quânticos que operam sobre  $n$  qubits, trabalharemos com matrizes unitárias de dimensões  $2^n \times 2^n$ , que serão decompostas no produto de Kronecker de matrizes  $2 \times 2$  e podem ser representadas como portas de 1 qubit em um circuito quântico.

### 3.2.1 Implementação do Algoritmo NKP

A aproximação do produto de Kronecker de matrizes proposto por Van Loan e Pitsianis (PITSIANIS; LOAN, 1993) fatora uma matriz como um produto de Kronecker de duas matrizes. Neste caso, para que uma matriz possa ser decomposta em um número arbitrário de fatores, é preciso reaplicar o algoritmo recursivamente nas matrizes resultantes tantas vezes quanto for preciso.

#### 3.2.1.1 Método 1

Aplicamos o método de Van Loan para aproximar uma matriz  $M$  pelo produto de Kronecker entre uma matriz  $2 \times 2$  e uma matriz  $n \times n$ .

Consideremos como exemplo uma matriz  $A_{8 \times 8}$ , dada por:

$$A = \left[ \begin{array}{cccc|cccc} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} & a_{17} & a_{18} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} & a_{27} & a_{28} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} & a_{37} & a_{38} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} & a_{47} & a_{48} \\ \hline a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & a_{56} & a_{57} & a_{58} \\ a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & a_{66} & a_{67} & a_{68} \\ a_{71} & a_{72} & a_{73} & a_{74} & a_{75} & a_{76} & a_{77} & a_{78} \\ a_{81} & a_{82} & a_{83} & a_{84} & a_{85} & a_{86} & a_{87} & a_{88} \end{array} \right] = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

Queremos minimizar  $\phi(A, B)$ , tal que

$$\begin{aligned} \phi(A, B) &= \left\| A - \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \otimes \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & c_{34} \\ c_{41} & c_{42} & c_{43} & c_{44} \end{bmatrix} \right\|_F \\ &= \left\| \tilde{A} - \text{vec}(B) \cdot \text{vec}(C)^T \right\|_F \end{aligned}$$

Para isso, as matrizes  $\tilde{A}$ ,  $\text{vec}(B)$  e  $\text{vec}(C)^T$  tomam a forma

$$\tilde{A} = \begin{bmatrix} a_{11} & a_{21} & a_{31} & a_{41} & a_{12} & a_{22} & a_{32} & a_{42} & a_{13} & a_{23} & a_{33} & a_{43} & a_{14} & a_{24} & a_{32} & a_{44} \\ \hline a_{51} & a_{61} & a_{71} & a_{81} & a_{52} & a_{62} & a_{72} & a_{82} & a_{53} & a_{63} & a_{73} & a_{83} & a_{54} & a_{64} & a_{74} & a_{84} \\ a_{15} & a_{25} & a_{35} & a_{45} & a_{16} & a_{26} & a_{36} & a_{46} & a_{17} & a_{27} & a_{37} & a_{47} & a_{18} & a_{28} & a_{38} & a_{48} \\ \hline a_{55} & a_{65} & a_{75} & a_{85} & a_{56} & a_{66} & a_{76} & a_{86} & a_{57} & a_{67} & a_{77} & a_{87} & a_{58} & a_{68} & a_{78} & a_{88} \end{bmatrix}$$

$$\tilde{A} = \begin{bmatrix} \text{vec}(A_{11})^T \\ \text{vec}(A_{21})^T \\ \text{vec}(A_{12})^T \\ \text{vec}(A_{22})^T \end{bmatrix}, \quad \text{vec}(B) = \begin{bmatrix} b_{11} \\ b_{21} \\ b_{12} \\ b_{22} \end{bmatrix},$$

$$\text{vec}(C)^T = \begin{bmatrix} c_{11} & c_{21} & c_{31} & c_{41} & c_{12} & c_{22} & c_{32} & c_{42} & c_{13} & c_{23} & c_{33} & c_{43} & c_{14} & c_{24} & c_{34} & c_{44} \end{bmatrix}$$

### 3.2.1.2 Método 2

Aplicamos o método de Van Loan para aproximar uma matriz  $M$  pelo produto de Kronecker entre uma matriz  $n \times n$  e uma matriz  $2 \times 2$ .

Consideremos como exemplo uma matriz  $A_{8 \times 8}$ , dada por:

$$A = \left[ \begin{array}{cc|cc|cc|cc} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} & a_{17} & a_{18} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} & a_{27} & a_{28} \\ \hline a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} & a_{37} & a_{38} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} & a_{47} & a_{48} \\ \hline a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & a_{56} & a_{57} & a_{58} \\ a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & a_{66} & a_{67} & a_{68} \\ \hline a_{71} & a_{72} & a_{73} & a_{74} & a_{75} & a_{76} & a_{77} & a_{78} \\ a_{81} & a_{82} & a_{83} & a_{84} & a_{85} & a_{86} & a_{87} & a_{88} \end{array} \right] = \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{bmatrix}$$

Queremos minimizar  $\phi(A, B)$ , tal que

$$\begin{aligned} \phi(A, B) &= \left\| A - \left[ \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \\ b_{41} & b_{42} & b_{43} & b_{44} \end{bmatrix} \otimes \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \right] \right\|_F \\ &= \left\| \tilde{A} - \text{vec}(B) \cdot \text{vec}(C)^T \right\|_F \end{aligned}$$

Para isso, as matrizes  $\tilde{A}$ ,  $\text{vec}(B)$  e  $\text{vec}(C)^T$  tomam a forma

$$\tilde{A} = \begin{bmatrix} a_{11} & a_{21} & a_{12} & a_{22} \\ a_{31} & a_{41} & a_{32} & a_{42} \\ a_{51} & a_{61} & a_{52} & a_{62} \\ a_{71} & a_{81} & a_{72} & a_{82} \\ a_{13} & a_{23} & a_{14} & a_{24} \\ a_{33} & a_{43} & a_{34} & a_{44} \\ a_{53} & a_{63} & a_{54} & a_{64} \\ a_{73} & a_{83} & a_{74} & a_{84} \\ a_{15} & a_{25} & a_{16} & a_{26} \\ a_{35} & a_{45} & a_{36} & a_{46} \\ a_{55} & a_{65} & a_{56} & a_{66} \\ a_{75} & a_{85} & a_{76} & a_{86} \\ a_{17} & a_{27} & a_{18} & a_{28} \\ a_{37} & a_{47} & a_{38} & a_{48} \\ a_{57} & a_{67} & a_{58} & a_{68} \\ a_{77} & a_{87} & a_{78} & a_{88} \end{bmatrix} = \begin{bmatrix} \text{vec}(A_{11})^T \\ \text{vec}(A_{21})^T \\ \text{vec}(A_{31})^T \\ \text{vec}(A_{41})^T \\ \text{vec}(A_{12})^T \\ \text{vec}(A_{22})^T \\ \text{vec}(A_{32})^T \\ \text{vec}(A_{42})^T \\ \text{vec}(A_{13})^T \\ \text{vec}(A_{23})^T \\ \text{vec}(A_{33})^T \\ \text{vec}(A_{43})^T \\ \text{vec}(A_{41})^T \\ \text{vec}(A_{42})^T \\ \text{vec}(A_{43})^T \\ \text{vec}(A_{44})^T \end{bmatrix}, \quad \text{vec}(B) = \begin{bmatrix} b_{11} \\ b_{21} \\ b_{31} \\ b_{41} \\ b_{12} \\ b_{22} \\ b_{32} \\ b_{42} \\ b_{13} \\ b_{23} \\ b_{33} \\ b_{43} \\ b_{14} \\ b_{24} \\ b_{34} \\ b_{44} \end{bmatrix},$$

$$\text{vec}(C)^T = \begin{bmatrix} c_{11} & c_{21} & c_{12} & c_{22} \end{bmatrix}$$

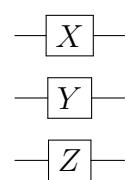
### 3.2.2 Implementação do Algoritmo para o cálculo da unitária mais próxima

A aplicação do algoritmo NKP sobre uma matriz unitária não garante que as matrizes resultantes da decomposição sejam unitárias. Como desejamos trabalhar apenas com matrizes unitárias, calculamos as unitárias mais próximas após aplicação do NKP. Conforme mencionado na seção 3.1.3.2, a solução é dada pelo produto  $UV^\dagger$  obtido a partir da decomposição SVD.

## 3.3 Resultados e Discussão

Apresentaremos, a seguir, os resultados de algumas simulações executadas. A fim de testarmos o algoritmo, partimos de soluções conhecidas, utilizando os métodos 1 e 2 descritos na seção 3.2.1. Vale mencionar, ainda, que os resultados aqui exibidos foram aproximados a fim de tornar sua visualização mais clara.

- Simulação 1:  $X \otimes Y \otimes Z$



$$U_{XYZ} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & -i & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & i \\ 0 & 0 & 0 & 0 & i & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -i & 0 \\ 0 & 0 & -i & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & i & 0 & 0 & 0 & 0 \\ i & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -i & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Decomposição método 1: Norma de Frobenius  $1.1102230246251565.e^{-15}$

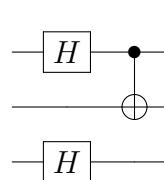
$$U_{XYZ} = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \otimes \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

Decomposição método 2: Norma de Frobenius  $8.906283103613227.e^{-16}$

$$U_{XYZ} = \begin{bmatrix} 0 & i \\ i & 0 \end{bmatrix} \otimes \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \otimes \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

Partimos, inicialmente, de uma matriz unitária que representa o produto  $X \otimes Y \otimes Z$  de um estado separável, conforme mencionado na seção 3.1.5. Observando os resultados obtidos pela norma de Frobenius, verificamos que o algoritmo aproxima-se de soluções exatas, embora diferentes para as simulações usando os métodos 1 e 2. De fato, a decomposição em produto de Kronecker não é única.

- Simulação 2:  $(CNOT \otimes I)(H \otimes I \otimes H) = (CNOT(H \otimes I)) \otimes H$



$$U_{HCNOTH} = \frac{1}{2} \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 & 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & 1 & 0 & 0 & -1 & -1 \\ 0 & 0 & 1 & -1 & 0 & 0 & -1 & 1 \\ 1 & 1 & 0 & 0 & -1 & -1 & 0 & 0 \\ 1 & -1 & 0 & 0 & -1 & 1 & 0 & 0 \end{bmatrix}$$

Decomposição método 1: Norma de Frobenius  $2.82842712474619$

$$U_{S_2} = \frac{1}{\sqrt{2}} \begin{bmatrix} -1 & -1 \\ -1 & 1 \end{bmatrix} \otimes \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} \otimes \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$U_{S_2} = \frac{1}{2} \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 & 0 & 0 & 1 & -1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & -1 & -1 \\ 0 & 0 & 1 & -1 & 0 & 0 & -1 & 1 \\ 1 & 1 & 0 & 0 & -1 & -1 & 0 & 0 \\ 1 & -1 & 0 & 0 & -1 & 1 & 0 & 0 \end{bmatrix}$$

Decomposição método 2: Norma de Frobenius 2.82842712474619

$$U_{S_2} = \frac{1}{\sqrt{2}} \begin{bmatrix} -1 & -1 \\ -1 & 1 \end{bmatrix} \otimes \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \otimes \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$$

$$U_{S_2} = \frac{1}{2} \begin{bmatrix} 0 & 0 & -1 & -1 & 0 & 0 & -1 & -1 \\ 0 & 0 & 1 & -1 & 0 & 0 & 1 & -1 \\ -1 & -1 & 0 & 0 & -1 & -1 & 0 & 0 \\ 1 & -1 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 & 0 & 0 & -1 & 1 \\ -1 & -1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 & -1 & 1 & 0 & 0 \end{bmatrix}$$

Neste caso, partimos de uma matriz unitária que gera um estado parcialmente separável. Podemos notar que, na decomposição pelo método 1, obtém-se a porta  $H$  no terceiro qubit, como no circuito original. No entanto, observamos que a norma de Frobenius para ambos os métodos apresenta valor próximo de 2.8, indicando que as fatorações obtidas pelo algoritmo possuem um erro grande e, portanto, não podem ser consideradas.

- Simulação 3: GHZ

$$U_{GHZ} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Decomposição método 1: Norma de Frobenius 2.8284271247461903

$$U_{S_3} = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} \otimes \frac{1}{\sqrt{2}} \begin{bmatrix} -1 & -1 \\ 1 & -1 \end{bmatrix}$$

$$U_{S_3} = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Decomposição método 2: Norma de Frobenius 2.8284271247461903

$$U_{S_3} = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \otimes \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$$

$$U_{S_3} = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Por fim, realizamos uma simulação com a unitária que gera os estados maxima-mente emaranhados GHZ, conforme visto na seção 3.1.5. Novamente, observamos que a norma de Frobenius resulta no valor aproximado de 2.8 para os dois métodos, indicando que o erro é suficientemente grande a ponto de inviabilizar a decomposição em produto

de Kronecker, como era de se esperar. Observamos apenas que não se percebe diferença significativa nos resultados obtidos nas simulações 2 e 3.

