

CURSO DE CIÊNCIA DA COMPUTAÇÃO – TCC		
( ) PRÉ-PROJETO	(X) PROJETO	ANO/SEMESTRE: 2023/2

## OTIMIZAÇÃO DE SIMULAÇÕES DE N CORPOS UTILIZANDO INTELIGÊNCIA ARTIFICIAL

Vinícius Manuel Martins

Prof. Aurélio Faustino Hoppe – Orientador

### 1 INTRODUÇÃO

A simulação computacional de sistemas gravitacionais é um campo de estudo fundamental e antigo na física. Segundo Maksimova (2021), este ramo da pesquisa tem como objetivo modelar o comportamento de partículas em sistemas complexos, como galáxias, estrelas, planetas e outras estruturas cósmicas. Portanto, compreender como a gravidade influencia esses sistemas permite aos cientistas observarem eventos novos, possibilitando a exploração de novas teorias (CIAMBUR *et al.*, 2020).

No entanto, a simulação de sistemas com múltiplas partículas é uma tarefa computacionalmente custosa (WANG *et al.*, 2020). Para cada partícula dentro do sistema, é necessário calcular as interações gravitacionais com todas as outras. Conforme destacado por Hampl *et al.* (2022), diversas abordagens matemáticas e interpretações alternativas do problema podem ser empregadas para simplificá-lo. Uma dessas abordagens envolve agrupar as partículas em conjuntos, ou tratar o sistema como um mapa de densidade com vetores de deslocamento (OLIVEIRA *et al.*, 2020).

Segundo Garrison *et al.* (2021), com o avanço da computação, houve um progresso significativo nesse campo, possibilitando a realização de simulações em larga escala. No entanto, algumas dessas melhorias baseiam-se em aproximações dos cálculos, o que pode introduzir imprecisões (HERNANDEZ, 2019). Essas imprecisões, de acordo com Gonzalez *et al.* (2020) podem alterar sutilmente as condições iniciais e transformar um sistema que deveria ser determinístico em algo imprevisível a longo prazo, um fenômeno conhecido como caos.

Ainda segundo Garrison *et al.* (2021), a partir deste cenário desafiador e em constante evolução, torna-se necessário a explorar estratégias de otimização para simulações computacionais de sistemas de N partículas, que representam uma classe crucial de sistemas físicos. Tais sistemas são caracterizados pelo estudo do movimento de N partículas que interagem mutuamente por meio da força gravitacional. Além disso, os autores também destacam que a complexidade computacional associada ao cálculo das interações entre todas as partículas em sistemas de N corpos aumenta exponencialmente com o número de partículas, tornando essas simulações demoradas e requerendo bastante recursos computacionais.

Neste contexto, este trabalho tem a seguinte pergunta de pesquisa: qual é a eficácia das abordagens de otimização, incluindo heurísticas e técnicas baseadas em inteligência artificial, na melhoria da eficiência computacional e da precisão das simulações de sistemas gravitacionais de N partículas, considerando a complexidade intrínseca desses sistemas e a influência do caos em suas dinâmicas?

#### 1.1 OBJETIVOS

Este trabalho tem como objetivo a criação de um artefato computacional para a simulação de sistemas gravitacionais com N partículas.

Os objetivos específicos são:

- investigar o uso de heurísticas que possam aprimorar tanto a eficiência quanto a precisão das simulações, permitindo a análise de sistemas cada vez mais complexos e numerosos;
- desenvolver uma abordagem baseada em inteligência artificial capaz de prever os movimentos das partículas em simulações de sistemas gravitacionais, mesmo diante da natureza caótica inerente desses sistemas;
- avaliar a eficiência e a precisão das técnicas de otimização aplicadas para simulações de sistemas gravitacionais, a fim de comparar sua eficácia em relação a outros modelos e abordagens;

### 2 TRABALHOS CORRELATOS

Nesta seção serão apresentados os trabalhos que correlacionam aos objetivos deste trabalho. Para realizar a busca, foi utilizada a plataforma Google Scholar. Os termos de pesquisa utilizados foram: *n-body simulation*, *three body problem*, *machine learning*, *optimization* e *graph network*. A busca combinada dessas palavras-chave retornou cerca de 160.000 registros, sendo limitado apenas a artigos criados após 2019. A escolha deles foi feita com base em sua relevância, uma medida do site que leva em conta número de citações, autores e locais de publicação.

Por fim, foram selecionados 4 artigos. A seção 2.1 descreve o uso de redes neurais profundas para o problema de N corpos, explorando seu comportamento em um ambiente caótico (BREEN *et al.*, 2020). Na seção 2.2 são descritos métodos de otimizações e heurísticas utilizados em simuladores (GARRISON *et al.*, 2021). A seção 2.3 aborda o tema por meio de redes neurais orientadas a grafos (GONZALEZ *et al.*, 2020). Por fim, na seção 2.4 será apresentado o desenvolvimento de uma rede neural profunda aplicada a conceitos de mapa de densidade e vetores de deslocamento (OLIVEIRA *et al.*, 2020).

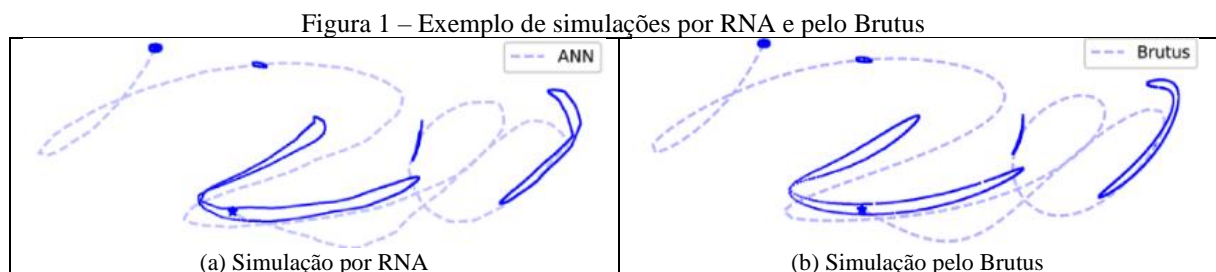
## 2.1 NEWTON VERSUS THE MACHINE: SOLVING THE CHAOTIC THREE-BODY PROBLEM USING DEEP NEURAL NETWORKS

De acordo com Breen *et al.* (2020), simulações que envolvem três corpos ou mais possuem uma natureza caótica. Dessa forma, pequenas imprecisões nos cálculos podem resultar em resultados completamente distintos. Do ponto de vista computacional, esses problemas apresentam novos desafios, como o uso intensivo de memória e *time-steps* que se aproximam de zero.

