



UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO
DEPARTAMENTO DE ESTATÍSTICA E INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA APLICADA

IGOR FERREIRA TAVARES

**SIMULAÇÕES COMPUTACIONAIS UTILIZANDO
SMOOTHED PARTICLE HYDRODYNAMICS PARA A BUSCA
DE MÍNIMOS DE FUNÇÕES**

RECIFE-PE

2025

IGOR FERREIRA TAVARES

**SIMULAÇÕES COMPUTACIONAIS UTILIZANDO
SMOOTHED PARTICLE HYDRODYNAMICS PARA A BUSCA
DE MÍNIMOS DE FUNÇÕES**

Dissertação submetida à Coordenação do Programa de Pós-Graduação em Informática Aplicada do Departamento de Estatística e Informática - DEINFO - Universidade Federal Rural de Pernambuco e pela Fundação de Amparo à Ciência e Tecnologia do Estado de Pernambuco - FACEPE, como parte dos requisitos necessários para obtenção do grau de Mestre.

ORIENTADOR: Prof. Dr. Tiago A. E. Ferreira.

RECIFE – PE
2025

Dados Internacionais de Catalogação na Publicação
Sistema Integrado de Bibliotecas da UFRPE
Bibliotecário(a): Auxiliadora Cunha – CRB-4 1134

T231s Tavares, Igor Ferreira.
Simulações computacionais utilizando Smoothed
Particle Hydrodynamics para a busca de mínimos de
funções / Igor Ferreira Tavares. – Recife, 2025.
62 f.

Orientador(a): Tiago Alessandro Espínola Ferreira.

Dissertação (Mestrado) – Universidade Federal Rural de
Pernambuco, Programa de Pós-Graduação em Informática
Aplicada, Recife, BR-PE, 2025.

Inclui referências.

1. Hidrodinâmica de Partículas Suavizadas (SPH). 2.
Modelagem computacional. 3. Otimização matemática. 4.
Simulação computacional I. Ferreira, Tiago Alessandro
Espínola, orient. II. Título

CDD 004

IGOR FERREIRA TAVARES

**SIMULAÇÕES COMPUTACIONAIS UTILIZANDO
SMOOTHED PARTICLE HYDRODYNAMICS PARA A BUSCA
DE MÍNIMOS DE FUNÇÕES**

Dissertação submetida à Coordenação do Programa de Pós-Graduação em Informática Aplicada do Departamento de Estatística e Informática - DEINFO - Universidade Federal Rural de Pernambuco e pela Fundação de Amparo à Ciência e Tecnologia do Estado de Pernambuco - FACEPE, como parte dos requisitos necessários para obtenção do grau de Mestre.

Aprovada em: 05 de agosto de 2025.

BANCA EXAMINADORA

Prof. Dr. Tiago A. E. Ferreira (Orientador)

Universidade Federal Rural de Pernambuco - UFRPE
Presidente da Banca Examinadora

Prof.^a Dr.^a Kerolly Kedma Felix do Nascimento

Universidade Regional do Cariri – URCA
Membro da Banca

Prof. Dr. Henrique Correia Torres Santos

Instituto Federal de Pernambuco - Campus Recife - IFPE
Membro da Banca

Dedico esta pesquisa ao meu orientador
Prof. Dr. Tiago Alessandro Espínola Ferreira,
pelas preciosas e imensuráveis tempos
empenhados ao projeto, comparecendo
sempre com muita satisfação e confiança.

AGRADECIMENTOS

Quero agradecer primeiramente a Deus e a Jesus, sem a crer neles eu não teria a força e condições necessárias para ingressar na Coordenação do Programa de Pós-Graduação no Mestrado em Informática Aplicada do Departamento de Estatística e Informática - DEINFO - Universidade Federal Rural de Pernambuco e pela bolsa de estudos fornecida pela Fundação de Amparo à Ciência e Tecnologia do Estado de Pernambuco - FACEPE, para realizar esta dissertação.

A minha família, pelos momentos de superação que precisei para elaborar esta pesquisa.

Ao meu orientador, o Prof. Dr. Tiago Alessandro Espínola Ferreira, pela atenção, disponibilidade e os conhecimentos fornecidos para finalizar esse trabalho.

E também ao todo corpo docente da Universidade Federal Rural de Pernambuco, por todos os empenhos para oferecer o melhor ambiente e condições para que assim eu realizasse esse curso.

“Mas há uma outra razão que explica a elevada reputação das Matemáticas, é que elas levam as ciências naturais exatas uma certa proporção de segurança que, sem elas, essas ciências não poderiam obter”

(Albert Einstein)

RESUMO

A Hidrodinâmica de partículas suavizadas, ou do inglês *Smoothed Particle Hydrodynamics* (SPH), é um procedimento computacional usado para realizar simulações computacionais em meios contínuos, como em processos mecânicos e fluxos de fluidos, a qual vem ganhando crescente destaque na representação da dinâmica de fluidos. Dado um sistema físico composto por partículas, o SPH basicamente calcula a pressão sobre uma partícula levando em consideração as interações com as suas partículas vizinhas, simulando a dinâmica do sistema físico como um fluido. A proposta deste trabalho é utilizar a técnica SPH para a busca por mínimos de funções, onde é simulada a queda (por gravidade) de um conjunto de partículas sob uma superfície. Esta superfície é representada por uma função matemática a qual deseja-se encontrar o seu valor mínimo global. O uso do SPH, tradicionalmente aplicado em simulações físicas e industriais, é aqui explorado como prova de conceito, demonstrando que a técnica também pode ser adaptada para problemas de otimização. Com a dinâmica de escoamento das partículas sob a superfície em análise é possível identificar a partícula que alcança o valor de mínimo, localizando-a no domínio da função. Vários experimentos foram realizados com diversas funções com múltiplos pontos de mínimos locais e um mínimo global, onde os resultados obtidos mostraram que o SPH é uma ferramenta capaz de identificar os mínimos globais das funções analisadas. Para comparação, também foram realizados experimentos com a técnica de PSO (*Particle Swarm Optimization*). Os resultados demonstram que os resultados experimentais do SPH são tão bons quanto os resultados do PSO.

Palavras-chaves: Hidrodinâmica de Partículas Suavizadas, Modelagem Computacional, Problemas de Otimização e Simulação Computacional.

ABSTRACT

Smoothed Particle Hydrodynamics (SPH) is a computational procedure used to perform simulations in continuous media, such as mechanical processes and fluid flows, and has gained increasing relevance in the representation of fluid dynamics. Given a physical system composed of particles, SPH calculates the pressure on a particle by considering its interactions with neighboring particles, simulating the dynamics of the physical system as a fluid. The purpose of this work is to use the SPH technique to search for function minima by simulating the fall (by gravity) of a set of particles onto a surface. This surface is represented by a mathematical function whose global minimum value is sought. The use of SPH, traditionally applied in physical and industrial simulations, is explored here as a proof of concept, demonstrating that the technique can also be adapted to optimization problems. Through the dynamics of particle flow on the analyzed surface, it is possible to identify the particle that reaches the minimum value, locating it in the function's domain. Several experiments were conducted with functions containing multiple local minima and one global minimum, and the results showed that SPH is a tool capable of identifying the global minima of the analyzed functions. For comparison, experiments were also performed using the PSO (*Particle Swarm Optimization*) technique. The results demonstrate that the experimental outcomes of SPH are as effective as those obtained with PSO.