### 3.4 Conclusão

Aprendemos que podemos usar o produto tensorial para construir sistemas quânticos complexos a partir de sistemas mais simples de qubit único. No entanto, estas transformações não têm tanto interesse como as transformações de múltiplos qubits, pois são equivalentes a executar as transformações de qubit único em cada um dos qubits separadamente, em uma determinada ordem. Mais interessantes são as transformações de múltiplos qubits que podem gerar o emaranhamento entre os qubits do sistema. Estados emaranhados são um ingrediente crítico da Computação Quântica.

Além disso, para aprimorar os benefícios da Computação Quântica, é importante usar uma decomposição eficiente de algoritmos quânticos em portas universais, buscando o circuito quântico mínimo capaz de resolver um determinado problema. Portanto, é de fundamental importância desenvolver novos métodos com o objetivo de superar a dificuldade de decompor essa matriz nas portas quânticas conhecidas. Nesse sentido, esse estudo é uma forma de melhor compreender o problema a fim de buscar novas soluções possíveis.

Das três simulações realizadas, podemos observar que quando a norma de Frobenius estiver próxima de zero, pode-se dizer que a matriz é fatorável e que o grau de emaranhamento é fraco, como na simulação 1. Quando a norma de Frobenius estiver distante de zero, não se pode tirar conclusões a partir da mesma, pois as simulações 2 e 3 possuem praticamente o mesmo valor dessa norma e enquanto a simulação 2 é parcialmente fatorável, a simulação 3 não pode ser fatorada.

A minha Iniciação Científica foi útil, pois eu me deparei com diversos obstáculos e dificuldades que aos poucos eu fui superando. Esses obstáculos me ajudaram no crescimento pessoal e profissional, pois os conceitos apresentados aqui não estão contidos em um único livro. Tive que fazer consultas a livros, artigos científicos, sites especializados e videos no Youtube.

No mais, as disciplinas de Álgebra Linear e Programação, envolvendo conceitos de Python e Numpy, me ajudaram a compreender melhor a Computação Quântica, pois o resultado desse trabalho só foi possível graças à sinergia entre elas.

Tive a oportunidade de acompanhar diversos seminários sobre Computação Quântica, realizar simulações no computador quântico da IBM e a manejar o Numpy para trabalhar com matrizes. Com isso, pude ampliar bastante os meus horizontes e conhecimentos sobre Computação Quântica, tendo em vista também que a matemática utilizada neste trabalho é altamente especializada.

A Computação Quântica se apresenta como a sucessora da Computação Clássica. Com ela, poderemos realizar operações complexas em um tempo bem reduzido se comparado à Computação Clássica. Ainda não se sabe ao certo qual o potencial da Computação Quântica, podendo até mesmo resolver problemas que a humanidade sequer tenha imaginado.

## REFERÊNCIAS

- BARENCO, A. et al. Elementary gates for quantum computation. *Physical Review A*, v. 52, p. 3457, 1995.
- BEDOYA, H.; CAMELIER, R. *Álgebra Linear II*. 2. ed. Rio de Janeiro: Fundação CECIERJ, 2010. ISBN 85-7648-021-2/85-7648-055-7.
- CYBENKO, G. Reducing quantum computations to elementary unitary operations. *Computing in Science Engineering*, v. 3, p. 27–32, 2001.
- ECKART, C.; YOUNG, G. The approximation of one matrix by another of lower rank. *Psychometrika*, v. 1, p. 211–218, 1936.
- FIGUEIREDO, L. M.; CUNHA, M. O. da. *Álgebra Linear I*. 2. ed. Rio de Janeiro: Fundação CECIERJ, 2010. ISBN 85-7648-315-7.
- GOLUB, G. H.; LOAN, C. F. van. *Matrix Computations*. 4th. ed. JHU Press, 2013. ISBN 1421407949 9781421407944. Disponível em: <<http://www.cs.cornell.edu/cv/GVL4/golubandvanloan.htm>>.
- JOZSA, R.; LINDEN, N. On the role of entanglement in quantum-computational speed-up. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, v. 459, n. 2036, p. 2011–2032, 2003.
- LI, C.-K.; ROBERTS, R.; YIN, X. Decomposition of unitary matrices and quantum gates. *International Journal of Quantum Information*, v. 11, n. 1, 2013.
- NIELSEN, M. A.; CHUANG, I. L. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. 10th. ed. USA: Cambridge University Press, 2011. ISBN 1107002176.
- PITSIANIS, N.; LOAN, C. V. Approximations with kronecker products. Springer, p. 293–314, 1993.
- PORTUGAL, R. et al. *Uma Introdução à Computação Quântica*. 2. ed. São Carlos/SP: SBMAC, 2012. ISBN 978-85-86883-61-3.

- RIOS, I. L.; FIGUEIREDO, L. M.; CUNHA, M. O. da. *Álgebra Linear I*. 3. ed. Rio de Janeiro: Fundação CECIERJ, 2010. ISBN 85-89200-44-2.
- SHONEMANN, P. H. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, v. 31, p. 1–10, 1966.
- ZHANG, Z. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.*, v. 22, p. 1330–1334, 2000.

Título do projeto: A interação da cevada com os nutrientes do solo e o pulgão afídeo.

Nome do bolsista: Priscila Luana Lopes de Barros Weisz

Nome do orientador Maurício Vieira Kritz

Nome do Co orientador: Lucas dos Anjos Tipo de Bolsa:

Iniciação Científica Período do relatório: Fevereiro a julho de 2020 Data: 27/07/2020

---

## **RELATÓRIO SEMESTRAL LNCC**

### **A INTERAÇÃO DA CEVADA COM OS NUTRIENTES DO SOLO E COM O AFÍDEO**

#### **1. Introdução**

A cevada é um cereal que pertence a família Poaceae, é Utilizada para fabricação de alimentos, ração para animais, e muito utilizada na indústria de cerveja, a espécie estudada neste trabalho, será *Hordeum vulgare L.*

Os afídeos conhecidos como pulgões, pertencem à família Aphidae, esses insetos causam danos diretos às culturas de cereais, pela succção da seiva elaborada da planta e em alguns casos, através injeção de toxinas presente em sua saliva (TOMANOVIC et al., 2008).

Neste trabalho, estudamos o pulgão-da-espiga-do trigo, *Sitobion avenae*;

Para este trabalho, estudamos o artigo Plant genotype mediates the effects of nutrients on aphids. Este artigo foi elaborado pelos autores Rowntree J; Vennon. A; Preziosi R.(2010).

Com base nessas informações realizamos experimentos numéricos computacionais utilizando modelos simples de predação de Lotka-Volterra. Com base nos resultados dos experimentos realizados, o método de Lotka-Volterra, sofreu alteração, pois em simulações computacionais, percebemos que ele era inadequado. Desta forma, programamos um modelo adequado para este trabalho.

#### **2. Objetivo**

O presente trabalho de iniciação científica tem como objetivo, desenvolver mapa conceitual a partir do artigo estudado, para facilitar o entendimento do mesmo. Posteriormente, desenvolver modelos matemáticos com base no modelo de predação de Lotka-Volterra com resposta funcional do tipo II (COSTA e GODOY, 2010) e com base nos resultados, propor uma alteração ao modelo tradicional criando uma função para cevada e uma mortalidade quadrática para o afídeo.

### 3. Materiais e Métodos

Inicialmente, desenvolvemos um mapa conceitual, para melhor compreender as informações contidas no trabalho de ROWNTREE et al. (2010) Utilizamos como base na elaboração do nosso trabalho.

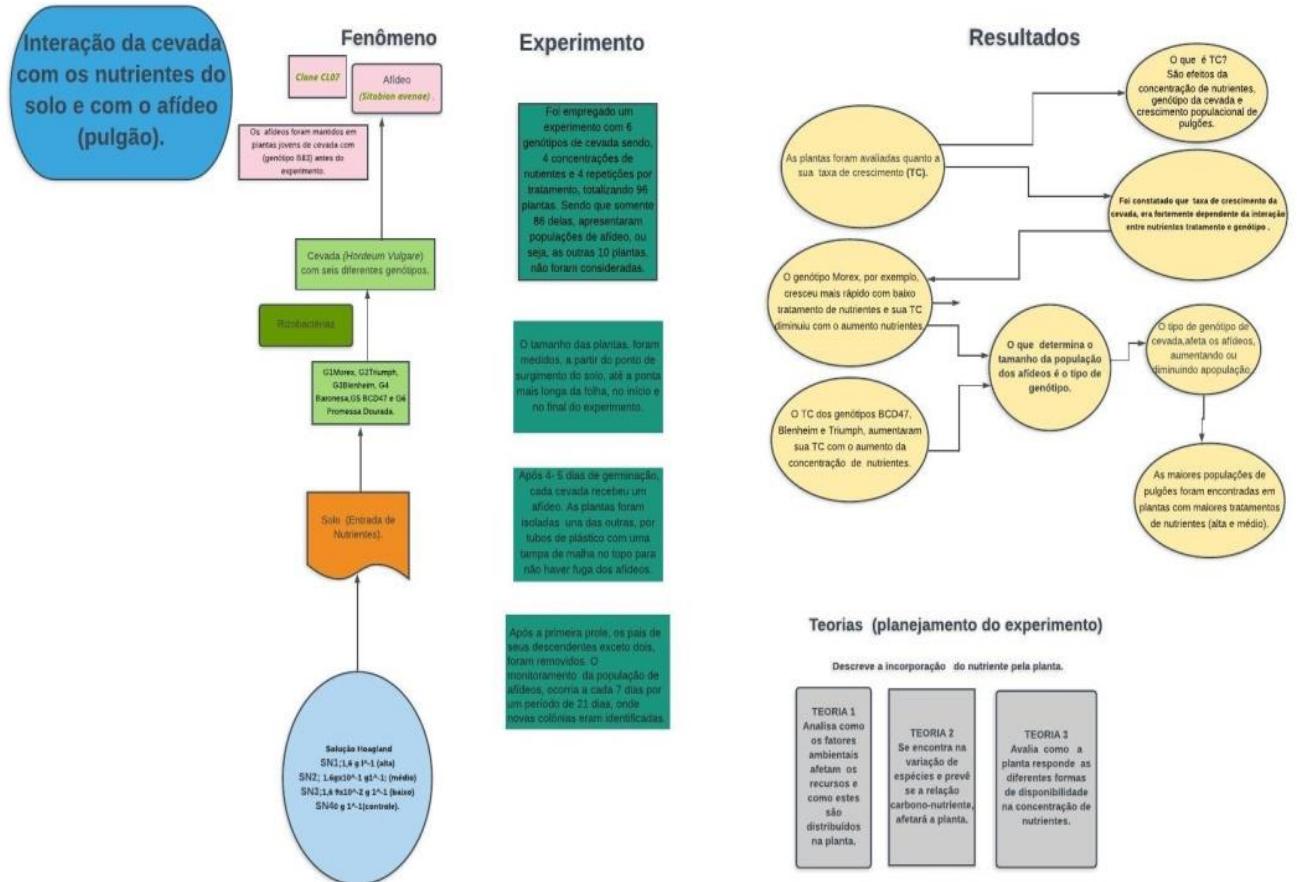


Figura 1: Mapa conceitual. Uma ferramenta para dar suporte a fim de facilitar o entendimento e auxiliar na construção dos modelos computacionais.

Após a construção do mapa conceitual, desenvolvemos modelos computacionais, baseados no método de predação de Lotka-Volterra com resposta funcional do tipo II e crescimento logístico.

O modelo de predação de Lotka-Volterra representa interação entre duas espécies, onde uma delas (presa), dispõe de alimentos em abundância e a outra espécie (predador), tem como suprimento alimentar a população de presa (FALCAO et al., 2015), a resposta funcional do tipo II, Considera o tempo que os consumidores (afídeos), gastam para processar o alimento.

O modelo logístico, utilizado, apresenta aspecto mais realista que limita o crescimento de uma determinada população (BIOMATH, 2006). Uma forma de incorporar a desaceleração de uma população é incluir a capacidade de suporte ( $K$ ), a partir da seguinte equação (COSTA e GODOY, 2010).

#### 4. Resultados e discussão

Após as etapas desenvolvidas acima, iniciamos os testes com modelo tradicional de predação de Lotka- Volterra e alcançamos os seguintes resultados:

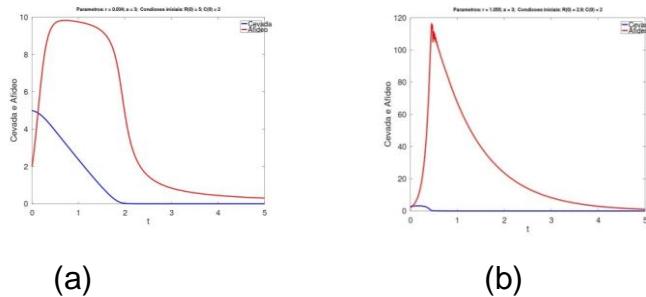


Figura 3: séries temporais que apresentam o crescimento negativo da cevada. Modelo de Lotka-Volterra. Azul é a cevada, vermelho é o afídeo. Valores dos parâmetros Para figura (a)  $Th=3.0$ ;  $K=8.0$ ;  $m=1.05$ ;  $g= 32$ . Valores de parâmetros para figura (b):  $Th=3.0$ ;  $K=100$ ;  $m=1.05$ ;  $g=32$ .