Para superar essa questão, Breen *et al.* (2020) propuseram o uso de uma Rede Neural Artificial (RNA) que fosse capaz de prever a trajetória de partículas durante encontros próximos. Eles construíram uma base de dados com 10.000 registros usando Brutus, um integrador numérico para problemas de N corpos. Por uma questão de simplicidade, segundo os autores, essa base de dados foi limitada a problemas envolvendo três corpos com massas iguais e sem velocidades iniciais.

Segundo Breen *et al.* (2020), após treinar a RNA por 1.000 épocas, utilizando um *time-step* de 3.9, obteve-se uma taxa de energia relativa com Mean Absolute Error (MAE) inferior a 0.1%. Apesar disso, em momentos de encontros próximos, no qual a energia é altamente sensível à posição, os erros podem chegar a uma média de 10%. Esse problema foi minimizado com a introdução de uma camada de projeção ao modelo. Essa camada, busca preservar a energia do sistema realizando ajustes compensatórios. Com isso, de acordo com Breen *et al.* (2020) conseguiu-se reduzir o MAE para 0.00001%.

A Figura 1 mostra a evolução temporal de uma simulação que não estava incluída nos dados de teste, realizada pela RNA (item a) e pelo Brutus (item b). Os trechos em azul escuro referem-se à trajetória realizada pelos dois métodos em cinco espaços de tempo. Pode-se notar que a RNA conseguiu reproduzir o comportamento caótico desejado por Breen *et al.* (2020).



Fonte: Breen *et al.* (2020).

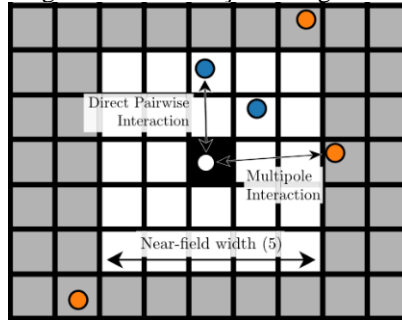
Como resultado, Breen *et al.* (2020) destacam o uso da RNA junto com integradores numéricos. Sugerindo o uso da RNA para realizar as previsões e em seguida usar essa saída como entrada para os integradores tradicionais. Isso permite, segundo os autores, eliminar gargalos ao mesmo tempo que mantém um alto nível de precisão. Além disso, Breen *et al.* (2020) ressaltam que o estudo explorou a resolução do problema eliminando algumas limitações, como a igualdade de massas entre os corpos, resultando em uma solução mais generalista para problemas de três corpos.

## 2.2 THE ABACUS COSMOLOGICAL N-BODY CODE

Garrison *et al.* (2021) desenvolveram o Abacus, um código de alta precisão e performance para a simulação cosmológica de N corpos. Segundo os autores, utilizando diferentes métodos matemáticos, em uma arquitetura orientada a eventos otimizada para o uso de unidades de processamento gráficas (GPUs), é possível realizar o processamento de milhões de partículas com uma perda na ordem de  $1.6 \times 10^{-4}$ . Além disso, Garrison *et al.* (2021) também implementaram estratégias de gerenciamento de memória, como o agrupamento das partículas em *clusters* e o uso otimizado do disco. Segundo os autores, essas melhorias foram projetadas para que pudessem ser executadas em *hardware* mais acessível.

A Figura 2 demonstra o cenário de otimização processado pelo algoritmo. Nele, a força resultante em uma partícula é determinada por duas componentes, interações de campo próximo e campo distante. Segundo Garrison *et al.* (2021), interações de campo próximo são mais custosas, mas valores mais precisos são cruciais para o resultado da simulação. Em contrapartida, as interações de campo distante são calculadas por meio de aproximações, visto que imprecisões nesse cenário não representam muita perda na solução final.

Figura 2 – Otimização no algoritmo



Fonte: Garrison *et al.* (2021).

Como resultado, Garrison *et al.* (2021) demonstraram o desempenho do Abacus em uma simulação que utilizou 44 núcleos de unidade central de processamento (Central Processing Unit - CPU) e 6 GPUs NVIDIA V100, alcançando o processamento de 60 trilhões de partículas, em 97 diferentes condições iniciais, a uma taxa de processamento superior a 70 milhões de partículas por segundo.

Por fim, Garrison *et al.* (2021) destacam que as evoluções no *hardware* de computadores são feitas buscando resolver problemas em diversos níveis (aprendizado de máquina, renderizações gráficas e análises de dados). Além disso, segundo os autores, o Abacus fez isso utilizando mudanças na arquitetura do código em alto nível, combinada como otimizações matemáticas, o uso personalizado da memória possibilitando uma *pipeline* orientada a eventos, entre outras coisas. Para Garrison *et al.* (2021), esse tipo de abordagem, torna a solução encontrada escalável para computadores do futuro, dado a sua característica altamente paralelística.