Keywords: Smoothed Particle Hydrodynamics, Computational Modeling, Optimization Problems, Computational Simulation.

LISTA DE FIGURAS

Figura 1	- Exemplo do problema encontrado, para modelar uma simulação de partículas sobre obstáculos	15
Figura 2	- A partícula q com raio de suporte h podem ser encontrados nas nove caixas vizinhas da célula à qual q pertence	32
Figura 3	- Fluxograma da simulação PySPH	43
Figura 4	- Ilustração da função Perm no PySPH <i>Particle View</i> , antes da simulação	46
Figura 5	- Ilustração da função Griewank no PySPH <i>Particle View</i> , antes da simulação	46
Figura 6	- Ilustração da função do Problema 02 do Teste Univariado no PySPH <i>Particle View</i> , antes da simulação	46
Figura 7	- Ilustração da função do Problema 07 do Teste Univariado no PySPH <i>Particle View</i> , antes da simulação	46
Figura 8	- Ilustração da função do Problema 12 do Teste Univariado no PySPH <i>Particle View</i> , antes da simulação	46
Figura 9	- Ilustração da função do Problema 14 do Teste Univariado no PySPH <i>Particle View</i> , antes da simulação	46
Figura 10	- Ilustração da função Perm no PySPH <i>Particle View</i> , após a simulação	47
Figura 11	- Ilustração da função Griewank no PySPH <i>Particle View</i> , após a simulação	47
Figura 12	- Ilustração da função do Problema 02 do Teste Univariado no PySPH <i>Particle View</i> , após a simulação	47
Figura 13	- Ilustração da função do Problema 07 do Teste Univariado no PySPH <i>Particle View</i> , após a simulação	47
Figura 14	- Ilustração da função do Problema 12 do Teste Univariado no PySPH <i>Particle View</i> , após a simulação	47
Figura 15	- Ilustração da função do Problema 14 do Teste Univariado no PySPH <i>Particle View</i> , após a simulação	47
Figura 16	- Gráfico da função Perm no PSO, após a simulação.....	48
Figura 17	- Gráfico da função Griewank no PSO, após a simulação.....	48
Figura 18	- Gráfico da função do Problema 02 do Teste Univariado no PSO, após a simulação	48

Figura 19 - Gráfico da função do Problema 07 do Teste Univariado no PSO, após a simulação	48
Figura 20 - Gráfico da função do Problema 12 do Teste Univariado no PSO, após a simulação	48
Figura 21 - Gráfico da função do Problema 14 do Teste Univariado no PSO, após a simulação	48

LISTA DE TABELAS

Tabela 1 - Resultados das simulações PySPH.....	50
Tabela 2 - Resultados das simulações da função Perm.....	50
Tabela 3 - Resultados das simulações da função Griewank.....	51
Tabela 4 - Resultados das simulações da função Problema 02 - Teste Univariado...	51
Tabela 5 - Resultados das simulações da função Problema 07 - Teste Univariado...	51
Tabela 6 - Resultados das simulações da função Problema 12 - Teste Univariado...	52
Tabela 7 - Resultados das simulações da função Problema 14 - Teste Univariado...	52
Tabela 8 - Resultados das simulações do PSO.....	52
Tabela 9 - Resultados de todas as simulações.....	53

LISTA DE SIGLAS

PySPH	<i>Código SPH implementado na linguagem Python</i>
SPH	<i>Smoothed Particle Hydrodynamics</i>
RKPM	<i>Reproducing Kernel Particle Method</i>
DEM	<i>Discrete Element Method</i>
FEM	<i>Finite Element Method</i>
PSO	<i>Particle Swarm Optimization</i>
TVF	<i>Transport Velocity Formulation</i>
SPHERIC	<i>SPH Research and EngineerRing International Community</i>
LES	<i>Large-Eddy Simulation</i>
FDM	<i>Finite Difference Method</i>
FVM	<i>Finite Volume Method</i>
PFEM	<i>Particle Finite Element Method</i>
CPUs	<i>Central Processing Units</i>
GPUs	<i>Graphics Processing Units</i>
MPI	<i>Message Passing Interface</i>
PIC	<i>Particle-in-Cell</i>
MD	<i>Molecular Dynamics</i>
MLPG	<i>Meshless Local Petrov–Galerkin</i>
EFG	<i>Element Free Galerkin</i>
EC	<i>Evolutionary Computation</i>

SSP	<i>Search Space Partitioning</i>
NS	<i>Naiver-Stokes</i>
EOS	<i>Equation of State</i>
ODE	<i>Ordinary Differential Equation</i>
CFL	<i>Courant-Friedrichs-Lewy</i>
CFD	<i>Computational Fluid Dynamics</i>
GS	<i>Smoothed Gradient</i>
DE	<i>Differential Evolution</i>
MSSOA	<i>Multi-Strategy Seeker Optimization Algorithm</i>
SOA	<i>Seeker Optimization Algorithm</i>
NNPS	<i>Nearest Neighbor Particle Searching</i>
SFC	<i>Continuous Transfer Function</i>
NCO	<i>Constrained Nonlinear Optimization</i>

SUMÁRIO

1 INTRODUÇÃO.....	13
1.1 PROBLEMA DA PESQUISA.....	15
1.2 OBJETIVOS.....	16
1.2.1 OBJETIVO GERAL.....	16
1.2.2 OBJETIVOS ESPECÍFICOS.....	16
1.3 JUSTIFICATIVA.....	17
1.4 ORGANIZAÇÃO DO TRABALHO.....	18
2 REVISÃO DA LITERATURA.....	19
3 FUNDAMENTAÇÕES TEÓRICAS.....	27
3.1 O MÉTODO SPH.....	28
3.1.1 INTEGRAÇÃO DE TEMPO.....	31
3.1.2 PESQUISA DE PARTÍCULAS VIZINHAS MAIS PRÓXIMAS.....	32
3.2 O PROBLEMA DE OTIMIZAÇÃO.....	34
3.2.1 FUNÇÃO DE FITNESS.....	35
4 METODOLOGIA.....	37
4.1 FUNÇÕES DE BENCHMARK.....	38
4.1.1 FUNÇÃO DE PERM.....	38
4.1.2 FUNÇÃO DE GRIEWANK.....	39
4.1.3 FUNÇÕES DE TESTE UNIVARIADO.....	39
4.1.3.1 FUNÇÕES DE TESTE UNIVARIADO ADAPTADAS.....	42
4.4 SIMULAÇÃO.....	43
4.5 BASE COMPARATIVA: O PSO.....	44
5 EXPERIMENTOS E RESULTADOS.....	45
5.1 FUNÇÕES DE OTIMIZAÇÃO ANTES DA SIMULAÇÃO NO PYSPH.....	45
5.2 FUNÇÕES DE OTIMIZAÇÃO APÓS A SIMULAÇÃO NO PYSPH.....	46
5.3 FUNÇÕES DE OTIMIZAÇÃO APÓS A SIMULAÇÃO NO PSO.....	47
5.4 RESULTADOS.....	48
5.4.1 ANÁLISE DOS RESULTADOS DE CADA FUNÇÃO NO PYSPH.....	49
5.4.2 ANÁLISE DOS RESULTADOS DE CADA FUNÇÃO NO PSO.....	50
5.4.3 ANÁLISE DE TODOS OS RESULTADOS ENCONTRADOS.....	53
6 CONCLUSÃO E RECOMENDAÇÕES.....	54
REFERÊNCIAS.....	57