Com base nos resultados, concluímos que o modelo tradicional de Lotka-Volterra não foi adequado para representar a interação entre cevada e o afídeo.

Na figura 4 é possível perceber uma função atribuída para a cevada, que melhor representa o modelo com crescimento negativo.

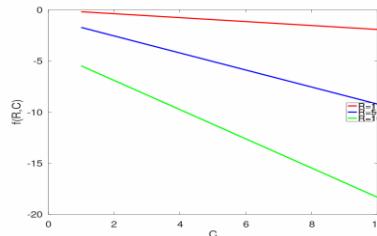


Figura 4: Série temporal representa o momento em que a taxa efetiva da população da cevada (mm), decresce alcançando um valor abaixo de 0. Exemplos de R (população de cevada) **R1=1**; **R2=5**; **R3=10**. Valores dos parâmetros:  $r=1$ ;  $a=1$ ;  $K=1$ ;  $Th=1$ .

A partir da função atribuída à cevada representada na figura 4, foi possível realizar uma alteração, deixando a figura 5 com o crescimento da cevada positivo.

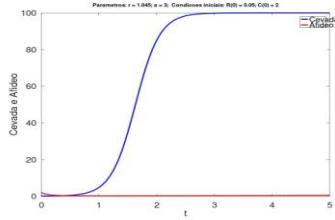
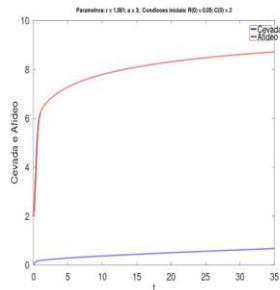
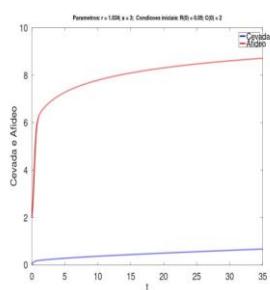


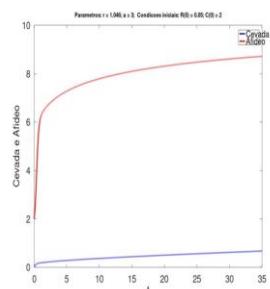
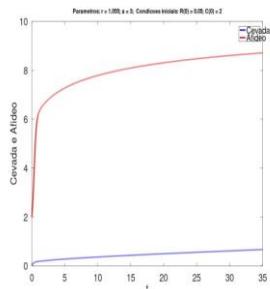
Figura 5: Série temporal representa o momento em que a população da cevada (cm), cresce e alcança a capacidade de suporte, enquanto a população de afídeo (número de indivíduos) diminui, a resposta funcional do tipo II foi retirada da equação. Além disso, a taxa de crescimento da cevada que normalmente é fixa, foi substituída por uma função  $r(C)$ , que coloca a taxa de crescimento da cevada dependente do tamanho da população de afídeo. Valores dos parâmetros:  $Th=3.0$ ;  $K=100$ ;  $m=10.4$ ;  $g=32$ ;  $Tmax=5$ .

A partir dos resultados acima, referentes à figura 6, removemos a resposta funcional do tipo II e adicionamos uma mortalidade quadrática para o afídeo, uma mortalidade com dependência de densidade, resultando na coexistência entre recurso e consumidor.



(a) Genótipo Morex  $r = 1.034$  com 0.00 nutrientes.

(b) Genótipo Morex  $r = 1.061$  com 0.01 nutrientes.



(c) Genótipo Morex  $r = 1.055$  com 0.10 nutrientes.

(d) Genótipo Morex  $r = 1.046$  com 1.00 Com 1.00 nutrientes

Figura 7: Série temporal que representam os modelos modificados sem resposta funcional do tipo II e com mortalidade quadrática para o afídeo. É importante destacar que o Genótipo Morex, aumentou sua velocidade de crescimento e baixas concentrações e teve sua velocidade diminuída em altas concentrações de nutrientes. Valores dos parâmetros:  $Th=3.0$ ;  $K=100$ ;  $m=1.05$ ;  $g=32$ .

## **6. Conclusão**

Com base em todas as etapas do trabalho realizadas, é importante salientar que o método tradicional de Lotka-Volterra, nos trouxe uma base para desenvolver modelos alternativos para representar situações que envolvem o recurso e consumidor, demonstrando suas interações entre a planta e o parasita, permitindo que houvesse entre eles, uma coexistência.

## **8. Referências:**

- ASTLES Te'tard-Jones.(ROWNTREE J;VENNON. A; PREZIOSI R.2010). PLANT-ANIMAL INTERACTIONS - ORIGINAL PAPER Oecologia (2010). Plant genotype mediates the effects of nutrients on aphids. Disponível em DOI 10.1007/s00442-010-1609-1; acesso em 15/07/2019.
- COSTA, Michel Iskin Da S.; GODOY, Wesley Augusto Conde. Fundamentos de Ecologia Teórica. Editora Manole Ltda, 2010.
- FALCAO, Danilo - RA: 154171 (2015). MENOR, Jorge-RA: 154372 (2015). MARCOLIN, Raimundo – RA 154170 (2015). Modelo Presa Predador Lotka – Volterra: Mestrado Profissional em Matemática Aplicada e Computacional: Disciplina: Modelos e Métodos Matemático. Professor: Ricardo M. Martins Universidade Estadual de Campinas –UNICAMP. Maio de 2015. Disponível em <http://www.ime.unicamp.br/~rmiranda/wordpress/wp-content/uploads/2015/12/PresaPred.pdf> acesso em 08 de maio de 2020.
- TOEBE, Josue. (2014).Um modelo baseado em agentes para o ciclo de vida de insetos: Aplicação na interação afídeo-planta- vírus. Programa de pós-graduação em agronomia. Universidade de passo Fundo. Faculdade de Agronomia e medicina veterinária. Passo Fundo- RS. 26 de fevereiro de 2014. Disponível em [http://tede.upf.br/jspui/bitstream/tede/433/1/2014\\_JosueToebe.pdf](http://tede.upf.br/jspui/bitstream/tede/433/1/2014_JosueToebe.pdf).
- PEREIRA, Paulo. LAU, Douglas. JÚNIOR, Alberto. PANIZZI, Antonio. Fitossanidade. Pragas iniciais que comprometem até o fim. Revista A Granja. Editora Centaurus. Porto Alegre – RS. Edição 763. Publicada em julho de 2012. Disponível em <https://edcentaurus.com.br/agranja/edicao/763/materia/4481>. Acesso em 15/07/2019.



## Avaliação de Métodos de Aprendizado em Aplicações de Saúde

Nome do Bolsista: Raquel de Abreu Junqueira Gritz

Nome do Orientador: Fábio André Machado Porto

Nome do Coorientador: Douglas Ericson Marcelino de Oliveira

Tipo de Bolsa e Período do Relatório: Bolsa de Iniciação Científica – IC/PIBIC – Julho de 2020

### 1. Objetivos

Este trabalho de iniciação científica tem como objetivo utilizar técnicas de pré-processamento de dados de saúde que influenciarão no aprendizado de máquina sendo úteis para predizer com qualidade, visto que tomadas de decisão no âmbito da saúde não podem conter falhas.

Tendo em vista a busca por uma predição de qualidade, busca-se também comparar as técnicas de aprendizado em modelos de *Decision Tree* e *Random Forest* aplicadas a problemas ligados à predição, em particular considerando dados de saúde relacionados à causas de óbito que podem ser evitáveis por ações de saúde, no contexto da cooperação LNCC-DEXL/FIOCRUZ-ICICT, e identificar características dos métodos que melhor se adéquem ao problema.

### 2. Introdução

A busca pela capacidade de analisar um grande volume de dados por meio de métodos estatísticos e o uso de uma variedade de algoritmos para encontrar padrões nos dados, mostra que, com base nestes padrões, é possível realizar predições.

A saúde, por exemplo, é uma área onde predições são muito bem vindas. Atualmente, um grande volume de dados se encontra disponível e pode ser usado em um processo de predição. Neste contexto, o artigo [1] foi utilizado como base para este trabalho buscando identificar, no período de 1996 a 2014, nos estados brasileiros, a ocorrência de óbitos relacionados à doenças que podem ser preveníveis por ações de imunoprevenção. O alvo utilizado no modelo de predição deste estudo foi um atributo que informa a probabilidade de ocorrência de causas de óbito evitáveis em determinados municípios, em dois grupos de causas, o de óbitos por tuberculose miliar e por hepatite aguda B.

O projeto deu início com o estudo dos dados, buscando identificar a porcentagem de ocorrência de óbitos previníveis em relação ao total de óbitos para cada estado brasileiro. Nele foi possível identificar as ocorrências de óbitos dos dois grupos de causas de óbito, definidas acima. A partir disso, foi realizada a seleção de atributos para o treinamento que possuíssem relevância para o alvo e que pudessem influenciar positivamente na performance do modelo de aprendizado.

Técnicas de pré-processamento de dados foram utilizadas para tratar os dados através da limpeza e identificação de valores faltantes, assim preparando os dados para a etapa de treinamento. Foi utilizada a técnica de aprendizagem supervisionada baseada em *Decision Tree* e também *Random Forest*. Os dados foram divididos em conjuntos de treino e teste, estes foram treinados e foi avaliada a qualidade da predição.

### 3. Metodologia

Esse trabalho considerou dados de estados brasileiros relacionados à causas de óbito evitáveis ao longo dos anos de 1996 a 2014. Ao longo da etapa de pré-processamento foram encontradas dificuldades que precisaram ser solucionadas. O trabalho teve como tarefas:

- a) observação do conjunto de dados divididos por estados buscando identificar padrões ao longo dos anos;
- b) pré-processamento dos dados, realizando o tratamento dos dados e identificando valores faltantes e preenchendo os mesmos;
- c) criação de uma função para identificação da porcentagem de ocorrências de óbitos evitáveis por ações imunopreventivas entre estados;
- d) escolha do alvo para treinamento correspondente ao atributo relacionado à causas de óbito previníveis;
- e) seleção de atributos relevantes para o alvo;
- f) retirada de classes do atributo alvo que identificavam outras causas de óbito que não fossem das classes de tuberculose miliar e hepatite aguda B;
- g) criação de modelos de aprendizado em *Decision Tree* e *Random Forest* para a classificação dos dois dos grupos relacionados a causas de óbito previníveis por ações de imunoprevenção, no subconjunto dos municípios brasileiros ao longo dos anos;
- h) avaliação do desempenho dos dois modelos com base na matriz de confusão e na métrica *f1-score*.

#### 4. Resultados e Discussão

Neste trabalho foram usados os algoritmos de *Decision Tree* e *Random Forest* em Python com Scikit-learning tendo como base dados relacionados a causas de óbito evitáveis por ações de saúde nos estados brasileiros entre os anos de 1996 e 2014.

Teve como objetivo avaliar comportamento e resultado dos modelos em *Decision Tree* e *Random Forest*, analisando a precisão para determinar causas de óbito relacionadas a tuberculose miliar e hepatite aguda B.

O conjunto de dados usado para treinamento e avaliação é composto por dados de todos os anos (1996 a 2014) e estados brasileiros. Este conjunto de dados possui 42 atributos, sendo um deles usado como atributo alvo. Foi possível retornar a matriz de confusão dos modelos em *Decision Tree* e *Random Forest*. A métrica *f1-score* do modelo em *Decision Tree* (Figura 1) obteve aproximadamente 0.69 para valores relacionados a tuberculose miliar, enquanto que 0.61 para valores relacionados a hepatite aguda B. Já a *f1-score* do modelo em *Random Forest* (Figura 2) obteve aproximadamente 0.72 para valores relacionados a tuberculose miliar e 0.74 para valores relacionados a hepatite aguda B.

	precision	recall	f1-score	support
A19	0.63	0.76	0.69	1272
B16	0.70	0.55	0.61	1264
accuracy				2536

Figura 1 – Métricas Avaliatórias da *Decision Tree* de Todos os Estados e Anos

	precision	recall	f1-score	support
A19	0.75	0.69	0.72	1272
B16	0.71	0.77	0.74	1264
accuracy				0.73 2536

Figura 1 – Métricas Avaliatórias da *Random Forest* de Todos os Estados e Anos

Na Tabela 1 é possível visualizar a matriz de confusão do modelo em *Decision Tree* onde, dos 1272 valores determinados como de tuberculose miliar, o algoritmo identificou 970 de forma correta, enquanto que dos 1264 de valores determinados como de hepatite aguda B, ele acertou 692.

Valor Predito	A19	B16	All
Valor Real			
A19	970	302	1272
B16	572	692	1264
All	1542	994	2536

A19 - Tuberculose miliar  
B16 - Hepatite aguda B

Tabela 1 – Matriz de Confusão *Decision Tree* de Todos os Estados e Anos

Na Tabela 2 é possível visualizar a matriz de confusão do modelo em *Random Forest* onde, dos 1272 valores determinados como de tuberculose miliar, o algoritmo identificou 884 de forma correta, enquanto que dos 1264 valores determinados como de hepatite aguda B, ele acertou 971. O modelo em *Random Forest* possui uma taxa de acerto, em relação a valores de hepatite aguda B, um pouco maior do que o modelo em *Decision Tree*.