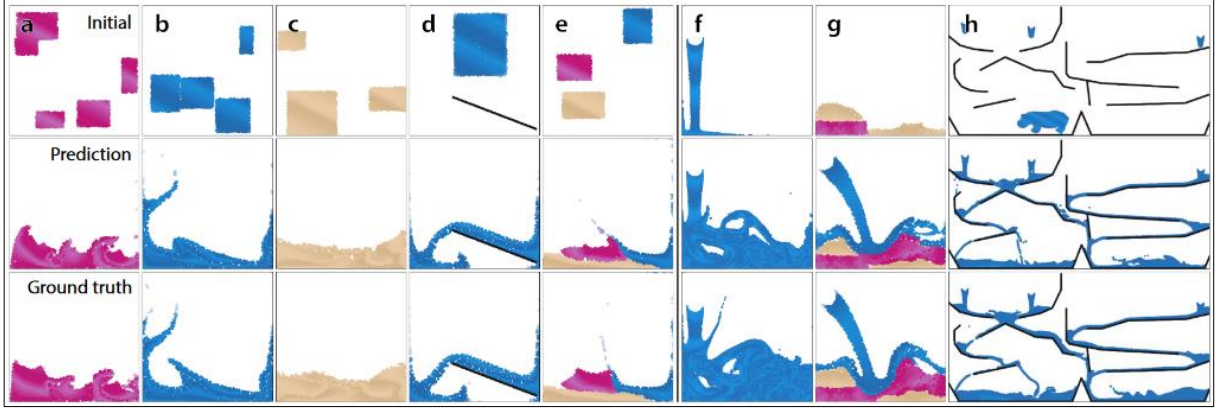
### 2.3 LEARNING TO SIMULATE COMPLEX PHYSICS WITH GRAPH NETWORKS

Gonzalez *et al.* (2020), desenvolveram um *framework* de machine learning orientado a grafos que denominaram “Simuladores orientados a redes de grafos”. Segundo os autores, ele é composto por três partes principais: (i) o *encoder*, (ii) o *processor* e (iii) o *decoder*. O *encoder*, cria um grafo no qual cada nó é uma partícula do sistema. As conexões do nó são criadas entre partículas que estejam dentro de um “raio de conectividade” e a cada *time-step*, essas ligações são recalculadas utilizando o algoritmo do vizinho mais próximo. O peso de cada aresta é a distância relativa entre os dois nós. Já o *processor*, realiza o processamento das interações entre os nós. Ele é composto por M camadas, em que para cada uma delas ocorre a passagem de mensagens entre os nós. Além disso, são utilizadas conexões residuais, permitindo que a saída de uma camada possa ser a entrada de outra, pulando algumas camadas intermediárias. Por fim, o *decoder*, é uma *perceptron* multicamadas que extrai a aceleração das partículas a partir do grafo resultante. Com a aceleração, é possível realizar a integração de Euler para extrair a velocidade e posição das partículas e enfim atualizar o sistema.

Para o treinamento, Gonzalez *et al.* (2020) utilizaram partículas com diferentes propriedades, como água, areia e gosma, e exploraram as interações entre elas. O *dataset* foi gerado utilizando ferramentas e simuladores já existentes, como o BOXBATH, SPlisHSPlasH e o Taichi-MPM. Com isso, Gonzalez *et al.* (2020) notaram que alguns hiperparâmetros impactavam diretamente no resultado, tais como: (i) o número de trocas de mensagens, (ii) o número de conexões (arestas do grafo) e (iii) o ruído inicial. Esse último, os autores destacam que foi uma estratégia adotada na etapa de treinamento para mitigar a acumulação de erros nos cálculos no momento da validação.

Na Figura 3 apresenta os resultados do estudo de forma visual. Cada coluna representa um cenário de validação (não incluídos na fase de treinamento), sendo que na primeira linha estão as condições iniciais, na segunda estão as previsões da Graph Networks Simulate (GNS) e na última a base de verdade. As partículas rosas são gosma, as azuis são água e as beges são areia. Os obstáculos em preto se referem a matéria sólidos estáticos. Pode-se notar, que o modelo conseguiu entender bem a dinâmica entre os diversos materiais, conseguindo realizar previsões plausíveis a longo prazo.

Figura 3 – Exemplo de previsão do GNS



Fonte: Gonzalez *et al.* (2020).

Como resultado, Gonzalez *et al.* (2020) obtiveram simulações com o Mean Squared Error (MSE) na ordem de  $10^{-3}$ , variando de acordo com a escolha dos materiais, sua interação, etc. Com isso, os autores constataram que o modelo conseguiu generalizar bem o problema e entender a natureza dos materiais, além dos comportamentos físicos envolvidos. Em comparação com outros modelos, o GNS se mostrou mais simples e exato, com espaço para otimizações com o uso de computação paralela.

## 2.4 FAST AND ACCURATE NON-LINEAR PREDICTIONS OF UNIVERSES WITH DEEP LEARNING

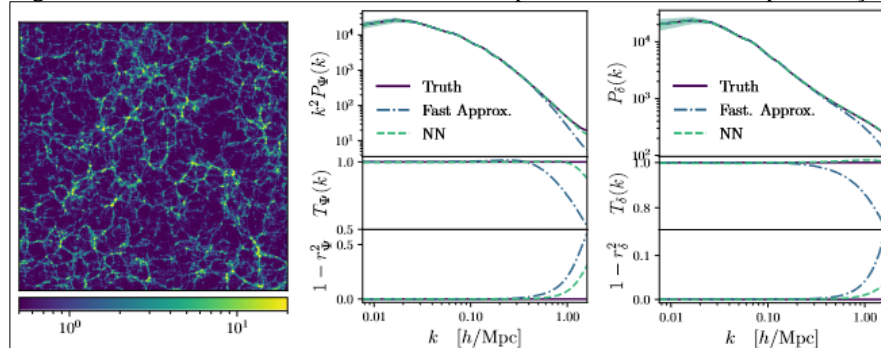
Oliveira *et al.* (2020) abordaram o problema das aproximações em simulações de N corpos em escalas menores. Nestes cenários, segundo os autores, a flutuação da densidade das partículas geralmente se torna significativa, de forma que pequenas imprecisões podem invalidar as previsões.