1 INTRODUÇÃO

O problema de encontrar o mínimo de uma função matemática é um tema central em diversas áreas da ciência e engenharias, em termos gerais, esse problema envolve a identificação de pontos onde a função atinge seus valores mais baixos, o que é crucial para a otimização de processos e sistemas. A busca por mínimos de funções é amplamente aplicável em diversos problemas, por exemplo, no treinamento de redes neurais, onde o objetivo é minimizar a função de erro (ou função de *loss*) para melhorar a precisão do modelo.

Neste contexto, podemos imaginar que um certo relevo é descrito por uma função matemática, se aplicarmos um fluido sob ação da gravidade sobre esse relevo, o fluido tenderá a se depositar nos pontos de mínimos, essa analogia pode ser explorada utilizando a abordagem da Hidrodinâmica de Partículas Suavizadas ou do inglês *Smoothed Particle Hydrodynamics* (SPH), que foi originalmente proposta para simular problemas em astrofísica por Lucy e Leon (1977) e por Gingold e Monaghan (1977). Desde então, o método tem sido adaptado e aplicado para a simulação de fluidos em diversas áreas como engenharia, biomecânica, geofísica e computação gráfica (Monaghan, 1992; Liu e Liu, 2003).

Nesta pesquisa, o termo “Lagrangeano” refere-se a métodos numéricos que acompanham individualmente o movimento das partículas ao longo do tempo, rastreando suas trajetórias e propriedades, que seguem as equações de Lagrange. Em contraste, abordagens Eulerianas analisam o fluido em pontos fixos no espaço, pelos quais as partículas passam. O método utilizado é “sem malha” (*meshless*), ou seja, não depende de uma grade fixa, mas de partículas que interagem segundo leis físicas. Isso permite maior flexibilidade para representar geometrias complexas, grandes deformações e interfaces móveis em simulações dinâmicas e não lineares.

O método SPH é uma técnica de simulação numérica que representa fluidos através de partículas e permite a resolução de complexas equações que governam a dinâmica de sistemas físicos reais, como as equações de Euler e Navier-Stokes. Além disso, integra conceitos de métodos como o DEM (*Discrete Element Method*) (Cundall e Strack, 1979) e RKPM (*Reproducing Kernel Particle Method*) (Liu et al., 1997) para melhorar a precisão das simulações.

Para realizar as implementações do método SPH e otimização de funções para encontrar pontos mínimos, utilizamos a biblioteca PySPH¹. Essa ferramenta, conforme destacado pelos autores Ramachandran e Puri (2019), apresenta uma estrutura de código aberto para SPH escrita em Python, com as características de fácil uso, fácil extensão e suporte a paralelização não intrusiva e ao balanceamento de carga dinâmica. A PySPH fornece uma implementação do *Transport Velocity Formulation* (TVF) (Adami et al., 2013), permitindo comparações dos resultados com os métodos padrão SPH e TVF quando necessário.

A proposta deste trabalho é empregar a técnica de SPH em problemas de otimização, em particular para a determinação de mínimos de funções. Neste sentido, para a construção de uma base comparativa, dada a semelhança entre as técnicas, também será utilizada a técnica de PSO (*Particle Swarm Optimization*) para a busca de mínimos de funções. Na ciência da computação, o *Particle Swarm Optimization* (PSO) (Kennedy e Eberhart, 1995) é um método utilizado para realizar simulações em meios contínuos, podendo ser empregado para problemas de otimização.

Dessa forma, com o escopo computacional definido e as referências literárias e teóricas já organizadas, foram iniciadas as implementações com o uso da biblioteca PySPH, utilizando as equações do SPH e os problemas de teste para a definição dos mínimos das funções de Perm, Griewank, e os *Problema 02, 07, 12 e 14 definidos* no site *Global Optimization Benchmarks and AMPGO*². As simulações foram realizadas, totalizando seis experimentos que apresentaram plena compatibilidade com o ambiente PySPH, a qual também disponibiliza o *Particle View*, ferramenta que permite observar visualmente o processo simulado das partículas de fluido.

Sumarizando, esta dissertação propõe utilizar o método SPH para simular o comportamento de um fluido caindo sob ação da gravidade sobre um relevo matematicamente descrito, com o objetivo de identificar os pontos mínimos dessas funções. Nesta abordagem, buscamos encontrar o valor mínimo que, para as

¹ Disponível em: <https://pysph.readthedocs.io/>. Acesso em: 8 ago. 2025.

² Disponível em: https://infinity77.net/global_optimization/test_functions_1d.html#d-test-functions. Acesso em: 8 ago. 2025.

funções de *benchmark* utilizadas, corresponde ao zero, obtendo assim os melhores resultados para a otimização das funções estudadas.

1.1 PROBLEMA DA PESQUISA

É possível imaginar uma superfície sólida definida por uma certa função matemática. Estando tal superfície em um campo gravitacional, ao se jogar um fluido sobre esta superfície é esperado que o mesmo escorra por ação da gravidade, acumulando-se nos locais de maior profundidade. Neste sentido, é proposto o uso da técnica de SPH para a simulação do fluido, sob ação da gravidade, escoando sobre uma superfície sólida, marcando a partícula que alcançar a maior profundidade e definido este local como o valor de mínimo encontrado para a função que descreve a superfície sólida.

O trabalho de Ting et al. (2019), mostra um exemplo similar ao problema aqui tratado, conforme a Figura 1:

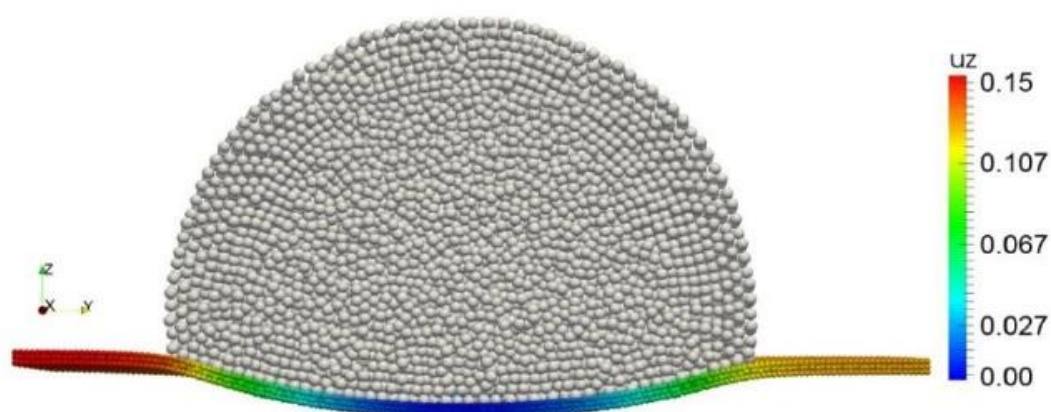


Figura 1 - Exemplo do problema encontrado, para modelar uma simulação de partículas sobre obstáculos. Induzida pela interação líquido-sólido. (Unidade: mm).