Valor Predito	A19	B16	All
Valor Real			
A19	884	388	1272
B16	293	971	1264
All	1177	1359	2536

A19 - Tuberculose miliar  
B16 - Hepatite aguda B

Tabela 2 – Matriz de Confusão da *Random Forest* de Todos os Estados e Anos

Desta forma foi possível chegar ao que era pretendido inicialmente, compreendendo a importância da etapa de pré-processamento dos dados tendo como resultado uma precisão de qualidade frente aos desafios enfrentados. Como visto através das tabelas a classificação do grupo de hepatite aguda B apresenta um acerto preditivo maior no modelo em *Random Forest* em relação ao modelo em *Decision Tree*, enquanto que o grupo de tuberculose miliar prediz menos ocorrências de forma correta no modelo em *Random Forest* em comparação com o modelo em *Decision Tree*.

No anexo I é possível visualizar a imagem da *Decision Tree* do resultado acima descrito. Os atributos presentes correspondem a *def\_necropsia* que define a confirmação do diagnóstico de óbito por necropsia; *def\_est\_civil* que define o estado civil do falecido; *ano\_obito* que define o ano de óbito; *res\_amazonia* que indica se o município de residência do falecido faz parte da Amazônia Legal; *res\_nomex\_municipio* que define o nome do município de residência da pessoa que foi à óbito e *ocor\_sigla\_uf* que define a sigla do estado de ocorrência do óbito.

## 5. Conclusão

Neste trabalho foi possível criar um modelo de aprendizado baseado nas técnicas de *Decision Tree* e *Random Forest* para classificação de atributos ligados à causas de óbito evitáveis por ações de imunoprevenção. Estes foram capazes de classificar novas entradas com características ligadas à causas de morte por tuberculose miliar e hepatite aguda B. É possível observar, nas tabelas acima que, para cada tipo de problema, determinados modelos de aprendizado de máquina classificam com maior exatidão e produzem menores erros de predição. No presente trabalho pode-se declarar que o modelo em *Random Forest* gera menos erros do que o modelo em *Decision Tree*. Concluímos, portanto, que o processo de aprendizado é complexo e bastante dependente de decisões tomadas durante o treinamento, assim como é possível afirmar que o tipo de problema a ser trabalhado influencia nos resultados dos modelos.

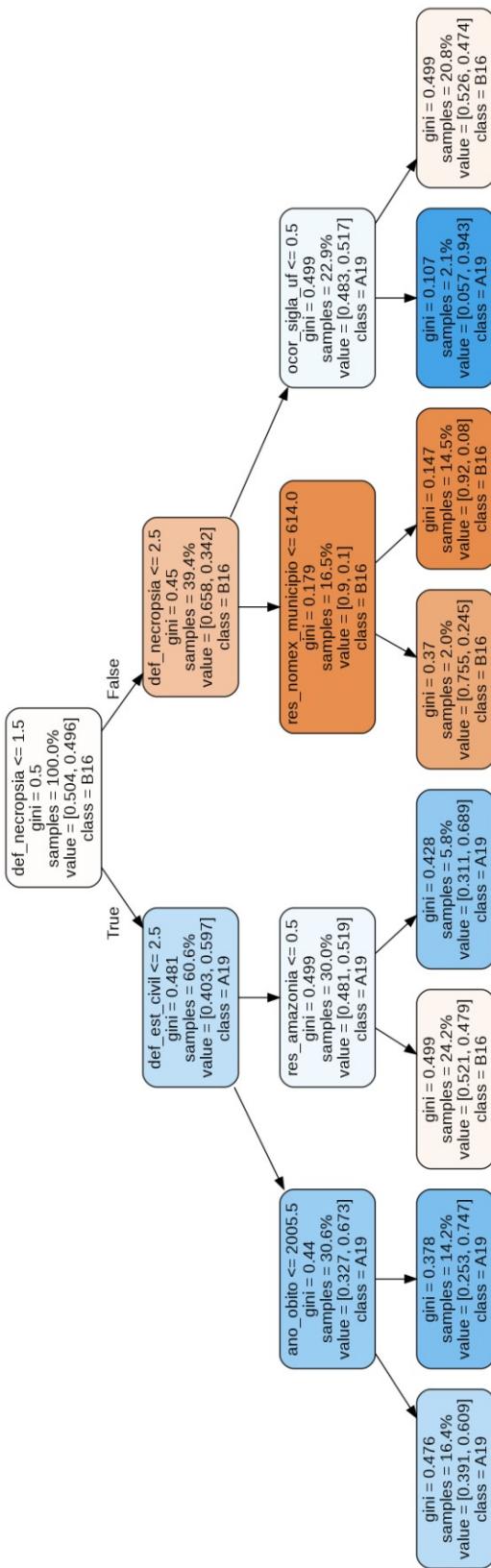
Como próximos passos, buscaremos avaliar outras causas de óbito evitáveis através de um problema multiclasse, além de testar como outros algoritmos de aprendizado se comportam nestes casos.

## 6. Referências Bibliográficas

1. Malta, Deborah Carvalho et al. Lista de causas de mortes evitáveis por intervenções do Sistema Único de Saúde do Brasil. *Epidemiologia e Serviços de Saúde*, v. 16, n. 4, p. 233-244, 2007
2. Thomas G. Dietterich, *Ensemble Methods in Machine Learning*, ISBN: 3-540-67704-6, Springer-Verlag London, UK, 2000

## ANEXO I

### Árvore de Decisão de Todos os Estados e Anos



Bolsista:

*René Constancio Nunes de Lima*

Orientadores:

*Elson M. Toledo(LNCC)*

*Regina Célia P. Leal Toledo*

*(UFF- co-orientadora)*

**Relatório do projeto:**

**SIMULAÇÃO NUMÉRICA E COMPUTACIONAL DO TRÁFEGO VIÁRIO**

LNCC

PROGRAMA PIBIC/PIBITI

**2019-2020**

## **1- Objetivos**

O projeto em desenvolvimento, continuidade de um trabalho anterior, tem o objetivo de propor e analisar modelos para o tráfego viário para melhor entender sua dinâmica e auxiliar no seu planejamento e controle. Utilizamos a modelagem microscópica, baseada em Autômatos Celulares (AC), e objetivamos considerar, em todas as análises, a influência de diferentes perfis de aceleração e desaceleração de motoristas, principal característica do projeto.

Na etapa atual, propomos e analisamos um algoritmo, chamado de antecipação. Essa família de algoritmos para definir a movimentação de um determinado veículo considera, de alguma forma, que o veículo à frente também se movimentará no instante de tempo considerado. Pretende-se, com isso, avaliar a importância (ou não) de se considerar a modelagem da antecipação no modelo proposto no projeto anterior.

## **2- Introdução**

O comportamento do tráfego viário influencia drasticamente a qualidade de vida dos cidadãos, principalmente no meio urbano. Muitas soluções já foram propostas com objetivo de reduzir os efeitos negativos do crescimento do número de veículos em grandes cidades, como veículos elétricos para diminuir a poluição do ar, ou veículos autônomos, que prometem reduzir os congestionamentos e o número de acidentes. Desta forma, analisar a dinâmica do tráfego, em diversas situações, é de extrema importância. Para essas análises, são utilizados modelos matemáticos macroscópicos e microscópicos. O modelo microscópico de Autômato Celulares (AC), utilizado neste estudo, tem se mostrado eficiente e utilizado em várias simulações com sucesso. Uma das suas principais vantagens é que é facilmente implementado, tem baixa custo operacional e representa bem as características fundamentais do tráfego. Contudo, de modo geral, esses modelos, apesar de probabilísticos, tratam todos os motoristas de maneira uniforme. Ao longo deste projeto, têm sido propostos e analisados modelos que possibilitem avaliar a influência de diferentes comportamentos de motoristas na dinâmica do tráfego. Zamith [1] propõe um modelo de AC em que as diferentes políticas de aceleração são modeladas utilizando uma Função de Densidade de Probabilidade (FDP) não uniforme, a FPD Beta.

No ano anterior, o tradicional modelo probabilístico de AC proposto por Nagel e Schreckenberg [2], conhecido como modelo NaSch, foi modificado para possibilitarmos a definição de diferentes perfis de comportamento, incluindo diferentes valores de aceleração e desaceleração a cada instante de tempo. Esta modificação consiste em utilizar uma discretização do espaço mais refinada do que a proposta usualmente para o modelo NaSch e uma Função de Densidade de Probabilidade (FDP) não uniforme. A FDP Beta foi utilizada para definir os diferentes perfis de aceleração e desaceleração. Esses perfis propostos variam de um mais cauteloso, que tem a tendência de acelerar/desacelerar mais suavemente, a um mais agressivo, que acelera e desacelera mais bruscamente, e são obtidas através das diferentes médias da FDP utilizada. Estudamos a influência desses perfis na dinâmica de pistas com interseção, tendo por base o estudo do Marzoug [3] para tentar reproduzir e analisar seus resultados, os quais foram publicados [4]. No entanto, o modelo NaSch opera como um método explícito que ao definir critérios para a movimentação de um determinado veículo no tempo atual, tendo em vista o espaço disponível à sua frente, desconsidera que o veículo à frente também se movimentará.

Dessa forma, apresentamos um algoritmo de antecipação em que, para o cálculo da velocidade de um veículo, é considerado o movimento do veículo à sua frente neste mesmo instante de tempo. Para isso, nos baseamos em diferentes algoritmos de antecipação já propostos

[5-7]. Além disso, propomos uma forma de diminuir as freadas bruscas, usuais nesse tipo de modelagem, para melhor representação da movimentação real de um veículo.

Este relatório relata o algoritmo de antecipação proposto, os resultados obtidos e comparações realizadas, para que no futuro implementemos um modelo de antecipação em pistas com múltiplas faixas e cruzamentos, que estão sendo desenvolvidas por outros membros do grupo, com objetivo de representar melhor a realidade.

### **3- Material e Métodos ou Metodologia**

Para apresentação do trabalho desenvolvido, primeiro, apresentamos o modelo NaSch modificado, anteriormente proposto e a seguir, o modelo de antecipação analisado nessa etapa do projeto.

#### **3.1 Modelo Nasch**

##### **(i) O modelo NaSch tradicional:**

Neste modelo de AC, em sua proposta tradicional, a via é unidimensional e discretizada em células, onde cada célula ou é ocupada por um veículo ou está vazia. Cada veículo ocupa uma célula, normalmente considerada com  $7,5m$ , e o tempo também é discretizado, sendo cada instante de tempo  $t$  considerado *1 segundo*. Cada veículo  $i$  ocupa a posição  $x(i, t)$ , no instante de tempo  $t$ , e tem velocidade dada por  $v(i, t)$ . A distância entre um veículo e outro, dada pelo número de células vazias na frente de cada veículo, é definida como  $d(i, t) = x(i + 1, t) - x(i, t) - L(i)$ , sendo  $L(i)$  o número de células ocupadas por um veículo  $i$ , sendo que nesse modelo,  $L(i) = 1$  para todos os veículos. O algoritmo consiste em um conjunto de quatro regras, aplicadas paralelamente a todos os veículos, a cada segundo.

##### **Algoritmo 1**

- 
- |                    |  |
|--------------------|--|
| (1) Aceleração:    | $v(i, t + 1) = \min[v(i, t) + A, V_{max}]$                         |
| (2) Desaceleração: | $v(i, t + 1) = \min[v(i, t + 1), d(i, t)]$                         |
| (3) Desaceleração: | $v(i, t + 1) = \max[v(i, t + 1) - A, 0]$ , com probabilidade $p_b$ |
| (4) Movimento:     | $x(i, t + 1) = x(i, t) + v(i, t + 1)$                              |
- 

sendo  $V_{max}$  a velocidade máxima da via,  $A = 1$  a aceleração dos veículos e  $p_b$ , que modela a incerteza no comportamento dos motoristas, a probabilidade de o motorista não querer acelerar, ou desacelerar, dependendo da situação. Usualmente, os testes são realizados em uma pista circular, isto é, com condições de contorno periódicas.