Em busca de alternativas para enfrentar esse desafio, Oliveira *et al.* (2020) desenvolveram uma *Convolutional Neural Network* (CNN) com a arquitetura V-NET que recebe como entrada os vetores de deslocamento das partículas e retorna o vetor de deslocamento resultante. Segundo os autores, a arquitetura trabalha com três resoluções diferentes do problema, com duas camadas de *downsampling* e duas de *upsampling*. Além disso, são utilizadas conexões residuais em cada uma das camadas de convolução.

De acordo com Oliveira *et al.* (2020), para o treinamento do modelo, foram utilizadas 210 simulações de N corpos do projeto Quijote. Dessa amostragem, 180 foram utilizados para treino, 20 para validação e 10 para teste. Cada simulação possui cerca de  $512^3$  partículas dispostas dentro de um cubo com volume de 1 (Gpc/h)<sup>3</sup>. Cada um dos universos simulados emprega 4 conjuntos distintos de parâmetros cosmológicos, sendo que um deles se assemelha muito às condições do nosso próprio universo.

Oliveira *et al.* (2020) utilizaram a função de perda MSE, ao qual foi aplicada tanto ao vetor resultante quanto à quantidade de partículas em cada *voxel*, que representa uma célula do *grid* no domínio da simulação. Segundo os autores, o modelo obteve uma acurácia superior aos métodos de aproximação tradicionais, com mais de 99% de acurácia e um tempo de treinamento duas vezes mais rápido. A Figura 4 apresenta os resultados obtidos. O painel à esquerda se refere ao campo de densidade esperado, conforme simulado pelo projeto Quijote. Os gráficos do meio e da direita, se referem aos resultados da rede neural considerando respectivamente, o vetor resultante e a densidade de partículas nos *voxels*. A partir da Figura 4, pode-se observar que conforme  $k$  (flutuações na densidade) aumenta, os métodos de aproximação tradicionais falham em prever os valores, em contraste com o modelo proposto por Oliveira *et al.* (2020).

Figura 4 – Gráficos dos resultados obtidos comparando a métodos de aproximação



Fonte: Oliveira *et al.* (2020).

Como conclusão, Oliveira *et al.* (2020) ressaltam que o estudo abre caminho para um maior aproveitamento científico de futuras missões astronômicas. Além disso, afirmam que tais melhorias podem

possibilitar futuros cientistas realizarem simulações complexas do universo, com alta precisão a partir do seu computador pessoal.

### 3 PROPOSTA DO ARTEFATO COMPUTACIONAL

Nesta seção será descrita a justificativa do estudo, definindo os requisitos funcionais e não funcionais e as metodologias abordadas.

#### 3.1 JUSTIFICATIVA

O Quadro 1 apresenta uma comparação dos quatro trabalhos correlatos selecionados, destacando suas principais características. A disposição do quadro é a seguinte: cada linha representa uma característica distinta e cada coluna um trabalho correlato.

Quadro 1 – Comparativo entre os trabalhos correlatos

Trabalhos Correlatos Características	Breen <i>et al.</i> (2020)	Garrison <i>et al.</i> (2021)	Gonzalez <i>et al.</i> (2020)	Oliveira <i>et al.</i> (2020)
Natureza do problema	Simulações de três corpos com natureza caótica	Simulação cosmológica de N corpos	Simulações de interação de partículas	Simulações de N corpos em escalas gravitacionais menores
Estratégia de otimização	Uso de uma RNA para prever as trajetórias	Otimizações no algoritmo, estratégias de paralelismo e métodos matemáticos	Rede neural orientada a grafos	Uso de uma CNN com arquitetura V-Net
Base utilizada	10.000 registros gerados pelo Brutus	Não se aplica	Dados gerados com diferentes ferramentas: BOXBATH, SPLisHSPlasH e Taichi-MPM	Dados obtidos a partir das simulações do projeto Quijote
Métricas de desempenho	MAE inferior a 0.1%	Perca na ordem de $1.6 \times 10^{-4}$	MSE na ordem de $10^{-3}$	Mais de 99% de acurácia
Resultados	RNA reproduziu natureza caótica de forma precisa	Processamento de 70 milhões de partículas por segundo	Previsões precisas e uma solução mais simples de se trabalhar	Modelo mais preciso e rápido que métodos tradicionais

Fonte: elaborado pelo autor.

A partir do Quadro 1, pode-se notar que todos os trabalhos enfrentam desafios computacionais significativos relacionados à simulação de sistemas de N partículas. Breen *et al.* (2020), Garrison *et al.* (2021) e Oliveira *et al.* (2020) focam em sistemas de N corpos gravitacionais, abordando o problema em diferentes escalas. Por outro lado, Gonzalez *et al.* (2020) exploram a simulação de partículas interagindo, considerando suas diferentes propriedades. Todos os problemas possuem complexidades intrínsecas e buscam explorar características únicas do tema.

Quanto às estratégias de otimização, os trabalhos divergem em suas abordagens. Breen *et al.* (2020) propõem o uso de uma RNA para prever trajetórias em simulações de três corpos. Garrison *et al.* (2021) desenvolvem o código Abacus, buscando uma maior otimização para GPUs, visando simulações cosmológicas em grande escala. Gonzalez *et al.* (2020) introduzem o GNS, para prever interações de partículas. Oliveira *et al.* (2020) utilizam uma CNN com arquitetura V-NET em escalas menores.

Em relação aos resultados, todos os trabalhos alcançaram melhorias significativas. Breen *et al.* (2020) conseguiram simular a natureza caótica do sistema com sua RNA, enquanto Garrison *et al.* (2021) conseguiram alto desempenho com o Abacus em simulações cosmológicas em larga escala. Gonzalez *et al.* (2020) demonstraram previsões precisas das interações de partículas com o GNS. Oliveira *et al.* (2020) destacaram a precisão e velocidade superior de sua CNN em comparação com métodos tradicionais.