Fonte: Ting et al. (2019).

O trabalho de Ting explica como funciona a modelagem computacional de fluxos complexos, que geralmente apresenta grandes desafios para métodos numéricos convencionais baseados em grade. A *Smoothed Particle Hydrodynamics* (SPH) é um método de partículas Lagrangianas sem malha e tem vantagens especiais na modelagem de fluxos complexos, especialmente aqueles com grandes deformações, interações fluido-estrutura e multiescala. Nesse artigo, foram

revisados os desenvolvimentos recentes do SPH em metodologia e aplicações para modelagem de fluxos complexos. Especificamente, algumas questões importantes, incluindo esquemas modificados de aproximação de partículas SPH para melhorar a precisão da discretização, diferentes técnicas de regularização de partículas e vários algoritmos de tratamento de contorno para sólidos, superfície livre, ou interface multifásica são descritas. Também foi abordado o método SPH com dinâmica de partículas dissipativas para fluidos complexos em escalas macro ou mesoscópica. Nas aplicações, diferentes fluxos complexos, incluindo fluxos biológicos, microfluídica e dinâmica de gotículas, fluxos não newtonianos, fluxos de superfície livre, fluxos multifásicos e fluxos com interação fluido-estrutura, são revisados.

Em particular, no trabalho de Ting et al. (2019), os obstáculos são flexíveis, ou seja, quando as partículas batem nele, ele se deforma. Em contraponto, aqui no trabalho apresentado, os obstáculos são rígidos, não se deformando ao choque com as partículas. Outro ponto de destaque nesta dissertação é que os obstáculos não são planos (ou regulares), existindo vales e picos.

A mudança de cores apresentada na Figura 1 é a pressão referente ao impacto, onde o azul é o nível mais baixo da deformação do substrato elasto-plástico e o vermelho é o nível mais alto, igual a 0,15 mm.

1.2 OBJETIVOS

1.2.1 OBJETIVO GERAL

Essa dissertação tem como objetivo geral investigar o uso da metodologia de Hidrodinâmica de Partículas Suavizadas (SPH) para a descoberta de mínimos de funções matemáticas. Aqui, as partículas são as entidades que irão buscar o valor de mínimo em uma superfície, cujo relevo é descrito pela função matemática que se deseja caracterizar o seu mínimo.

1.2.2 OBJETIVOS ESPECÍFICOS

Neste trabalho é possível enumerar os seguintes objetivos específicos:

- 1) Analisar a biblioteca PySPH, desenvolvida em Python, que estabelece um *framework* para a criação de simulações computacionais por meio da técnica de SPH;

- 2) Propor funções matemáticas com mais de um mínimo local e pelo menos um mínimo global para definir um conjunto de *benchmarks* para a realização dos experimentos;
- 3) Desenvolver e implementar simulações computacionais usando o PySPH para descobrir os mínimos das funções matemáticas selecionadas para o conjunto de *benchmarks*;
- 4) Realizar uma análise comparativa entre os resultados obtidos com a metodologia SPH e outro método baseado em partículas com interações lineares, como a Otimização por Enxame de Partículas (PSO).

1.3 JUSTIFICATIVA

O problema de encontrar os mínimos de funções em um conjunto de partículas posicionadas sobre um obstáculo é um desafio matemático complexo, a ideia central é utilizar uma metodologia computacional livre de grades, que não exige o cálculo direto das derivadas dessas funções. Como dispomos apenas dos pontos, o método da Hidrodinâmica de Partículas Suavizadas (SPH) é adequado para calcular e determinar esses mínimos de forma prática.

O PySPH é uma biblioteca eficaz para modelar fluidos de partículas utilizando o método SPH. Ao empregar o PySPH, podemos realizar simulações que permitem encontrar e visualizar os mínimos das funções em tempo real.

Considerando um cenário onde temos apenas amostras da função e não sua expressão explícita, encontrar os mínimos torna-se um desafio computacional significativo, a abordagem tradicional envolveria o cálculo das primeiras e segundas derivadas para localizar os mínimos, mas, sem a expressão da função, isso pode não ser viável. Portanto, a metodologia proposta se justifica pela necessidade de uma solução numérica eficiente, como a oferecida pelo SPH, para resolver esse problema de otimização complexa.

Com todas as ideias expostas sobre o *Smoothed Particle Hydrodynamics*, observamos sua aplicação em simulações complexas em trabalhos na literatura, como por exemplo no trabalho de Myers et al. (2023), que destaca a facilidade do uso do método SPH, expondo uma abordagem híbrida *Finite Volume Method* (FVM)-(SPH) para captura de choques de sistemas compressíveis. No trabalho de Myers o *framework* Python PySPH é utilizado para modelar o fluido em regiões com

altos gradientes através de partículas com a técnica SPH. Novas abordagens de acoplamento FVM-SPH são exploradas, incluindo injeção e exclusão on-line de partículas em torno de regiões próximas ao choque de sistemas persistentes no tempo. A simulação SPH é acoplada à simulação FVM através de partículas de limite que circundam as regiões SPH ativas, com valores de FVM interpolados para partículas de limite SPH em cada intervalo de tempo. Os valores nas regiões SPH ativas são interpolados de forma semelhante à malha FVM sobreposta em cada intervalo de tempo.

1.4 ORGANIZAÇÃO DO TRABALHO

Este trabalho está estruturado em 6 capítulos. Após o Capítulo 1, que apresenta a introdução, o Capítulo 2 realiza uma revisão da literatura, analisando os trabalhos mais relevantes que contribuíram para o desenvolvimento desta dissertação.

No Capítulo 3 são apresentadas as fundamentações teóricas do método SPH, incluindo as equações necessárias para a sua implementação, como as equações de Euler-Lagrange, Navier-Stokes, DEM, RKPM, bem como as funções de otimização de *benchmark*, a função de *fitness* e o método PSO (que será utilizado como método comparativo).

O Capítulo 4 detalha toda a metodologia aplicada, descrevendo as funções escolhidas para o conjunto de *benchmarks*.

Os experimentos e resultados são apresentados no Capítulo 5, onde são discutidas as funções de otimização, incluindo *Fitness*, *Perm*, *Griewank* e Testes Univariados. Este capítulo também aborda os detalhes das implementações dos algoritmos que simulam os resultados obtidos utilizando o *framework* PySPH e o PSO.

Por fim, no Capítulo 6, são apresentadas as conclusões, com observações sobre as soluções encontradas, as dificuldades enfrentadas e as recomendações para trabalhos futuros.