##### **(ii) O modelo NaSch modificado (NaSch com beta):**

No modelo NaSch original,  $A = 1$  e cada veículo acelera ou desacelera sempre de  $1\text{cel}/s^2$ , o que corresponde a  $7,5m/s^2$ , que é um valor alto para dados reais. Para gerar diferentes políticas de aceleração, dividimos a célula de  $7,5m$  do NaSch em células menores, para que seja possível que cada motorista acelere, caso possa, de diferentes formas. Assim, o valor da variável  $A$ , no *Algoritmo 1*, é probabilístico e é definido, para cada motorista, pela FDP Beta, com diferentes médias e variâncias. Essa FDP gera um valor de  $\alpha$  ( $0 < \alpha < 1$ ), que é utilizada para calcular a nova aceleração, dada por  $A = \text{int} [ (1 - \alpha) A_{max}]$ , onde  $\text{int}$  é uma função que retorna o inteiro mais próximo. Assim, o parâmetro  $\alpha$  designa como ele irá acelerar: mais suavemente ou mais bruscamente. A FDP Beta utilizada é definida como:

$$\beta(a, b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} x^{a-1} (1-x)^{b-1}$$

onde  $0 \leq x \leq 1$  e  $\Gamma(n+1) = n!$ ,  $n$  um inteiro positivo e para cada conjunto de parâmetros  $a$  e  $b$ , resulta uma função  $\beta(a,b)$ . Para gerar esses valores, utiliza-se a Técnica de Rejeição do Método de Monte Carlo. Valores de  $\alpha$  perto de 0 produzem acelerações mais próximas de  $A_{max}$ , enquanto valores mais perto de 1 produzem acelerações próximas de 0. Para cada par de valor  $a$  e  $b$ , podemos calcular a média  $M = \frac{a}{a+b}$ . Então, podemos relacionar cada perfil de aceleração a um par de parâmetros  $a$  e  $b$ , em que o valor médio determina a tendência de aceleração de um perfil.

### 3.2 Modelo proposto

#### (i) Antecipação:

Neste modelo, o algoritmo muda em relação ao anterior em relação aos seguintes aspectos: (a) não há a etapa de desaceleração do item 3 do Algoritmo 1. O motorista só não acelerará se o valor de  $\alpha = 0$  ou se não tiver distância suficiente para acelerar. Além disso, muda também no cálculo da distância que um veículo tem para andar. Antes ela era considerada como  $d(i,t) = x(i+1,t) - x(i,t) - L(i)$ . Aqui chamamos a distância de uma distância efetiva,  $d_{ef}(i,t)$ , que considera o movimento do veículo da frente:

$$d_{ef}(i,t) = d(i,t) + \min [v(i+1,t), d(i+1,t)]$$

assim,  $d_{ef}(i,t) \geq d(i,t)$ , fazendo com que o veículo de trás tenha mais distância para andar. Isso significa que considera-se que o veículo que está à frente do veículo considerado manterá sua velocidade, caso ele possa, ou seja, caso exista distância à sua frente para isso, dada por  $d(i+1,t)$ . No algoritmo sem antecipação, considera-se que o veículo à frente não se movimentará. Para exemplificar essa diferença, apresentamos na Figura 1 uma configuração que representa a ideia de antecipação, onde evidenciamos a diferença entre as distâncias consideradas pelos veículos para suas movimentações nos dois modelos:

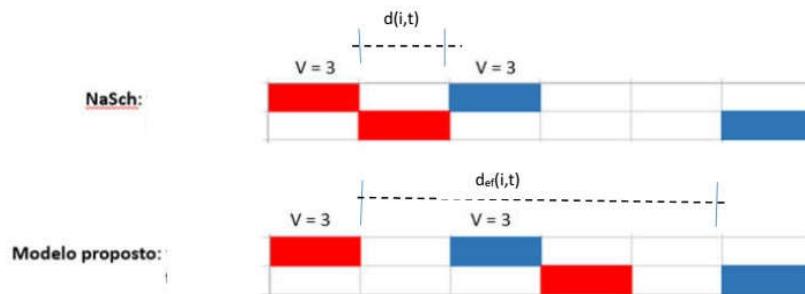


Figura 1 - Comparando a dinâmica de um algoritmo sem antecipação vs com antecipação

#### (ii) Freadas bruscas:

Nos resultados analisados, tanto no modelo NaSch com beta quanto no modelo com antecipação, notamos que as freadas podem ser muito bruscas quando um veículo precisa reduzir sua velocidade por não ter distância suficiente para percorrer. Então, modificamos o algoritmo para amenizar esses casos citados e, com isso, reintroduzimos a possibilidade de desaceleração. Assim, em algumas circunstâncias, o veículo antecipa sua freada para poder frear mais

suavemente. Consideramos que o veículo irá frear se o veículo à sua frente freou, e se ele estiver mais rápido e próximo a esse veículo à sua frente. Para isso, introduzimos o conceito de luz de freio,  $b(i)$ , valor inicialmente nulo, e que passa a ser  $b(i) = 1$  sempre que o veículo  $i$  frear. Como um veículo só tem a informação da luz de freio do veículo da sua frente no instante anterior, ele não sabe se o veículo da frente freou no instante atual. Por isso, foi necessário introduzir uma distância de segurança,  $d_{seg}$ , para garantir que não haja colisão. Assim, o algoritmo ficou da forma:

### **Algoritmo 2**

---

(1) Aceleração:  $v(i, t + 1) = \min[v(i, t) + A, Vmax]$

(2) Desaceleração: Se  $\Delta v(i, t) > 0$  e  $b(i + 1) = 1$  e  $Ts < Th$ :

$$v(i, t + 1) = \max [0, \min[v(i, t + 1), d(i, t)] - d_{seg}] \text{ e } b(i) = 1$$

(3) Desaceleração: Se  $v(i, t + 1) > d_{ef}(i, t)$ :

$$v(i, t + 1) = d_{ef}(i, t) \text{ e } b(i) = 1$$

(4) Movimento:  $x(i, t + 1) = x(i, t) + v(i, t + 1)$

onde  $\Delta v(i, t) = v(i, t) - v(i + 1, t)$  indica se o veículo está se aproximando ao da sua frente,  $Ts = d(i, t)/\Delta v(i, t)$  é o intervalo temporal para alcançar o carro da frente e :

$$d_{ef}(i, t) = d(i, t) + \max [0, \min[v(i + 1, t), d(i + 1, t)] - d_{seg}]$$

## **4- Resultados e Discussão**

Embora no projeto tenham sido considerados diferentes perfis de comportamento dos motoristas, apresentamos nesse relatório apenas resultados para os perfis de aceleração/desaceleração: (a) Agressivo, com  $\beta(10,30)$ , Aceleração média =  $4 \text{ cell}/\text{s}^2 = 6 \text{ m/s}^2$ ; (b) Cauteloso: com  $\beta(30,10)$ , Aceleração média =  $1 \text{ cell}/\text{s}^2 = 1.5 \text{ m/s}^2$ .

### **4.1 Modelo de antecipação**

#### **(a) Movimentação dos veículos**

Inicialmente, apresentamos nas Figuras 2 (a) e 2(b) resultados da Posição-Tempo para o modelo com antecipação, para os perfis agressivo e cauteloso. A pista contém 400 células de 1,5m e foram colocados 4 veículos parados no início da pista um à frente do outro. Os veículos aceleram, de acordo com seu perfil, até o final da pista, onde devem parar. Consideramos  $p_b = 0$  e a velocidade máxima da via, nas duas situações, é  $V_{max} = 25 \text{ cel/s} = 135 \text{ km/h}$ .

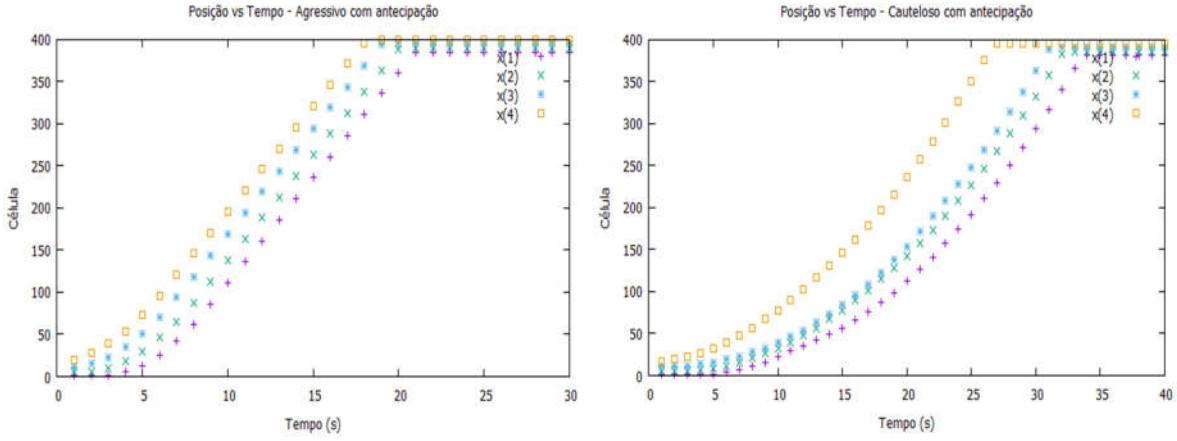


Figura 2 – Posição-Tempo para o modelo de antecipação com (a) perfil Agressivo e (b) perfil Cauteloso

Podemos observar que a movimentação dos diferentes perfis representa o resultado esperado, com o motorista com perfil de aceleração agressivo chegando mais rapidamente ao final da via.

### (b) Diagramas fundamentais

As Figuras 3 (a) e 3(b) apresentam resultados para os diagramas de Fluxo-Densidade para o modelo NaSch com Beta e para o modelo de antecipação proposto. Consideramos uma via de comprimento total 7,5km e as simulações reproduzem 20.000 instantes de tempo, onde os 17.000 primeiros instantes foram descartados, para desconsiderar efeitos transitentes. Para cada densidade, são distribuídos veículos aleatoriamente pela pista, com velocidade inicial nula. A densidade  $\rho$  é a porcentagem de células ocupadas na pista. Um veículo ocupa 5 células de 1,5 m (total de 7.5m, como no modelo NaSch tradicional) e a aceleração máxima,  $A_{max} = 5 \text{ cell/s}^2$ . Consideramos  $p_b = 0$  e a velocidade máxima é  $V_{max} = 25 \text{ cel/s} = 135 \text{ km/h}$ .

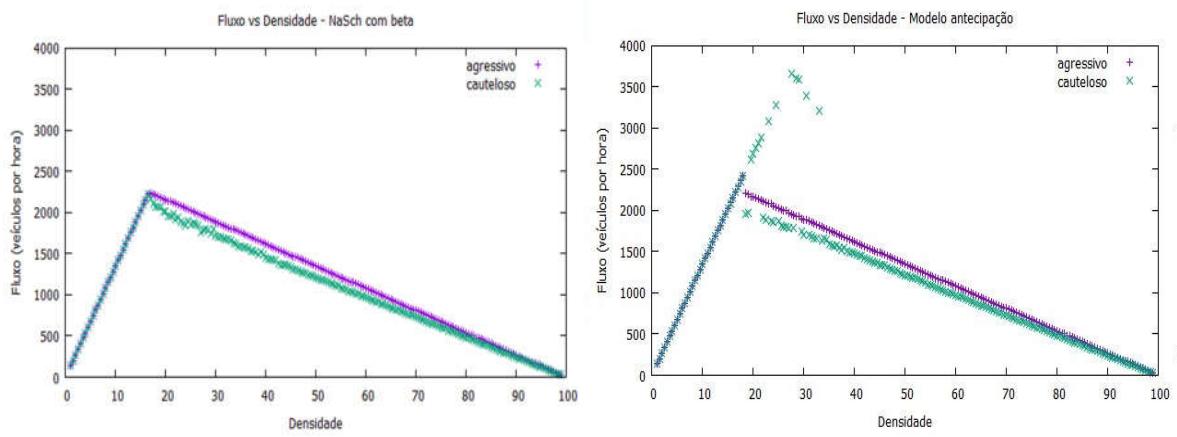


Figura 3 – Fluxo-Densidade para os perfis agressivo e cauteloso (a) do modelo NaSch com beta; (b) modelo de antecipação

Podemos observar que o valor máximo de fluxo livre não foi substancialmente alterado pelo modelo de antecipação. No entanto, como mostrado em resultados anteriores, esse valor máximo já tem uma boa aproximação com os dados reais, para uma pista com as características da modelada. Além disso, e principalmente, cabe observar que o modelo de antecipação proposto

foi capaz de reproduzir a região de metaestabilidade, uma das fases do diagrama fundamental, que não era representada pelo modelo NaSch com Beta.

#### 4.2 Modelo de antecipação com suavização de freada

A seguir, comparamos as variações de velocidades do modelo NaSch com beta com o modelo de antecipação com suavização de freada, que segue o Algoritmo 2, com  $Th = 6s$  e  $d_{seg} = 7,5m$ . As configurações da simulação da via foram as mesmas do exemplo anterior. Nas Figuras 4 (a) e (b) foram acompanhadas as variações de velocidades de veículos dos perfis agressivo e cauteloso em alguns instantes de tempo:

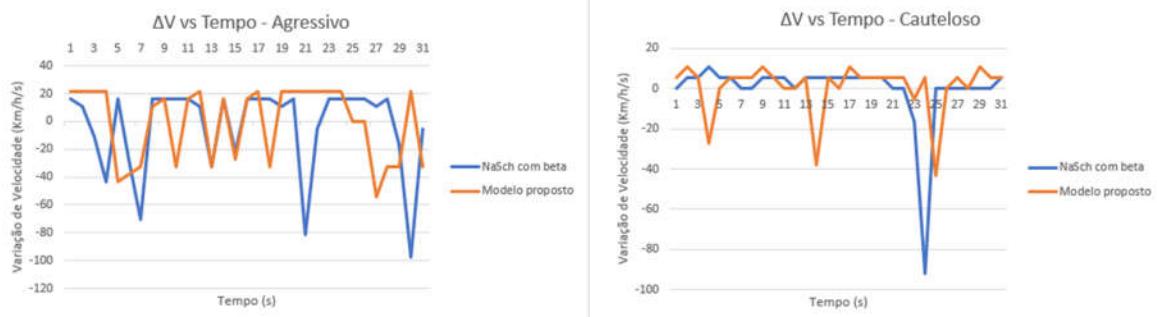


Figura 4 –  $\Delta V$ -Tempo de um veículo dos perfis (a) Agressivo e (b) Cauteloso do modelo NaSch com beta e um do Modelo proposto

Podemos observar que o modelo de suavização de freadas foi capaz de diminuir o valor máximo da variação de velocidade, como queríamos, porém, ainda necessitando de calibração em relação aos valores empíricos utilizados no exemplo aqui apresentado.