Com isso, pode-se observar que os quatro trabalhos abordam desafios de simulações de N partículas de maneiras distintas, utilizando estratégias de otimização variadas e obtendo melhorias consideráveis em eficiência e precisão. Cada abordagem contribui para o avanço da pesquisa nesse campo, destacando a versatilidade de técnicas computacionais e de aprendizado de máquina para lidar com problemas complexos de física e cosmologia.

Neste trabalho serão exploradas diferentes técnicas e abordagens para melhorar a eficiência das simulações de sistemas de N corpos. Isso pode envolver o uso de algoritmos de aproximação que reduzem o número de interações a serem calculadas, o desenvolvimento de técnicas de paralelização para acelerar o processamento em sistemas de alto desempenho, ou até mesmo a aplicação de inteligência artificial para prever o comportamento das partículas de forma mais eficiente. Além disso, será avaliada a precisão das simulações após a otimização, uma vez que é importante garantir que as simplificações ou aproximações não comprometam a qualidade dos resultados. O trabalho também se concentrará em criar visualizações e ferramentas que tornem as simulações mais acessíveis e informativas, facilitando a interpretação dos resultados e a tomada de decisões em contextos científicos e de engenharia. Por fim, este trabalho busca contribuir para o avanço no campo das simulações de sistemas de N corpos, explorando estratégias de otimização que permitam simulações mais rápidas e eficientes, mantendo a precisão dos resultados. Isso tem implicações significativas em diversas áreas do conhecimento, tais como Física e Astronomia, e pode abrir portas para uma compreensão mais profunda de sistemas complexos regidos pela gravidade.

### 3.2 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

O artefato computacional a ser desenvolvido deverá:

- a) processar uma simulação utilizando forças gravitacionais dado as configurações iniciais para as partículas, incluindo posições, massas, velocidades iniciais, *time-step* e outras propriedades relevantes (Requisito Funcional - RF);
- b) aplicar soluções baseadas em redes neurais e outras heurísticas de otimização (RF);
- c) incluir estratégias de otimização para reduzir o tempo de processamento e minimizar a demanda de recursos computacionais, especialmente em simulações em larga escala (RF);
- d) disponibilizar métricas de execução após a execução de cada teste (como tempo de execução e consumo de memória) (RF);
- e) gerar visualizações gráficas das simulações, facilitando a compreensão dos resultados e permitindo a análise visual (RF);
- f) ser implementada utilizando as linguagens Go e Python (Requisitos Não Funcional - RNF);
- g) deve ser capaz de lidar com simulações de sistemas de N partículas com eficiência, proporcionando tempos de execução razoáveis mesmo para simulações complexas (RNF).

### 3.3 METODOLOGIA

O trabalho será desenvolvido observando as seguintes etapas:

- a) revisão bibliográfica: busca de conceitos gerais de simulação computacional, aprendizado de máquina, heurísticas de otimização e trabalhos correlatos;
- b) levantamento dos requisitos: baseando-se nas informações da etapa anterior, reavaliar os requisitos propostos para a aplicação;
- c) definição das ferramentas para modelagem e armazenamento das simulações: pesquisar e escolher as ferramentas mais apropriadas para a modelagem e armazenamento das simulações;
- d) definição das técnicas de otimização: pesquisar e (re)definir as técnicas/algoritmos que serão utilizadas na otimização da simulação de N partículas;
- e) definição do cenário de simulação: a partir do item (d) estabelecer o cenário de simulação, a escala do sistema (N) e os níveis de precisão;
- f) construção da base de dados: utilizar integradores numéricos para gerar uma base de dados e utilizar bases de dados já existentes, como a do projeto Quijote;
- g) implementação do simulador: desenvolvimento de uma simulação com a solução mais simples (menos ótima) utilizando Go, exportando os dados simulados para um arquivo;
- h) implementação da visualização: desenvolvimento de uma visualização dessas simulações utilizando Python e a biblioteca Pygame;
- i) otimizações no simulador: explorar estratégias que possam aumentar a capacidade de processamento do simulador, como aproximações e heurísticas;
- j) implementação do modelo de inteligência artificial: desenvolver um modelo em Python capaz de prever as forças envolvidas em uma simulação de N corpos, levando em consideração seu comportamento caótico;
- k) validação dos resultados: realizar uma comparação dos resultados obtidos a partir da solução mais simples, com otimizações e a prevista pelo modelo de inteligência artificial. Além disso, apontar a eficiência (medir o tempo de execução e o consumo de recursos), precisão (utilizar métricas de erro), pontos positivos e negativos das soluções pesquisadas (observar o comportamento do artefato em diferentes situações), realizando um comparativo com bases de dados já estabelecidas.



As etapas serão realizadas nos períodos relacionados no Quadro 2.

Quadro 2 – Cronograma de atividades a serem realizadas

etapas / quinzenas	2024									
	fev.		mar.		abr.		maio		jun.	
	1	2	1	2	1	2	1	2	1	2
revisão bibliográfica										
levantamento dos requisitos										
definição das ferramentas para modelagem e armazenamento das simulações										
definição das técnicas de otimização										
definição do cenário de simulação										
construção da base de dados										
implementação do simulador										
implementação da visualização										
otimizações no simulador										
implementação do modelo de inteligência artificial										
validação dos resultados										

Fonte: elaborado pelo autor.

## 4 REVISÃO BIBLIOGRÁFICA

Esta seção tem como objetivo explorar os principais assuntos que fundamentarão o estudo a ser realizado. A seção 4.1 aborda o conceito de simulação de N corpos. A seção 4.2, descreve heurísticas de otimização. Por fim, a seção 4.3 disserta sobre aprendizado de máquina.

### 4.1 SIMULAÇÃO DE N CORPOS

De acordo com Maksimova (2021), a simulação de N corpos emerge como uma técnica computacional essencial para modelar o movimento e a interação de N partículas em sistemas regidos por forças específicas. Amplamente aplicada em campos como física, biologia computacional e dinâmica de fluidos, essa abordagem computacional desempenha um papel crucial na análise de fenômenos complexos (RUCCI *et al.*, 2020).