2 REVISÃO DA LITERATURA

Os avanços nas pesquisas sobre *Smoothed Particle Hydrodynamics* (SPH), modelagem computacional, problemas de otimização e simulação computacional têm sido a base de numerosos estudos em diversos campos do conhecimento. Muitos pesquisadores apresentaram estudos com aplicações das equações de Euler-Lagrange e Navier-Stokes, em técnicas baseadas nos métodos *Discrete Element Method* (DEM) (Cundall e Strack, 1979) e *Reproducing Kernel Particle Method* (RKPM) (Liu et al., 1997).

O método SPH foi originalmente desenvolvido como uma técnica numérica para modelagem em astrofísica. Nas últimas décadas, sua aplicação se expandiu significativamente, abrangendo diversas áreas da ciência e da engenharia. Sua capacidade de resolver problemas complexos sem a necessidade de uma malha fixa tornou-se uma ferramenta essencial para simulações computacionais envolvendo fluidos, sólidos e específicos naturais. Este capítulo explora os principais avanços no uso do SPH, com foco em suas aplicações recentes e conclusões com a introdução do *Particle Swarm Optimization* (PSO) como uma base referencial para análises comparativas.

Os primeiros passos do SPH foram dados por Lucy e Leon (1977), que introduziram uma abordagem numérica para investigar as hipóteses de fissão em astrofísica. No mesmo ano, Gingold e Monaghan (1977) publicaram o trabalho *Smoothed Particle Hydrodynamics: Theory and Application to Non-Spherical Stars*. Estes trabalhos são referências fundamentais até os dias de hoje, nas diversas simulações em problemas de mecânica do contínuo e implementações com o método de partículas sem malha.

Após os primeiros passos no campo da astrofísica, o SPH começou a ser aplicado em novas áreas, com destaque para a integração com o *Discrete Element Method* (DEM) por Cisternas et al. (2019), gerando o DEM-SPH. Essa abordagem híbrida tem sido usada para prever o transporte de partículas sólidas interagindo com fluxos de lama. Essa técnica demonstra como o SPH pode ser eficaz em aplicações industriais, como a mobilidade autônoma, ao modelar interações complexas entre partículas sólidas e fluidas. Como já relatado de forma introdutória,

o SPH é um tipo de método baseado em partículas, sem malha, que emprega um conjunto finito de partículas discretas para representar o estado e a evolução de um sistema de fluxo. Este método acoplado DEM-SPH foi aplicado com sucesso à moagem autógena por Cleary et al. (2006) e Cleary et al. (2018).

Além de sua combinação com o DEM, em 2019, Asadi et al. (2019), também utilizaram o SPH em um processo de ruptura de uma membrana, localizada entre dois fluidos no centro de um canal tridimensional. A SPH e o *Finite Element Method* (FEM) foram utilizados, respectivamente, para modelar as fases fluida e sólida. Uma série de diferenças de pressão e espessuras de membrana foram estudadas e dois processos de ruptura diferentes foram identificados. Esses processos diferem na escala de tempo da ruptura, na localização do início da ruptura, no nível de destruição e no mecanismo de acionamento. Em continuidade ao uso do SPH em combinação com outros métodos, Xi et al. (2019), enfatizaram que as técnicas de partículas lidam principalmente com variações de partículas com base física para fenômenos e formas em animação computacional e modelagem geométrica. Técnicas típicas de partículas incluem simulação de SPH para animação e sistemas de partículas para modelagem de formas. Simulação SPH e sistemas de partículas são métodos sem malha e têm sido amplamente aplicados em engenharia e ciências aplicadas. Esses métodos aplicados incluem principalmente o método Lagrangiano, para simular fenômenos fluidos, como fumaça, fogo, explosões, ondas de água, bolhas e superfícies livres, que foram foco de pouca atenção no passado.

Prosseguindo com a evolução do SPH, Lind et al. (2020) aprofundaram os estudos sobre a aplicação do SPH em fluxos complexos com interfaces móveis e múltiplas fases. Os autores enfatizaram que, apesar de ser computacionalmente exigente, o SPH se beneficia significativamente de arquiteturas de computação paralelas, como GPUs, tornando-o mais eficiente em simulações de larga escala. Essa evolução tecnológica permitiu que o SPH fosse adotado em cenários anteriormente limitados por restrições computacionais.

Observando que vários pesquisadores progrediram em seus trabalhos e produziram resultados de destaque nas suas áreas, Koschier et al. (2020) mostram que o conceito SPH é cada vez mais popular em uma grande variedade de áreas de aplicação que vão desde tecnologias de entretenimento até engenharia. Por um

lado, esta popularidade baseia-se no fato de que as abordagens Lagrangianas em geral, e o SPH em particular, podem naturalmente lidar com cenários que seriam bastante complicados para abordagens Eulerianas. Um exemplo favorável é um fluido de superfície livre com limites sólidos geometricamente complexos e dinâmicos. Tal configuração é especialmente relevante para produções de efeitos especiais em indústria. O cenário, porém, tem a mesma relevância na engenharia, por exemplo, para a análise de veículos em passagens de água, para a previsão de evacuação de água da chuva em um veículo com limpadores de para-brisa em movimento, ou ainda para o projeto de caixas de engrenagens com lubrificação otimizada.

Em 2021, Vacondio et al. (2021) mostram uma breve revisão dos grandes desafios do método SPH. Como um método sem malha, o SPH pode simular uma ampla gama de aplicações, desde astrofísica a fluxos de superfície livre. Sendo um método computacional em desenvolvimento, o SPH ainda requer avanços para abordar elementos importantes que impedem uma utilização mais generalizada. Seguindo essa linha, Antuono et al. (2021) apresentam alguns desenvolvimentos recentes relativos à inclusão da *Large-Eddy Simulation* (LES) na estrutura de SPH fracamente compressíveis. Esse objetivo foi alcançado aplicando um filtro Lagrangiano às equações de Navier-Stokes para fluidos compressíveis e, em seguida, aproximando os operadores diferenciais no estilo SPH. Uma vez que a natureza Lagrangiana do esquema derivado revelou-se um obstáculo para simulações precisas de problemas com números de Reynolds elevados, a abordagem acima é aqui modificada para obter um modelo LES-SPH quase Lagrangiano.

Em continuidade, Wang et al. (2021) exploraram o uso do SPH para simular fluidos de filme fino, uma aplicação que combina deformações agressivas de superfície com fluxos tangenciais vigorosos. Os autores demonstraram que a compressibilidade do SPH, geralmente vista como uma limitação, pode ser explorada de maneira vantajosa para calcular tensão superficial e fortalecer a incompressibilidade de fluidos em filmes finos. Paralelamente, Zhang et al. (2021) coletam algumas aplicações recentes de SPH no campo dos riscos naturais ligados a fluxos rapidamente variados de água e densas misturas granulares, incluindo erosão de sedimentos e transporte de carga em leitos de rios. Este artigo reúne e

delineia os aspectos básicos de alguns trabalhos relevantes que tratam de inundações em topografia complexa, lavagem de sedimentos, dinâmica rápida de deslizamentos de terra e onda de surto induzida. Além disso, os resultados preliminares de um novo estudo sobre a dinâmica pós-falha de deslizamentos rasos induzidos pela chuva são apresentados. Considerando que a maioria dos métodos atuais de simulação computacional de sistemas dinâmicos são baseados em técnicas que utilizam uma malha ou grade, Fuchs et al. (2021) explicam que métodos, como *Finite Element Method* (FEM), *Finite Difference Method* (FDM) ou *Finite Volume Method* (FVM), requerem esforços metodológicos e computacionais substanciais para modelar o movimento de corpos rígidos embebidos em fluidos. Para superar esses problemas, várias abordagens baseadas em *Particle Finite Element Method* (PFEM), *Discrete Element Method* (DEM), ou no método de SPH foram propostas como alternativas mais viáveis. O SPH, que pode ser visto como um esquema de discretização sem malha, é devido à sua natureza Lagrangiana muito adequado para problemas de fluxo envolvendo múltiplas fases, transições de fase dinâmicas e reversíveis e topologias de interface complexas.