#### Conclusões e próximos passos

A forma como o motorista acelera e desacelera, normalmente, é um comportamento não observável em medições usuais, o que torna importante de se modelar esse comportamento, para conseguirmos avaliar sua influência na dinâmica do tráfego. Ao longo desse projeto [5,7,8] temos mostrado que a dinâmica do tráfego é influenciada não só pela densidade da via e pelas velocidades máximas, mas também pela forma como chegam nessa velocidade.

Neste trabalho, avaliamos como um modelo simples de antecipação pode melhorar a representação do fluxo de tráfego numa pista circular com faixa única. Com isso, conseguimos representar a região de metaestabilidade, uma das fases do fluxo, existente quando começa a haver interação entre veículos.

Além disso, observamos que o modelo de antecipação proposto com uma freada suavizada é capaz de representar um pouco melhor a realidade, porém ainda necessita de calibração dos parâmetros sugeridos e comparações com dados reais de frenagem.

#### Bibliografia

1. Zamith, M.P., Um Modelo de Autômato Celular Aplicado ao Tráfego Viário com Múltiplos Perfis de Condutores, Tese Doutorado, Pós-Graduação em Computação, Instituto de Computação-UFF.
2. Nagel, K., Schreckenberg, M.: A cellular automaton model for freeway traffic. Journal de physique I 2(12), 2221-2229 (1992).

3. Marzoug,R., Ez-Zahraouy, H., Benyoussef, A.: Cellular automata traffic flow behavior at the intersection of two roads, *Physica Scripta*, 89(6), 1-7 (2014).
4. de Almeida I.M., Leal-Toledo R.C.P., Toledo E.M., Cacau D.C., de Lima R.C.N., Malta S. (2019) Influence of Drivers' Behavior on Traffic Flow at Two Roads Intersection. In: Misra S. et al. (eds) Computational Science and Its Applications – ICCSA 2019. ICCSA 2019. Lecture Notes in Computer Science, vol 11619. Springer, Cham.
5. Knospe, W., Santen, L., Schadschneider, A., Schreckenberg, M.: Towards a realistic microscopic description of highway traffic. *Journal of Physics A: Mathematical and general* 33(48), pp. L477 (2000).
6. Knospe, W. et al. A realistic two-lane traffic model for highway traffic. *J. Phys. A*, v. 35, p. 3369–3388, 2002.
7. Leal-Toledo, R. C. P., Magalhães, G.V. P., Miranda, I., Toledo, E. M., Traffic Flow Simulation Under the Influence of Different Acceleration Policies, Int'l Conf. Scientific Computing ( CSC'17), pp 90-94 (2017).
8. I. M. Almeida , R. C. P. Leal-Toledo, E. M. Toledo, D. C. Cacau, e G. V. P. Magalhães, Drivers' behavior effects in the occurrence of dangerous situations which may lead to accidents, 5th Workshop on Traffic and Cellular Automata, Lecture Note in Computer Science, 2018.

Bolsista:

*Tiago Maia de Lacerda Brasil*

Orientadores:

*Sandra Mara Cardoso Malta*

*Regina C. P. Leal Toledo*

*(UFF – Co-orientadora)*

**ESTUDO DE MODELOS DE SIMULAÇÃO E DE PREVISÃO DO  
TRÁFEGO VIÁRIO**

PROGRAMA PIBIC

**2019-2020**

## **1. Objetivo**

Este projeto de Iniciação Científica tem como objetivo principal o estudo e análise dos principais métodos para a modelagem do tráfego viário.

Modelos macroscópicos e microscópicos, baseados em car following ou em autômatos celulares (AC), têm sido propostos como opções para a modelagem do tráfego. Essas modelagens utilizam desde sistemas contínuos, onde o tráfego é considerado como uma corrente de fluxo contínua, e descrito por equações diferenciais parciais, a sistemas discretos no tempo e no espaço, como os modelos baseados em AC [1-5].

Neste projeto, busca-se avaliar a influência que diferentes comportamentos adotados na direção podem causar na dinâmica do tráfego. Esse enfoque, modelagem de comportamentos, já vem sendo desenvolvido na pesquisa, em particular, através de modelos microscópicos baseados em AC [6-9]. No entanto, recentemente, observa-se uma tendência, em todos os tipos de abordagens dada ao problema, de se considerar, de alguma forma, os diferentes perfis de motoristas na avaliação da dinâmica do tráfego, em particular, da direção agressiva e da cautelosa ou tímida [10-13]. Dessa forma, nesta etapa do projeto, modelos microscópicos foram estudados e analisados.

## **2. Introdução**

A modelagem e simulação do tráfego viário é uma importante ferramenta para auxiliar na compreensão de fenômenos e na otimização do uso da malha viária (educação, sistema de interconexão), no planejamento da construção e manutenção de redes de trânsito e na predição de cenários que possam minimizar congestionamentos e, consequentemente, a poluição sonora e do ar.

Muitas técnicas já foram propostas para modelar o tráfego, apresentando vantagens e desvantagens para situações específicas de interesse, e podem ser categorizadas em modelos macroscópicos e modelos microscópicos.

Os modelos macroscópicos descrevem a dinâmica do fluxo viário de forma análoga a fluidos em movimento, sendo ocasionalmente chamados modelos hidrodinâmicos. Estes modelos têm como resultado campos dinâmicos de variáveis como fluxo e velocidade média ao longo do tempo, e são especialmente úteis quando não se tem interesse em detalhes em relação à direção dos veículos (comportamento do motorista, características do veículo ou a interação entre veículos), quando há grande variação de dados ou quando o tempo de simulação deve ser o menor possível (por exemplo em aplicações em tempo real) [14].

Os modelos microscópicos descrevem a dinâmica do fluxo viário a partir das interações entre cada veículo da malha viária, aplicando-se as mesmas regras de movimentação para todos os veículos da via. Estes modelos são capazes de representar aspectos específicos de cada motorista e/ou veículo na malha viária, e são especialmente úteis quando se tem interesse na interação entre veículos ou na análise da influência do comportamento do motorista. Além disso, permitem considerar frotas de diferentes veículos, como veículos de tamanhos variados, veículos automatizados, entre outros.

Durante o presente projeto foram estudados e testados os principais modelos microscópicos baseados em Autômatos Celulares (AC) e em *Car-following*, cuja metodologia e resultados serão comentados a seguir.

### 3. Metodologia

A metodologia adotada pode ser dividida nas seguintes etapas:

**(a)** Inicialmente, foi realizado um estudo da bibliografia recomendada para conhecimento do problema do tráfego viário e da técnica de modelagem, via AC, que vem sendo utilizada pelo grupo de pesquisa [6 - 9]. Com isso, foi possível participar das reuniões e discussões de trabalho do grupo e foi desenvolvido um *software* para auxiliar no trabalho que vinha sendo desenvolvido via AC. Esse *software* possibilitou visualização dos dados gerados, permitindo analisar o impacto do tempo inicial descartado na coleta de dados da simulação para a determinação de velocidades médias e fluxo da pista. Esse descarte é utilizado nas simulações para minimizar os efeitos transitentes oriundos das diferentes condições iniciais, onde os veículos são alocados na via.

**(b)** Numa segunda etapa, foi iniciado um estudo da modelagem via Car-following, ainda não utilizada pelo grupo de pesquisa. Inicialmente foi feita uma pesquisa bibliográfica para conhecimento da técnica e escolha do método a ser implementado [14, 15]. Foi decidido que o modelo a ser implementado seria o modelo Gipps. Depois de uma análise inicial e discussão sobre sua implementação, optou-se pelo modelo Gipps Simplificado, que poderia ser melhor adaptado para representar diferentes perfis de direção.

**(c)** Por fim, foi feita uma implementação do modelo Gipps Simplificado [14], que assim como outros modelos *Car-following*, é constituído num modelo incremental onde o tempo é discretizado e procura-se obter uma aproximação para a velocidade a cada instante de tempo. Essa velocidade é escolhida com o menor valor entre o que o veículo se movimentaria, dada uma determinada aceleração, a velocidade máxima definida para a via, ou deve frear, dada uma determinada desaceleração e a distância do veículo que está a sua frente. Assim, para cada veículo, é calculada a velocidade

$$v(t + \Delta t) = \min[v + a\Delta t, v_0, v_{seg}(s, v_l)] \quad (1)$$

$$v_{seg}(s, v_l) = -b\Delta t + \sqrt{b^2 T^2 + v_l^2 + 2b(s - s_0)} \quad (2)$$

onde:

*a*: aceleração máxima dos veículos (m/s)

*b*: desaceleração máxima dos veículos (m/s)

*s*: distância do parachoque frontal do veículo (*n*) ao traseiro do veículo à frente (*n* – 1)

*s*<sub>0</sub>: distância mínima que o veículo (*n*) deseja ficar do veículo à frente (*n* – 1)

*v*<sub>0</sub>: velocidade máxima em que o veículo (*n*) deseja chegar

*v*<sub>*l*</sub>: velocidade do veículo (*n* – 1)

*Δt*: tempo de reação do motorista / passo da simulação

Normalmente, a velocidade é determinada pelos dois primeiros termos da expressão (1), no caso de fluxo livre, e pela velocidade de segurança, dada por (2), quando há interação entre os veículos. Nota-se que todos os veículos têm que ser precedidos por um outro veículo, o veículo líder. Dessa forma, ou precisamos fornecer a velocidade ao longo do tempo para o líder ou implementar uma pista circular, com condições de contorno periódicas.

#### 4. Resultados e Discussão

Neste relatório serão apresentados somente os resultados obtidos pelo modelo Gipps Simplificado, na modelagem *Car-following*. Para isso, e para utilizar a mesma metodologia utilizada em modelos AC, foi implementada uma pista circular (condições de contorno periódicas) de comprimento L, e definidos dois perfis teste de motoristas, um cauteloso e outro agressivo, com características similares às adotadas em AC [7-9]. Assim, o motorista cauteloso acelera mais suavemente e mantém uma maior distância do veículo à sua frente. De forma oposta, o motorista agressivo acelera e desacelera mais bruscamente, e mantém uma menor distância do veículo líder. Assim, foram definidos os perfis de motoristas:

$$\text{Cauteloso: } a = 1.5m/s^2, d = 1.5m/s^2, s_0 = 3.0m$$

$$\text{Agressivo: } a = 6.0m/s^2, d = 3.0m/s^2, s_0 = 1.5m$$

Os valores da aceleração e desaceleração utilizados foram os sugeridos em [9] para modelos AC. Nesse caso, como o modelo de AC é probabilístico, esses valores representavam a média das acelerações de cada perfil.

Foram realizadas então diversas simulações para cada perfil, a fim de determinar a velocidade média dos veículos na pista para diferentes densidades (ocupação da pista).

A Figura 1 apresenta o diagrama de velocidade – densidade em uma pista de 5km de comprimento, com velocidade máxima de 135km/h, comparando os diferentes perfis. Como condição inicial, os veículos começam parados e espaçados um do outro a uma distância de  $s = 3m$ , sendo simulados 1800s e considerados apenas os últimos 900s no cálculo das velocidades médias, a fim de minimizar a influência das condições iniciais (de acordo com o estudo realizado no início do projeto).

Foram realizadas simulações para densidades variando de 0.01 a 0.99, com intervalos de 0.01, e os resultados podem ser comparados aos obtidos pelo modelo de AC descrito em [9].