No âmbito da física, conforme destacado por Breen *et al.* (2020), a simulação de N corpos é frequentemente empregada no estudo do comportamento de sistemas gravitacionais, no qual as partículas representam corpos celestes, como planetas, estrelas ou galáxias. As leis fundamentais da física, como as leis de Newton, são aplicadas para calcular as forças entre as partículas, permitindo a determinação de suas acelerações, velocidades e posições ao longo do tempo (OLIVEIRA *et al.*, 2020).

A complexidade computacional das simulações de N corpos aumenta substancialmente com o número de partículas envolvidas, resultando em um problema de complexidade  $O(n^2)$  (WANG *et al.*, 2020). Hampl *et al.* (2022) indicam que diversas técnicas e algoritmos otimizados foram desenvolvidos para enfrentar esse desafio computacional, destacando (i) o método de Barnes-Hut e (ii) o método Multipolo Rápido (*Fast Multipole Method* - FMM).

O método de Barnes-Hut, introduzido por Barnes e Hut (1986), utiliza uma estrutura de *quad-tree* para agrupar partículas distantes, tratando-as como uma única partícula de massa equivalente. Essa abordagem reduz a complexidade da simulação de N corpos para  $O(n \log n)$ , melhorando a eficiência computacional. Em contrapartida, o FMM é um algoritmo especializado em lidar com interações de longo alcance, utilizando aproximações por meio de métodos multipolos (HAMPL, 2022). Inicialmente, apresentava complexidade  $O(n \log n)$ , mas, após as contribuições de Dehnen (2014), alcançou uma complexidade  $O(n)$ , embora fosse inicialmente limitado a simulações sem colisões entre as partículas.

Além da física, as simulações de N corpos têm aplicações na biologia computacional, permitindo modelar a interação entre moléculas em sistemas biológicos complexos (JEWETT *et al.*, 2021). Segundo Zaporozhets e Clementi (2023), essas simulações proporcionam uma compreensão mais profunda de fenômenos como o dobramento de proteínas e interações moleculares.

Mukherjee *et al.* (2021), ressaltam que, apesar de sua utilidade, as simulações de N corpos geralmente incorporam simplificações e aproximações para viabilizar os cálculos computacionais. Para Hernandez (2019), tais simulações são sensíveis às condições iniciais e à precisão numérica, com pequenas variações podendo resultar em diferenças significativas ao longo do tempo, caracterizando o fenômeno como caótico.

Neste contexto, segundo Ciambur *et al.* (2020), as simulações de N corpos destacam-se como uma ferramenta poderosa para modelar e compreender sistemas complexos em diversas áreas, proporcionando *insights* valiosos sobre o comportamento dinâmico de partículas em interação. Tais simulações viabilizam a exploração de fenômenos raros ou complexos, muitas vezes inacessíveis por meio de análises puramente teóricas. Conforme

Mukherjee *et al.* (2021), além dos métodos tradicionais discutidos para simulações de N corpos, como o método de Barnes-Hut e técnicas de integração temporal, a aplicação de algoritmos metaheurísticos também se destaca na otimização e aprimoramento do desempenho dessas simulações.

#### 4.2 HEURÍSTICAS DE OTIMIZAÇÃO

Segundo Drake *et al.* (2020), as heurísticas de otimização representam uma categoria crucial de métodos e estratégias utilizados na resolução de problemas complexos, especialmente aqueles nos quais a busca por soluções ótimas se mostra desafiadora. Ao contrário dos algoritmos exatos, que garantem a obtenção da solução ótima, as heurísticas buscam encontrar soluções suficientemente boas em um intervalo de tempo razoável (WU *et al.* 2022). Ao empregar heurísticas para abordar um problema, na prática, está se lidando com aproximações e simplificações, visando atingir um equilíbrio ideal entre eficiência e precisão (KARIMI-MAMAGHAN *et al.*, 2022).

De acordo com Parpart *et al.* (2018), a simplicidade emerge como uma característica distintiva dessas técnicas, tornando-as acessíveis mesmo em contextos nos quais a formulação matemática exata do problema se revela desafiadora. Adicionalmente, as heurísticas ajustam dinamicamente suas estratégias, adaptando-se com base nas informações adquiridas ao longo do processo (ZHANG *et al.*, 2020). Sua capacidade diante da incerteza lhe traz aplicações relevantes em cenários do mundo real, nos quais a disponibilidade de dados confiáveis nem sempre é garantida, e a escala dos casos frequentemente limita as soluções (ZHU *et al.*, 2019).

Arnold e Sörensen (2018), pontuam que existem diversos tipos de heurísticas. A busca local, por exemplo, explora iterativamente vizinhanças de soluções existentes em busca de melhorias locais. Os algoritmos genéticos, fundamentados na teoria da Seleção Natural de Darwin, combinam soluções para gerar "indivíduos" mais eficientes. Já os algoritmos de colônia de formigas modelam o comportamento desses insetos para encontrar caminhos eficientes em problemas de otimização de roteamento. O *simulated annealing*, inspirado no processo de recozimento de metais, é outra abordagem que introduz uma probabilidade de aceitar temporariamente soluções piores, permitindo escapar de mínimos locais.

No contexto prático, as heurísticas de otimização encontram aplicação em uma ampla variedade de domínios, incluindo logística, programação, design, finanças e inteligência artificial (ZHU *et al.*, 2019). Sua flexibilidade e eficiência as tornam úteis em situações nas quais métodos exatos se mostram impraticáveis devido à complexidade computacional (WU *et al.* 2022). Entretanto, os desafios incluem a seleção apropriada de heurísticas para diferentes cenários, bem como uma parametrização eficaz que favoreça um desempenho superior em ambientes dinâmicos e altamente interativos (KARIMI-MAMAGHAN *et al.*, 2022).