No campo da engenharia computacional, Wang et al. (2022) estudaram as técnicas de simulação numérica também foram aplicadas para prever a profundidade e o tempo de processamento no processo de perfuração a laser. Os métodos sem malha e baseados em partículas foram adequados para a simulação do processo de ablação a laser em relação à perda de material no processo. A técnica de modelagem *Smoothed Particle Hydrodynamics* foi usada para modelar a condução transitória de calor durante a interação entre o laser e o material.

Nos anos mais recentes, o método SPH tem demonstrado suas descobertas e aplicabilidade em uma ampla gama de estudos. Mohammadi e Riazi (2022) utilizaram o SPH para modelagem de fluxo de fluidos através de meios porosos com geometrias uniformes e não uniformes, simples e complexas é mostrada, e sua caracterização baseada na formulação SPH é investigada. Dado o esquema baseado em partículas do método SPH, permite extrair o histórico temporal das partículas e calcular a taxa de transferência do sistema. Resultados interessantes foram observados em relação ao efeito da geometria do cilindro nos perfis de velocidade e linhas de corrente. Eles mostram a possibilidade de utilização do método SPH para realizar estudos de caracterização dinâmica de escoamentos

subterrâneos. Nesse sentido, os resultados foram avaliados para fornecer relações práticas para aplicações futuras, enquanto Pahlke e Sbalzarini (2023) analisaram e comprovaram que esses mesmos algoritmos são amplamente utilizados para simulação computacional de fenômenos complexos em vários campos, como dinâmica de fluidos, física de plasma, química molecular e fluxos granulares, usando diversos métodos de simulação, incluindo o SPH, *Particle-in-Cell* (PIC), *Molecular Dynamics* (MD) e *Discrete Element Methods* (DEM). Apesar do uso crescente de métodos de partículas impulsionado pelo melhor desempenho computacional, a relação entre esses algoritmos permanece formalmente obscura e falta uma definição formal unificadora de métodos de partículas.

Nos últimos anos, métodos sem malha foram desenvolvidos e demonstrados como sendo mais eficazes na análise de processos envolvendo a deformação finita, isto é, grandes variações na geometria de materiais sob esforços extremos, como impacto ou compressão. Nestes contextos, técnicas tradicionais como o FEM enfrentam dificuldades relacionadas à distorção da malha, o que pode comprometer a precisão das simulações. Segundo Mountris et al. (2023), métodos sem malha aliviam parcial ou totalmente os requisitos de malha, evitando o problema de distorção da malha FEM. Exemplos de métodos sem malha incluem a *Smoothed Particle Hydrodynamics*, os *Meshless Local Petrov–Galerkin* (MLPG) (Atluri e Zhu, 1998; Hu et al., 2007) e os métodos de *Element Free Galerkin* (EFG) (Belytschko et al., 1994; Wu et al., 2021). O método SPH tem sido estendido com sucesso para adotar uma formulação Lagrangiana total, permitindo simulações mais precisas de grandes deformações. Como gradiente de funções constantes e lineares não são obtidos corretamente no método SPH padrão, um método de partículas suavizadas corrigidas foi utilizado para garantir a conservação do momento linear e angular. De maneira semelhante, o *Reproducing Kernel Particle Method* (RKPM) (Liu et al., 1997) utiliza uma função de correção para a aproximação de *Kernel* em SPH para atender às condições de reprodução e tem sido empregado para resolver grandes problemas de deformação.

Ainda em 2023, Raja et al. (2023) destacam os métodos sem malha, como a SPH, como ferramentas versáteis na simulação de processos complexos em manufatura aditiva, incluindo a impressão 3D de alimentos. Trata-se de uma técnica que consiste na deposição sucessiva de camadas de material comestível para

formar estruturas tridimensionais. Embora o SPH seja, em essência, um método lagrangiano, os autores ressaltam sua aplicabilidade em contextos que envolvem fenômenos comumente tratados por abordagens eulerianas, como a dinâmica de deposição de material. Por utilizar um conjunto móvel de partículas para discretizar as equações governantes, o SPH apresenta alta adaptabilidade, sendo adequado para lidar com grandes deformações, interfaces móveis e superfícies livres, além de permitir o acompanhamento do histórico temporal das variáveis de campo durante o processo de impressão.

Com o crescimento dessas aplicações, torna-se cada vez mais relevante integrar técnicas de otimização aos modelos computacionais, de modo a aprimorar o desempenho dos sistemas simulados. Nesse contexto, Duan et al. (2022) propuseram o algoritmo *Multi-Strategy Seeker Optimization Algorithm* (MSSOA), uma evolução do *Seeker Optimization Algorithm* (SOA), voltada para a solução de problemas complexos e restritos em engenharia. O MSSOA incorpora três estratégias principais: um mecanismo de captura tripla inspirado em buracos negros, que melhora a exploração do espaço de busca e evita a estagnação em ótimos locais; a interferência aleatória multidimensional, que amplia a diversidade das soluções; e a interferência precoce, que antecipa ajustes durante o processo iterativo. A eficácia do algoritmo foi validada por meio de experimentos com 15 funções de *benchmark* e aplicações reais, nas quais o MSSOA apresentou desempenho superior a outros seis métodos de otimização. As análises dos resultados incluíram a avaliação das curvas de convergência da função de *fitness*, da evolução da melhor solução ao longo das iterações e da distribuição espacial das soluções na população. Esses resultados reforçam a importância de estratégias inovadoras para o aprimoramento de algoritmos de otimização em problemas de engenharia.

Além dos avanços em algoritmos de otimização aplicados a problemas complexos de engenharia, esforços também têm sido direcionados ao aprimoramento de métodos numéricos para simulações físicas de alta complexidade. Nesse contexto, Myers et al. (2024) propuseram uma abordagem híbrida que combina o método de Volumes Finitos (FVM) com a *Smoothed Particle Hydrodynamics* para simulações de explosões com maior eficiência e precisão. Utilizando o *framework* PySPH, que integra processamento com NumPy e Cython

para acelerar os cálculos, os autores demonstraram que a combinação dos dois métodos permite capturar com mais fidelidade os efeitos dinâmicos e transientes de explosões em meios complexos. O PySPH destacou-se como uma ferramenta versátil, permitindo tanto a modificação de esquemas já existentes quanto a criação de novos, adaptando-se facilmente às necessidades específicas do problema. Os testes realizados incluíram análises comparativas de desempenho entre o SPH, o FVM puro e a versão híbrida, reforçando a viabilidade computacional e a precisão da proposta.