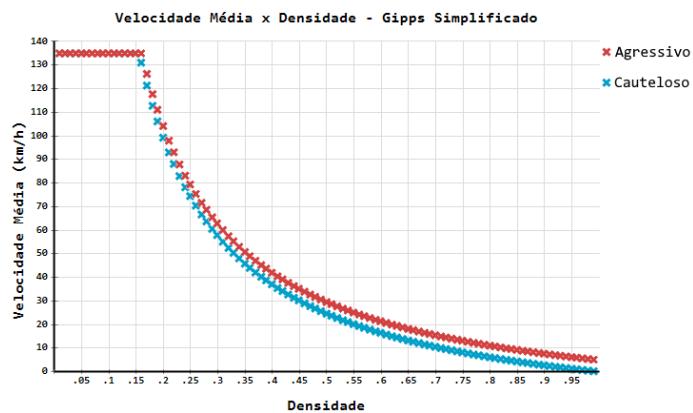


Figura 1 – Diagrama velocidade média – densidade no modelo Gipps Simplificado

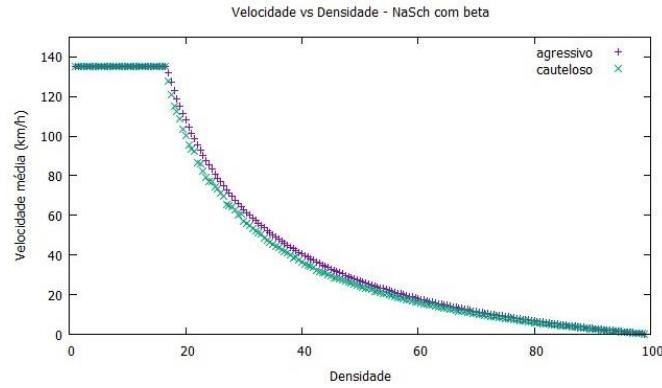


Figura 2 – Diagrama velocidade média – densidade no modelo NasCh com beta [9]

Nota-se que o modelo Gipps Simplificado apresentou o mesmo tipo de comportamento que o obtido com o algoritmo em AC, também proposto pelo grupo e descrito em [9]. Pode-se observar, também, que na região de fluxo livre, os dois perfis conseguem andar na velocidade média da via e, quando começa a interação entre os veículos, o perfil cauteloso tem uma velocidade média mais baixa.

A seguir, para mostrar a diferença da política de aceleração dos dois perfis, apresenta-se nas *Figura 3* a velocidade do veículo mais à frente da frota em cada perfil nos instantes de tempo iniciais (descartados no exemplo anterior). Na figura 3a a densidade da via era 0.12, região de fluxo livre, enquanto na figura 3b a densidade da via é 0.40 e há interação entre veículos, o que faz com que a velocidade diminua até que o motorista possa começar a se movimentar novamente. Como esperado, o motorista com perfil mais agressivo alcança mais rapidamente a velocidade máxima.

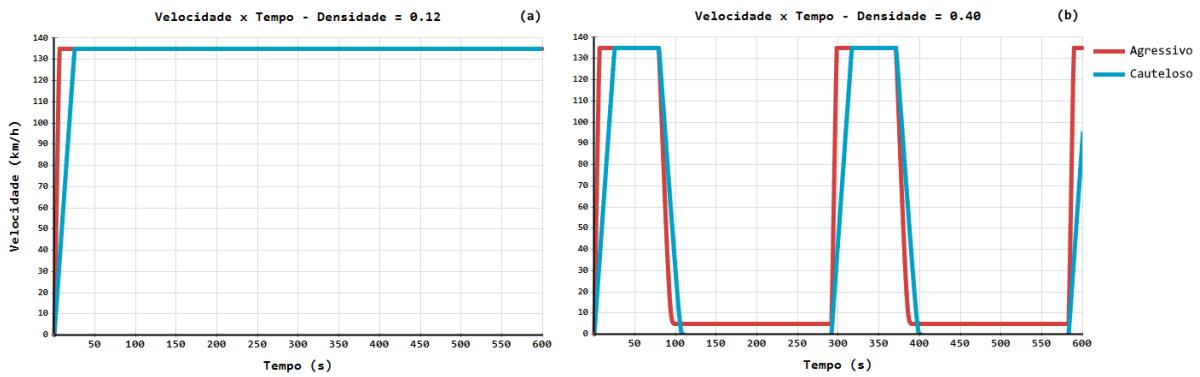


Figura 3 – Diagrama velocidade – tempo dos diferentes perfis no modelo Gipps Simplificado (600s)

Observa-se na figura 3a que na região de fluxo livre (densidades baixas, 0.01~0.15), ambos os perfis alcançam e mantém a velocidade máxima. Já nas regiões onde há congestionamento, na figura 3b, este se propaga quando o veículo líder encontra com o último veículo da frota, resultando em situações onde um perfil mais agressivo se mantém em movimento, enquanto um mais cauteloso para e espera os veículos à frente se distanciem para acelerar novamente.

É possível também notar que o congestionamento ocorre mais cedo com o perfil agressivo, visto que os veículos completam a volta na pista em menos tempo, encontrando mais rapidamente com o último da frota. Percebe-se também que o congestionamento no perfil cauteloso gera uma diminuição maior da velocidade, pois os veículos não arriscam freadas tão bruscas e mantém uma maior distância do veículo à sua frente.

Esse conjunto de resultados mostra que o modelo escolhido é capaz de representar, em média, o que se espera dos diferentes comportamentos no tráfego e poderá ser promissor nas análises dessas considerações.

## 5. Conclusão

Os modelos de Autômatos Celulares são, de forma geral, probabilísticos, discretizados no tempo e no espaço, enquanto os modelos de *Car-following* são determinísticos e discretizados apenas no tempo. Nestes, o movimento de um veículo é regido pelo movimento do veículo que o antecede (à sua frente), enquanto nos modelos AC, como o proposto por Nagel e Schreckenberg [16], os veículos seguem regras que descrevem seu *desejo* de movimentação, que não é realizado caso não haja espaço, com alguma probabilidade  $\rho$  de não seguir este comportamento.

Apesar do estudo dos modelos *Car-following* ainda estar em fase inicial, o modelo selecionado se mostrou de simples implementação e apresentou possibilidade do uso de diferentes calibrações para representar diferentes perfis de comportamento no tráfego e a avaliar influência destes em sua dinâmica.

Foi possível observar semelhança aos resultados obtidos com o modelo AC previamente implementado, possibilitando um acompanhamento próximo das diferenças e semelhanças entre diferentes classes de modelos microscópicos. Como o espaço não é discretizado, isso pode trazer facilidades para implementar diferentes tamanhos de veículos em relação aos modelos AC e também um menor custo computacional, pois há necessidade somente de armazenamento de informações sobre cada veículo enquanto com modelos de AC há necessidade de se guardar informações sobre cada célula e um veículo pode ocupar mais de uma célula. De forma geral, em todas as modificações propostas pelo grupo para permitir uma melhor representação da realidade, os veículos comuns ocupam mais de uma célula. Nos melhores resultados encontrados, ocupam 5 células.

Numa continuidade do projeto, outros modelos de *Car-following* podem ser avaliados e comparados, e, possivelmente, introduzida certa aleatoriedade no comportamento dos motoristas, a fim de possibilitar análises mais reais em situações específicas.

## 6. Referências Bibliográficas

[1] Daganzo, C.F. The cell transmission model. Part I: A Simple Dynamic Representation of Highway Traffic, *Transportation Research Part B: Methodological*, v. 28, 269-287, 1994.

[2] Daganzo, C.F. The cell transmission model, Part II: network traffic, *Transportation Research Part B: Methodological*, v. 29, 79-93, 1995.

[3] Schadschneider, A. and M. Schreckenberg, Cellular Automaton Models and Traffic Flow, *Journal of Physics A*, v. 26, 679-683, 1993.

[4] Braekstone, M., McDonald, M. Car-following: a historical review, *Transportation Research Part F: Traffic Psychology and Behaviour*, v. 2, pp. 181-196, 1999.

[5] Hoogendoorn, S.P.; Bovy, P.H.L. State-of-the-art of vehicular traffic flow modelling. Institution of Mechanical Engineers. Part I: *Journal of Systems and Control Engineering*, Vol. 4, N.. 215.,283-304, 2001.

[6] M. Zamith, R.C.P. Leal-Toledo, E.M. Toledo, G.V.P. Magalhães. A new stochastic cellular automata model for traffic flow simulation with drivers' behavior prediciton, *J. Comput. Science*, 9, 51-57, 2015.

[7] Leal-Toledo, R.C.P., Magalhães, G.V.P., Almeida, I.M., Toledo, E.M.:

Traffic flow simulation under the influence of different acceleration policies. In: *Proceedings of the International Conference on Scientific Computing*, pp. 90–93. CSREA Press, Las Vegas (2017).

[8] Almeida, I.M., Leal-Toledo, R.C.P., Toledo, E.M., Cacau, D.C., Magalhães, G.V.P.: Drivers' behavior effects in the occurrence of dangerous situations which may lead to accidents. In: Mauri, G., El Yacoubi, S., Dennunzio, A., Nishinari, K., Manzoni, L. (eds.) *ACRI 2018. LNCS*, vol. 11115, pp. 441–450. Springer, Cham (2018). [https://doi.org/10.1007/978-3-319-99813-8\\_40](https://doi.org/10.1007/978-3-319-99813-8_40)

[9] Almeida, I.M., Leal-Toledo, R.C.P., Toledo, E.M., Cacau, D.C., Nunes de Lima, R.C., Malta, S.: Influence of Drivers' Behavior on Traffic Flow at Two Roads Intersection, S. Misra et al. (Eds.): *ICCSA 2019, LNCS 11619*, pp. 1–13, 2019. [https://doi.org/10.1007/978-3-030-24289-3\\_11](https://doi.org/10.1007/978-3-030-24289-3_11)

[10] Qu, X., Yang, M., Yang, F., Ran, B., Li, L., An Improved Single-Lane Cellular Automaton Model considering Driver's Radical Feature, *Journal of Advanced Transportation*, v. 2018, ID 3791820, 2018.

[11] An extended continuum model accounting for the driver's timid and aggressive attributions, *Physics Letters A*, April 18, 2017, Vol.381(15), p.1302

[12] The effect of driver's characteristics on the stability of traffic flow under honk environment, *Nonlinear Dynamics*, 2016, Vol.84(3), pp.1517-1528.

[13] Bette, H. M., Habel, L., Emig, T., Schreckenberg, M. Mechanisms of jamming in the Nagel-Schreckenberg model for traffic flow. *Physical Review E*, v, 95, 012311, 2017.

[14] Treiber, M., Kesting, A.: *Traffic Flow Dynamics: Data, Models and Simulation* (2012).

[15] Jordanoska, J., Gjurkov, I., Danev, D.: Comparative Analysis of Car Following Models Based on Driving Strategies Using Simulation Approach.

[16] Nagel, K., Schreckenberg, M.: A cellular automaton model for freeway traffic. *Journal de physique I* 2(12), 2221-2229 (1992).

# **Relatório de atividades**

## **Dados gerais**

### **Identificação de lentes gravitacionais utilizando técnicas de aprendizagem profunda**

Bolsista: Viviane de Mattos Matioli

Orientador: Fabio Porto

Bolsa de iniciação científica (PIBIC) - 1/8/2019 a 31/7/2020

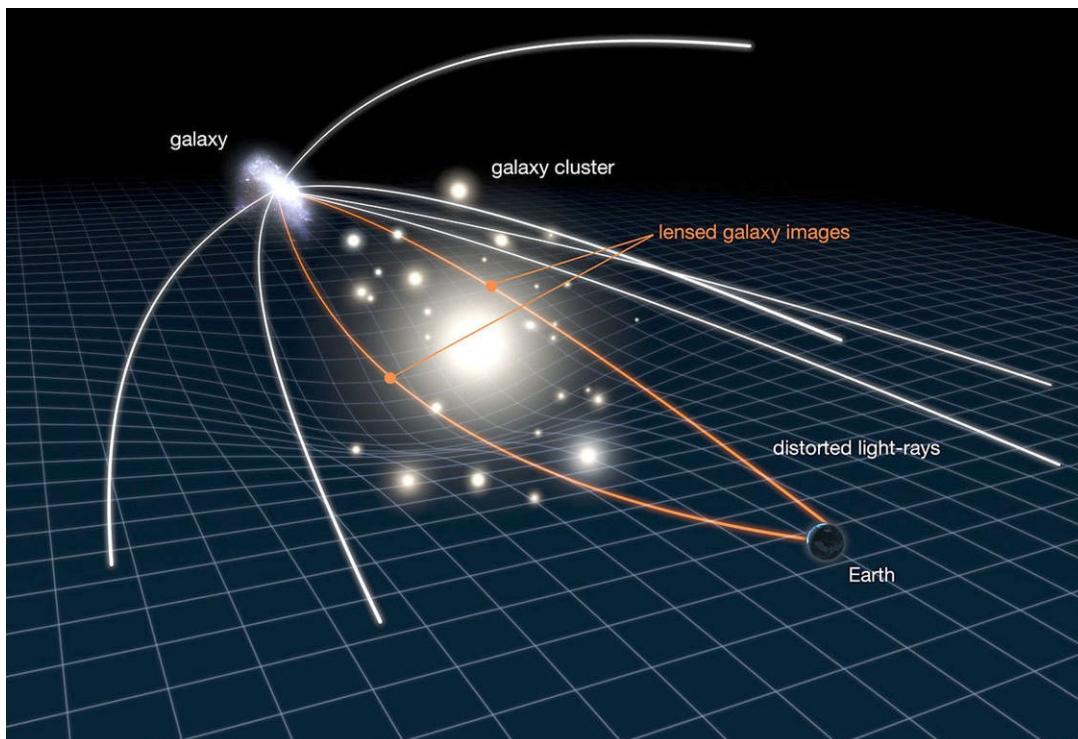
## **Objetivos**

O objetivo do projeto é dar continuidade e auxiliar na pesquisa da área de astronomia computacional desenvolvida no laboratório DEXL, relacionada mais especificamente à detecção de lentes gravitacionais a partir de levantamentos astronômicos. Para isto utilizamos, no último ano, redes neurais artificiais profundas na tentativa de desenvolver um método inicial automatizado para identificação de imagens contendo lentes.