Na área de Inteligência Artificial (IA), conforme Zounemat-Kermani e Mahdavi-Meymand (2018), a convergência entre heurísticas de otimização e algoritmos metaheurísticos é evidente, destacando a versatilidade dessas abordagens em lidar com problemas complexos. Assim como nas aplicações práticas mencionadas, a inteligência artificial (IA) se beneficia significativamente da flexibilidade e eficiência das heurísticas em ambientes dinâmicos (WU; MENG, 2020). Os desafios enfrentados na seleção de heurísticas e na parametrização eficaz também ecoam no campo do aprendizado de máquina, no qual a escolha de algoritmos e a sintonia de hiperparâmetros são cruciais para o desempenho otimizado dos modelos. Algoritmos genéticos, por exemplo, encontram aplicação na otimização de arquiteturas de redes neurais, enquanto otimização por enxame de partículas é empregada na busca por parâmetros ideais (ZOUNEMAT-KERMANI; MAHDAVI-MEYMAND, 2018). A sinergia entre heurísticas e aprendizado de máquina ressalta não apenas a aplicabilidade dessas técnicas em uma ampla gama de domínios, mas também a interseção crescente entre métodos computacionais e inteligência artificial na busca por soluções eficazes e adaptáveis como as de simulações de N corpos (WALCZAK, 2019).

#### 4.3 APRENDIZADO DE MÁQUINA

De acordo com Wu e Meng (2020), o Aprendizado de Máquina (*Machine Learning* - ML) é um campo da inteligência artificial concentrado no desenvolvimento de algoritmos que permitem o aprendizado por meio da análise estatística de conjuntos de dados. Ao invés de seguir instruções programadas de maneira imperativa, os sistemas de ML são projetados para identificar padrões e tomar decisões baseadas nessas informações (BI *et al.*, 2019).

Segundo Alzubi *et al.* (2018), o processo de aprendizado de máquina pode ser categorizado em três tipos principais: (i) aprendizado supervisionado, (ii) aprendizado não supervisionado e (iii) aprendizado por reforço, cada um com algoritmos, casos de uso e técnicas específicas para resolver diferentes problemas. Conforme Greener *et al.* (2021), no aprendizado supervisionado, o algoritmo é treinado com dados rotulados, no qual a saída desejada é conhecida. O modelo aprende a mapear entradas para saídas corretas e pode fazer previsões ou tomar decisões ao ser alimentado com novos dados. Os autores também afirmam que é um método comum em problemas de classificação e regressão, utilizando algoritmos como *K-Nearest Neighbors* (KNN), *Decision Tree*, *Random Forest*, *Naive Bayes* e *Support Vector Machine* (SVM).



Por outro lado, de acordo com Usama *et al.* (2019), o aprendizado não supervisionado envolve conjuntos de dados não rotulados, exigindo que o algoritmo identifique padrões e estruturas por conta própria. Isso é frequentemente utilizado em tarefas de agrupamento, em que o objetivo é encontrar similaridades entre dados e organizá-los em grupos ou clusters, utilizando algoritmos como *K-Means* e *Gaussian Mixture* (USAMA *et al.*, 2019). Por fim, Mousavi *et al.* (2018) estabelecem que no aprendizado por reforço um agente interage com um ambiente e aprende a realizar ações guiado por um sistema de recompensa. Segundo eles, esse método é aplicado em jogos e robótica, utilizando algoritmos como *Q-Learning* e *Deep Q-Networks*.

Além disso, Abiodun *et al.* (2018), também destacam o uso as Redes Neurais, estruturas inspiradas no cérebro humano, compostas por camadas de neurônios artificiais interconectados. Ainda segundo os autores, por meio do treinamento com conjuntos de dados, as redes neurais aprendem padrões complexos e realizam tarefas como reconhecimento de imagem e tradução automática. Quando há mais de uma camada oculta, denomina-se Aprendizado Profundo (*Deep Learning*), destacando-se em áreas como visão computacional e processamento de linguagem natural. Para Walczak (2019), de maneira mais intuitiva, as Redes Neurais calculam a importância de cada parâmetro do modelo por meio de um sistema de pesos. Durante o treinamento, esses pesos são ajustados para seguir o padrão presente nos dados de entrada, utilizando a técnica chamada *Backpropagation*.

Na prática, segundo L’Heureux *et al.* (2017), o ML encontra aplicações em diversas áreas, como reconhecimento de padrões, processamento de linguagem natural, visão computacional, diagnóstico médico, finanças, entre outras. À medida que os conjuntos de dados crescem em tamanho e complexidade, o aprendizado de máquina torna-se uma ferramenta essencial para extrair *insights* e tomar decisões precisas em cenários do mundo real (KEITH *et al.*, 2021). No entanto, desafios como a interpretabilidade dos modelos, ética e privacidade também emergem à medida que essa tecnologia se difunde e se enraíza em várias esferas da sociedade (L’HEUREUX *et al.*, 2017).