Como ferramenta complementar ao SPH, o *Particle Swarm Optimization* (PSO) (Kennedy e Eberhart, 1995) surge como uma alternativa promissora para abordagens comparativas ao SPH. Inspirado no comportamento de enxames, o PSO utiliza interações simples entre partículas para explorar o espaço de busca de maneira estocástica. Sua simplicidade e eficácia permitem avaliar o desempenho do SPH em termos de custo computacional e eficiência na busca por soluções ótimas, servindo como referência para comparação em problemas de otimização de funções.

Em contrapartida à técnica do PSO, há um ramo da Inteligência Artificial que é comumente chamada de computação evolutiva. Este ramo usa as ideias de processos evolutivos, encontrados na natureza, para criar sistemas que possam evoluir uma solução inicial, otimizando-a para uma dada situação de estudo. Tais métodos computacionais são geralmente baseados em um conjunto de partículas, ou população, ou ainda enxame, que se assemelha a filosofia empregada no SPH, mas em geral com iterações entre as partículas bem mais simples que as utilizadas no SPH. Desta forma, tais métodos podem ser empregados para a criação de uma referência comparativa, tanto em termos de custo computacional quanto de otimização da solução gerada para a resolução dos problemas em análise. Para conseguir uma solução ótima referente a Controle de diversidade em computação evolutiva usando populações duais assíncronas com particionamento de espaço de busca, Bashir e Hassan (2020), mostraram que o controle da diversidade é vital para uma otimização global eficaz usando técnicas de *Evolutionary Computation* (EC), abordando o compromisso entre exploração do espaço de busca e refinamento das soluções promissoras. Neste artigo, os autores implementam políticas de controle de diversidade que garantem a exploração sustentada do espaço de busca sem

comprometer a exploração eficaz das suas regiões promissoras. Primeiro, é proposto um algoritmo EC de *dual-pool* que facilita uma estratégia de diversificação da evolução temporal. Em seguida, uma inicialização heurística quase aleatória baseada no *search space partitioning* (SSP) (Bashir e Hassan, 2020) é introduzida para garantir uma amostragem uniforme do espaço de busca inicial. Em segundo lugar, para a medição da diversidade, um mecanismo robusto de detecção de convergência que combine uma medida de diversidade espacial é utilizado como uma medida de evolução da população. Verificou-se que o algoritmo proposto precisava de um tamanho de conjunto de apenas 50 amostras para convergir para soluções ótimas de uma variedade de *benchmarks* de otimização global. No geral, o algoritmo proposto produziu 33,34% de redução no custo incorrido por um algoritmo EC padrão. O resultado justifica a eficácia do controle eficaz da diversidade na resolução de cenários complexos de otimização global.

3 FUNDAMENTAÇÕES TEÓRICAS

As propriedades das dinâmicas de fluidos, especialmente em processos envolvendo o movimento de partículas líquidas ao longo de um canal, podem ser representadas por equações e funções matemáticas que descrevem o comportamento do fluxo ao longo do tempo. No trabalho de Antony e Maniyeri (2020), a simulação numérica do escoamento de fluido em um canal é realizada utilizando o SPH, um esquema Lagrangiano e sem malha, amplamente utilizado para modelar fenômenos em mecânica dos fluidos. As equações e funções utilizadas no SPH permitem descrever a interação entre as partículas de fluido e os contornos sólidos, como as paredes do canal, possibilitando simulações realistas dos padrões de escoamento. A abordagem dos autores reforça a aplicabilidade do SPH em sistemas confinados, nos quais a interação fluido-estrutura desempenha papel essencial para a compreensão do comportamento dinâmico do fluido.

Em métodos de partículas sem malha, as partículas distribuídas arbitrariamente são usadas para discretizar o domínio do problema. Nesta classe de métodos, as partículas não requerem uma conectividade fixa como nos métodos baseados em grade, portanto, o tratamento de grandes deformações é modelado computacionalmente de forma mais natural. Além disso, a discretização de domínios complexos é muito mais fácil em comparação com os métodos baseados em grade, pois apenas uma discretização inicial da partícula é necessária. Ao contrário de outros métodos sem malha onde os nós sem malha atuam apenas como pontos de interpolação, o SPH tem uma característica atraente: os nós também funcionam como componente material, carregando propriedades do material e movendo-se de acordo com as interações internas e externas. No SPH, utiliza-se a suavização para aproximar as variáveis de campo e suas derivadas a partir das partículas vizinhas.

Para enriquecer este trabalho e gerar uma base comparativa de desempenho na busca por mínimos de funções, também utilizamos o método PSO, que foi criado por Kennedy e Eberhart (1995). A otimização por enxame de partículas desenvolvida pelos autores citados compreende um conceito muito simples, podendo ser implementado em poucas linhas de código. Requer apenas operadores matemáticos primitivos e é computacionalmente barato em termos de requisitos de memória e

velocidade. Os primeiros testes revelaram que a implementação é eficaz para vários tipos de problemas.

Podemos citar, como exemplo de aplicação do PSO em contextos complexos, o trabalho de Tseng et al. (2021). Neste exemplo, os autores utilizaram o algoritmo para resolver problemas de otimização não linear com restrições (*Nonlinear Constrained Optimization* – NCO), nos quais o espaço de busca apresenta múltiplos ótimos locais e regiões inviáveis. O estudo demonstrou que, mesmo em cenários desafiadores, o PSO mantém desempenho satisfatório encontrando possíveis soluções. No presente trabalho, como já afirmado, o PSO é adotado como método de referência, permitindo avaliar comparativamente a eficiência da abordagem baseada em SPH na busca por mínimos em funções matemáticas.

A fim de organizar os fundamentos teóricos que embasam este trabalho, este capítulo está estruturado da seguinte forma: inicialmente, apresenta-se o SPH, destacando seus princípios e aplicações. Em seguida, discute-se a abordagem de otimização adotada, com foco na definição da função de *fitness* e na seleção de funções *benchmark* utilizadas para avaliação dos resultados.

3.1 O MÉTODO SPH

Como já mencionado e relatado nessa dissertação, o método SPH é um conjunto de métodos numéricos baseados em partículas, as quais são usadas como agentes dinâmicos que respeitam as características físicas do problema estudado. Progridem por meio da modelagem de equações diferenciais que ditam as leis que governam o problema. Em geral, o SPH pode ser usado para simular problemas que envolvem mecânica contínua, suas partículas são normalmente lagrangeanas e evoluem em um processo de interação mútua sem malha, permitindo lidar com problemas com superfícies livres e grandes deformações.

É possível observar os conceitos do SPH como método de partículas com uma visão computacional geral no trabalho dos autores Ramachandran et al. (2021). Este trabalho apresenta a ferramenta computacional PySPH empregada nos experimentos realizados nesta dissertação. Contudo, focando em uma visão mais formal matematicamente, remete-se o leitor ao trabalho de Violeau (2012) para uma detalhada discussão do formalismo do SPH.