# 1. Introdução

De acordo com a Teoria da Relatividade Geral de Einstein, o efeito observado da gravidade é na verdade causado pela curvatura do espaço-tempo devido a concentrações de massa/energia. Grandes concentrações de massa (como galáxias e aglomerados) fazem com que o espaço-tempo ao seu redor se curve, o que afeta a trajetória de corpos próximos e também da luz.

Assim, quando a luz proveniente de um objeto astronômico (fonte) passa próxima a um outro corpo massivo (lente) antes de chegar até os observadores, ela será desviada. O efeito observado por nós são distorções na imagem da fonte (lenteamento fraco), ou até formação de várias imagens da mesma fonte (lenteamento forte).



*Esquema de uma lente gravitacional em que a luz de uma galáxia fonte é desviada por um aglomerado de galáxias*

Lentes gravitacionais fortes são de grande importância para a astronomia, possuindo diversas aplicações, como estimativa de constantes cosmológicas, determinação da distribuição de massa de galáxias e aglomerados, e estudo de efeitos da matéria escura. [1][2]

Apesar disso, até hoje há um número relativamente baixo de sistemas de lentes gravitacionais conhecidas quando comparado à quantidade crescente de dados astronômicos disponíveis. E com grandes telescópios planejados para os próximos anos, a necessidade de métodos mais eficiente de identificação de lentes se torna cada vez mais evidente.

Tradicionalmente este processo de identificação depende de inspeção visual de imagens. Porém, alguns esforços recentes já se dedicam ao desenvolvimento de métodos mais automatizados, utilizando técnicas de aprendizado de máquina (AM). Neste contexto está inserido nosso projeto.

## 2. Material e métodos

Neste período do projeto, o maior foco foi verificar como a utilização de técnicas de aprendizagem profunda pode nos auxiliar na busca por lentes. Assim primeiramente foi preciso estudar e compreender conceitos importantes de aprendizado de máquina e funcionamento de redes neurais artificiais, em específico redes neurais convolucionais (CNNs) e redes neurais siamesas (SNNs).

Aprendidos os fundamentos de AM, seguiu-se um período de tentativas de desenvolvimento de um algoritmo, utilizando as técnicas vistas, capaz de auxiliar na busca por lentes gravitacionais. Experimentando com diferentes arquiteturas de redes e estruturação do problema, chegamos ao sistema que até então apresentou os melhores resultados : a utilização de uma rede neural siamesa para classificação binária de imagens. Ou seja, desenvolvemos uma SNN que recebe exemplos de imagens contendo lentes gravitacionais, e exemplos de imagens sem lentes, e realiza a tarefa de classificação das imagens em duas classes: lente ou não-lente.

Com a mesma arquitetura da SNN, realizamos experimentos com diferentes conjuntos de imagens, dividindo o processo em duas etapas: Na primeira etapa a rede é treinada e testada com imagens reais de lentes e não-lentes, já na segunda etapa o treinamento da rede é realizado com imagens simuladas, e avaliação em ambos os tipos de imagens, simuladas e reais.

Para isto, utilizamos imagens reais de lentes do tipo galáxia-quasar parte do levantamento CASTLES [4] [5] e lentes do tipo galáxia-galáxia retiradas da base *Master Lens Database* [6], que reúne imagens de lentes de diferentes levantamentos. Como exemplos de não-lentes utilizamos imagens de galáxias observadas com o telescópio Hubble, obtidas através do portal *Hubble Legacy Archive* [7]. Por fim, na segunda etapa utilizamos simulações provenientes de um desafio para identificação de lentes realizado no portal do projeto *Bologna Lens Factory* [8] [9], contendo imagens simuladas de lentes do tipo galáxia-galáxia e de não lentes. Para utilizar todas estas imagens no treinamento e teste da rede, foi preciso também realizar um pré-processamento das mesmas, retirando textos e símbolos localizados sobre elas.

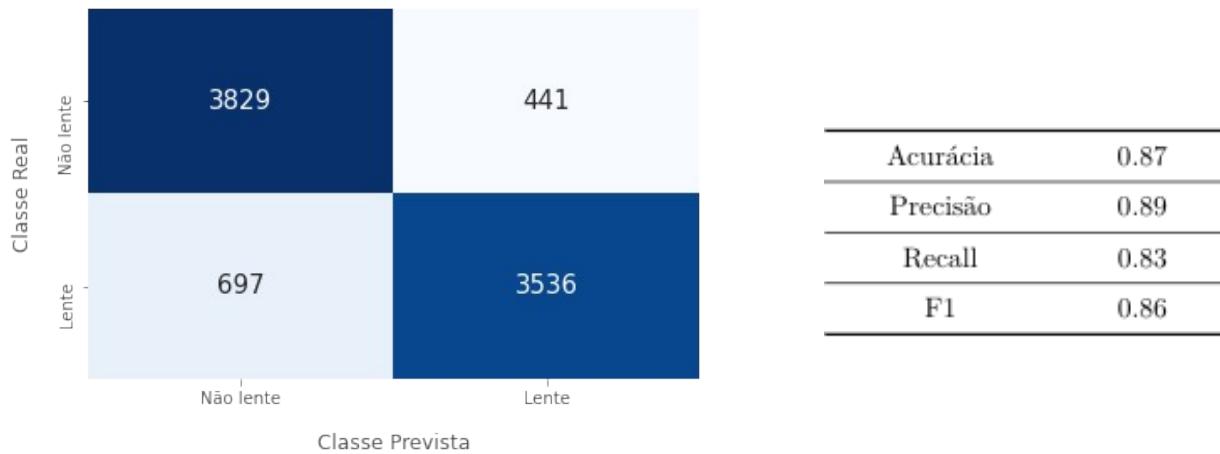
Os resultados dos testes realizados para avaliação da performance da rede na classificação de imagens não vistas no treinamento são apresentados na próxima seção. Este algoritmo foi pensado como um método inicial a ser utilizado para identificação de lentes, já que, após treinado, sua utilização não demanda muito tempo e poder de processamento. Assim ele pode ainda ser integrado a outras técnicas, como o projeto já desenvolvido no grupo DEXL na linha de identificação de lentes - Constellation Queries [3].

## 3. Resultados e Discussão

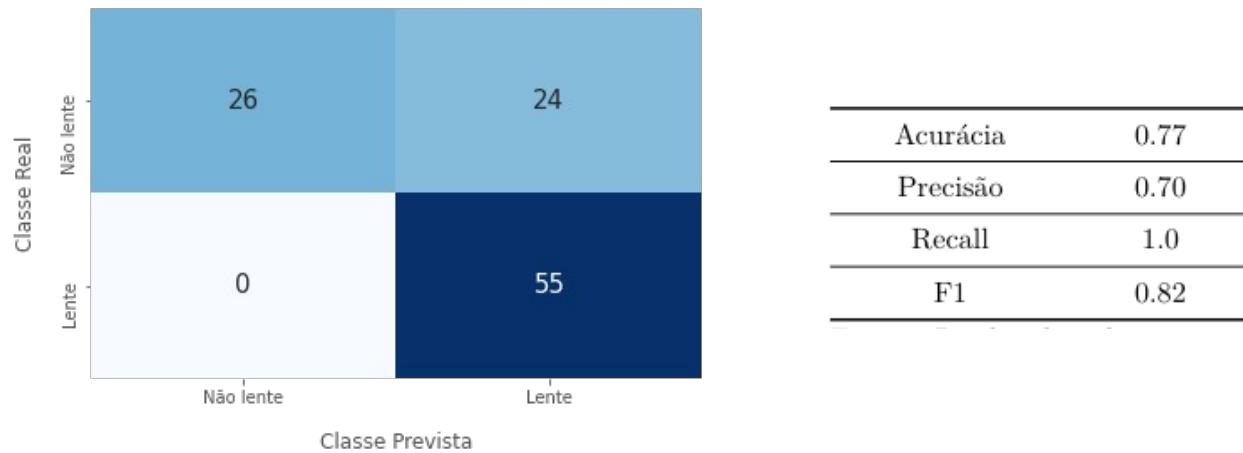
Na primeira etapa, em que treinamos e testamos a rede com imagens reais, os resultados obtidos não foram satisfatórios, como já esperávamos, devido ao número muito baixo de exemplos de treinamento, ou seja, de imagens reais de lentes já conhecidas.

Por isso, na segunda etapa utilizamos imagens simuladas no treinamento, que estão disponíveis em maior quantidade, assim obtemos melhores resultados. Para avaliação do desempenho da rede utilizamos a matriz de confusão e métricas de classificação como acurácia, precisão, *recall* e *F1-score*. Alguns exemplos de resultados obtidos nesta etapa são apresentados a seguir.

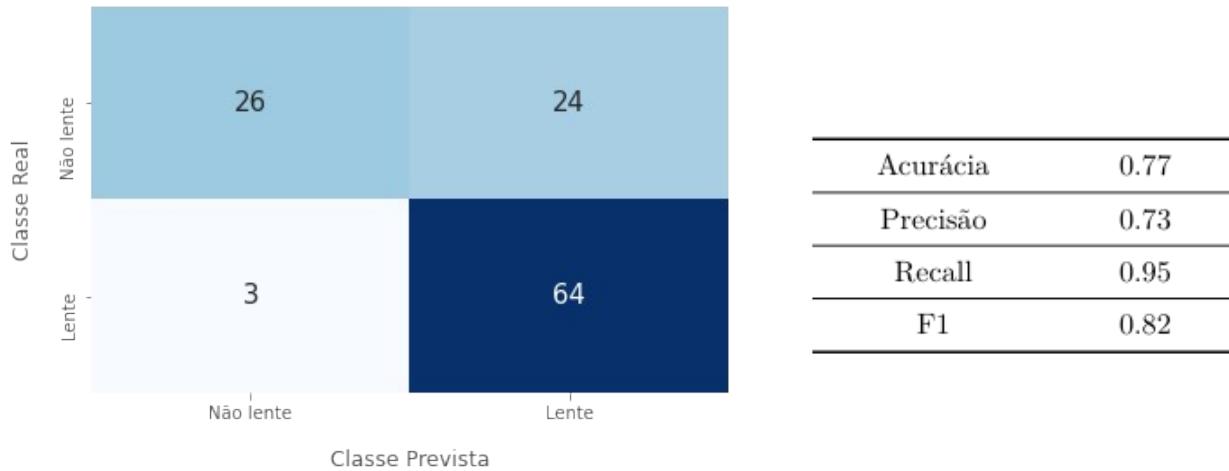
- Matriz de confusão e métricas – teste com imagens simuladas (lentes galáxia-galáxia):



- Teste com imagens reais de lentes galáxia-quasar:



- Teste com imagens reais de lentes galáxia-galáxia:



## 4. Conclusões

O método desenvolvido para auxílio na busca de lentes gravitacionais utilizando redes neurais siamesas apresenta resultados satisfatórios quando treinado com imagens simuladas e testado em imagens também simuladas, ou imagens reais pré-processadas. Nos testes realizados verificamos que ela é capaz de classificar a maior parte (acima de 80%) das lentes de forma correta, porém possui maior dificuldade para a classificação de não lentes.

Assim, pensamos que esta ideia pode seguir sendo aprimorada para sua utilização como método inicial em um projeto de identificação de lentes. Se trata de um método de baixo custo computacional, que após treinado, pode ser utilizado para avaliar imagens de candidatos à lentes, e de acordo com os resultados obtidos até então, eliminar ao menos metade das não-lentes entre eles, sem descartar mais de 20% das lentes, que são a classe de maior interesse.

Alguns exemplos de elementos que ainda podem ser estudados e desenvolvidos são os efeitos de diferentes pré-processamentos em imagens reais antes da classificação pela rede, e testes com diferentes tipos e maior número de objetos não-lentes reais.

## 5. Referências bibliográficas

- [1] Joachim Wambsganss, “Gravitational Lensing in Astronomy”. Living Rev. Relativity, 1, 1998
- [2] P. Schneider, C. Kochanek, J.Wambsganss. Gravitational Lensing: Strong,Weak andMicro, 2006
- [3] Fabio Porto, Amir Khatibi, João N. Rittmeyer, Eduardo Ogasawara, Patrick Valdoriez, Dennis E. Shasha, Constellation Queries in Big Data, SBBD 2018.
- [4] <https://www.cfa.harvard.edu/castles/>
- [5] Falco, EE et al, The CASTLES gravitational lensing tool, 2001
- [6] <http://admin.masterlens.org/index.php>
- [7] <https://hla.stsci.edu/>
- [8] [http://metcalf1.difa.unibo.it/blf-portal/gg\\_challenge.html](http://metcalf1.difa.unibo.it/blf-portal/gg_challenge.html)
- [9] Metcalf, R Benton et al, The strong gravitational lens finding challenge, 2019