## REFERÊNCIAS

- ABIODUN, O. I. *et al.* State-of-the-art in artificial neural network applications: A survey. **Heliyon**, v. 4, n. 11, p. e00938, nov. 2018.
- ALZUBI, J. *et al.* Machine Learning from Theory to Algorithms: An Overview. **Journal of Physics: Conference Series**, v. 1142, p. 012012, nov. 2018.
- ARNOLD, F.; SÖRENSEN, K. What makes a VRP solution good? The generation of problem-specific knowledge for heuristics. **Computers & Operations Research**, v. 106, p. 280–288, jun. 2019.
- BARNES, J.; HUT, P. A hierarchical  $O(N \log N)$  force-calculation algorithm. **Nature**, v. 324, n. 6096, p. 446–449, dez. 1986.
- BI, Q. *et al.* What is Machine Learning? A Primer for the Epidemiologist. **American Journal of Epidemiology**, v. 188, n. 12, 21 out. 2019.
- BREEN, P. G. *et al.* Newton versus the machine: solving the chaotic three-body problem using deep neural networks. **Monthly Notices of the Royal Astronomical Society**, Oxford, v. 494, n. 2, p. 2465–2470, 22 abr. 2020.
- CIAMBUR, B. C. *et al.* Double X/Peanut structures in barred galaxies – insights from an N-body simulation. **Monthly Notices of the Royal Astronomical Society**, Oxford, v. 503, n. 2, p. 2203–2214, 11 dez. 2020.
- DEHNEN, W. A fast multipole method for stellar dynamics. **arXiv (Cornell University)**, 9 maio 2014.
- DRAKE, J. H. *et al.* Recent advances in selection hyper-heuristics. **European Journal of Operational Research**, v. 285, n. 2, p. 405–428, set. 2020.
- GARRISON, L. H. *et al.* The abacus cosmological N-body code. **Monthly Notices of the Royal Astronomical Society**, Oxford, v. 508, n. 1, p. 575–596, 7 set. 2021.
- GONZALEZ-SANCHEZ, A. *et al.* **Learning to Simulate Complex Physics with Graph Networks**. Ithaca, [2020]. Disponível em: <https://arxiv.org/abs/2002.09405>. Acesso em: 16 set. 2023.
- GREENER, J. G. *et al.* A guide to machine learning for biologists. **Nature Reviews Molecular Cell Biology**, v. 23, 13 set. 2021.
- HAMPL, S. K. *et al.* Comparison of computational algorithms for simulating an electrospray plume with a n-body approach. **Journal of Electric Propulsion**, v. 1, n. 1, 7 out. 2022.
- HERNANDEZ, D. M. Improving the accuracy of simulated chaotic N-body orbits using smoothness. **Monthly Notices of the Royal Astronomical Society**, Oxford, v. 490, n. 3, p. 4175–4182, 27 set. 2019.
- ZAPOROZHETS, I.; CLEMENTI, C. Multibody Terms in Protein Coarse-Grained Models: A Top-Down Perspective. **Journal of Physical Chemistry B**, v. 127, n. 31, p. 6920–6927, 27 jul. 2023.
- JEWETT, A. I. *et al.* Moltemplate: A Tool for Coarse-Grained Modeling of Complex Biological Matter and Soft Condensed Matter Physics. **Journal of Molecular Biology**, v. 433, n. 11, p. 166841, 28 maio 2021.

- KARIMI-MAMAGHAN, M. *et al.* Machine Learning at the service of Meta-heuristics for solving Combinatorial Optimization Problems: A state-of-the-art. **European Journal of Operational Research**, Leeds, v. 296, n. 2, p. 393-422, 2022.
- KEITH, J. A. *et al.* Combining Machine Learning and Computational Chemistry for Predictive Insights Into Chemical Systems. **Chemical Reviews**, v. 121, n. 16, p. 9816-9872, 7 jul. 2021.
- L'HEUREUX, A. *et al.* Machine Learning With Big Data: Challenges and Approaches. **IEEE Access**, Piscataway, v. 5, p. 7776-7797, 2017.
- MAKSIMOVA, N. A. AbacusSummit: a massive set of high-accuracy, high-resolution N-body simulations. **Monthly Notices of the Royal Astronomical Society**, Oxford, v. 508, n. 3, p. 4017-4037, 7 set. 2021.
- MOUSAVI, S. S. *et al.* Deep Reinforcement Learning: An Overview. **Proceedings of SAI Intelligent Systems Conference (IntelliSys) 2016**, p. 426-440, 23 ago. 2018.
- MUKHERJEE, D. *et al.* Fast Multipole Methods for N-body Simulations of Collisional Star Systems. **The Astrophysical Journal**, v. 916, n. 1, p. 9, 1 jul. 2021.
- OLIVEIRA, R. A. *et al.* Fast and Accurate Non-Linear Predictions of Universes with Deep Learning. **arXiv (Cornell University)**, 2 dez. 2020.
- PARPART, P. *et al.* Heuristics as Bayesian inference under extreme priors. **Cognitive Psychology**, v. 102, p. 127-144, maio 2018.
- RUCCI, E. *et al.* Optimization of the N-Body Simulation on Intel's Architectures Based on AVX-512 Instruction Set. **Communications in computer and information science**, p. 37-52, 1 jan. 2020.
- USAMA, M. *et al.* Unsupervised Machine Learning for Networking: Techniques, Applications and Research Challenges. **IEEE Access**, v. 7, p. 65579-65615, 2019.
- WANG, L. *et al.* petar: a high-performance N-body code for modelling massive collisional stellar systems. **Monthly Notices of the Royal Astronomical Society**, Oxford, v. 497, n. 1, p. 536-555, 24 jul. 2020.
- WALCZAK, S. Artificial Neural Networks. **Advanced Methodologies and Technologies in Artificial Intelligence, Computer Simulation, and Human-Computer Interaction**, p. 40-53, 2019.
- WU, H.; MENG, F. J. Review on Evaluation Criteria of Machine Learning Based on Big Data. **Journal of Physics: Conference Series**, Bristol, v. 1486, n. 5, p. 052026, 1 abr. 2020.
- WU, Y. *et al.* Learning Improvement Heuristics for Solving Routing Problems. **IEEE Transactions on Neural Networks and Learning Systems**, v. 33, n. 9, p. 5057-5069, 1 set. 2022.
- ZHANG, F. *et al.* Evolving Scheduling Heuristics via Genetic Programming With Feature Selection in Dynamic Flexible Job-Shop Scheduling. **IEEE Transactions on Cybernetics**, v. 51, n. 4, p. 1797-1811, abr. 2021.
- ZHU, X. *et al.* Efficient Utilization of Missing Data in Cost-Sensitive Learning. **IEEE Transactions on Knowledge and Data Engineering**, v. 33, n. 6, p. 2425-2436, 1 jun. 2021.
- ZOUNEMAT-KERMANI, M.; MAHDAVI-MEYMAND, A. Hybrid meta-heuristics artificial intelligence models in simulating discharge passing the piano key weirs. **Journal of Hydrology**, v. 569, p. 12-21, fev. 2019.