Na sequência, apresenta-se uma breve introdução ao formalismo matemático do método SPH, visando destacar seus princípios fundamentais e sua formulação contínua.

SPH é um método de partículas lagrangeano, sem malha, onde as partículas representam o espaço contínuo ao seu redor. Uma variável de campo $f(\mathbf{x})$ sobre um domínio Ω pode ser aproximada por uma convolução com a função do *kernel* $W(\mathbf{x}, h)$, indicada na Equação (1),

$$f(x) = \int_{\Omega} f(y)W(x - y, h)dy \quad (1)$$

onde h é o comprimento de suavização e dy é o elemento de volume diferencial. A função do *Kernel* (ou núcleo) aproxima a função delta no limite $h \rightarrow 0$ e é uma função compacta tal que $W(r) = 0$, quando $|r| > kh$ para alguma constante k . Para obter a contraparte discreta, o campo é discretizado em partículas com massa m_i , densidade ρ_i , e o volume diferencial é substituído pelo volume $V_i = m_i/\rho_i$ onde o subscrito denota o índice da i -ésima partícula na posição $\mathbf{x}_i$. A integral da Equação (1) é então aproximada por uma soma discreta,

$$f(x_i) = \sum_j V_j f(x_j)W(x_i - x_j, h) \quad (2)$$

onde a soma é feita sobre toda a partícula no suporte da função *kernel*. Se a função $f(\mathbf{x}_i)$ é substituída pela densidade $\rho(\mathbf{x}_i)$, obtém-se a fórmula da densidade no SPH,

$$\rho(x_i) = \sum_j m_j W(x_i - x_j, h) \quad (3)$$

A aproximação SPH pode ser usada para discretizar equações diferenciais parciais. As Equações de movimento para um fluxo de fluido, expressas na forma lagrangeana, são,

$$\frac{d\rho}{dt} = -\rho \nabla \cdot \mathbf{u} \quad (4)$$

$$\frac{d\mathbf{u}}{dt} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{g} \quad (5)$$

onde $\frac{d(\cdot)}{dt}$ é a derivada do temporal, ρ é a densidade, p é a pressão, ν é a viscosidade cinemática, e $\mathbf{g}$ é a aceleração externa (em muitos exemplos reais a aceleração da gravidade). No caso de um fluido fracamente compressível, a Equação de Estado (ou do inglês, **Equation of State - EOS**), que completa as equações acima é,

$$p = \frac{c_0^2 \rho_0}{\gamma} \left[\left(\frac{\rho}{\rho_0} \right)^\gamma - 1 \right] \quad (6)$$

onde γ determina como a pressão varia com a densidade e influencia a compressibilidade do fluido, ρ_0 é a densidade de referência e C_0 é a velocidade do som. No caso de um fluido compressível, as equações incluem uma equação de energia é uma *Ordinary Differential Equation* (EOS) diferente. A forma discreta das equações acima usando a aproximação SPH pode ser escrita como,

$$\frac{d\rho_i}{dt} = \sum_j m_j (\mathbf{v}_i - \mathbf{v}_j) \cdot \nabla_i W_{ij} \quad (7)$$

$$\frac{d\mathbf{u}_i}{dt} = \mathbf{a}_{p,i} + \mathbf{a}_{visc,i} + \mathbf{a}_{body,i} \quad (8)$$

onde ρ_i , $\mathbf{u}_i$ são, respectivamente, a densidade e a velocidade da i -ésima partícula. $\mathbf{a}_{p,i}$ é a aceleração devida ao gradiente de pressão, comumente discretizada pela Equação (9) (Monaghan, 1992):

$$\mathbf{a}_{p,i} = - \sum_j m_j \left(\frac{p_i}{\rho_i^2} + \frac{p_j}{\rho_j^2} \right) \cdot \nabla_i W_{ij} \quad (9)$$

Essa expressão representa a contribuição das partículas vizinhas para o cálculo da aceleração devida ao gradiente de pressão sobre a partícula i , considerando as diferenças de pressão e as propriedades físicas envolvidas.

Na equação, p_i é a pressão da i -ésima partícula, m_j é a massa e ρ_j é a densidade da j -ésima partícula, $\nabla_i W_{ij} = \frac{\partial W}{\partial x_i} (\mathbf{x}_i - \mathbf{x}_j, \mathbf{h})$ é o gradiente do *kernel* de suavização e o somatório é realizado sobre todas as partículas que estão no domínio da função de *kernel*, exceto a partícula i . $\mathbf{a}_{visc,i}$ é a aceleração devido a forças viscosas, dada pela Equação (10) (Monaghan, 2005) é,

$$\mathbf{a}_{visc,i} = \sum_j m_j \frac{4\nu}{(\rho_i + \rho_j)} \frac{\mathbf{r}_{ij} \cdot \nabla_i W_{ij}}{(|\mathbf{r}_{ij}|^2 + \eta h^2)} \mathbf{u}_{ij} \quad (10)$$

onde $\mathbf{r}_{ij} = \mathbf{x}_i - \mathbf{x}_j$. Por fim, $\mathbf{a}_{body,i}$ é a aceleração devido às forças externas aplicadas ao corpo. Finalmente, as posições das partículas são obtidas integrando no tempo a seguinte equação diferencial,

$$\frac{dx_i}{dt} = \mathbf{u}_i \quad (11)$$

O problema que consideramos na próxima seção é a simulação de um sistema bidimensional referente à simulação de rompimento de uma barragem (Gomez-Gesteira et al., 2010), onde uma coluna de fluido é colocada em um tanque sob a influência da gravidade. Para fins de ilustração, consideramos um tratamento simples das condições de contorno. O tanque é representado por partículas sólidas com a mesma massa que as partículas fluidas nas quais resolvemos a equação de continuidade e, em seguida, define-se a pressão usando o EOS, Equação (6).

3.1.1 INTEGRAÇÃO DE TEMPO

As equações de movimento e de momento são integradas no tempo para obter as posições e velocidades, respectivamente. Segundo Monaghan (2005), um integrador é usado para realizar a integração no tempo, chegando-se às equações,

$$\mathbf{u}_i^{n+\frac{1}{2}} = \mathbf{u}_i^n + \frac{\Delta t}{2} \left(\frac{d\mathbf{u}_i}{dt} \right)^n \quad (12)$$

$$x_i^{n+1} = x_i^n + \Delta t u_i^{n+\frac{1}{2}}, \quad (13)$$

$$u_i^{n+1} = u_i^{n+\frac{1}{2}} + \frac{\Delta t}{2} \left(\frac{du_i}{dt} \right)^{n+1} \quad (14)$$

O passo de tempo para a integração é baseado no critério de *Courant-Friedrichs-Lewy* (CFL), o qual depende das equações diferenciais a serem resolvidas em Leveque (2002). Na literatura sobre o método SPH, o uso de passo de tempo implícito (*implicit time stepping*) é recomendado em problemas com alta rigidez numérica, pois melhora a estabilidade do sistema. No entanto, o *framework* PySPH, atualmente, não oferece suporte direto a esse tipo de integração. Adicionalmente, a rigidez do sistema, entendida como a presença de diferentes escalas de tempo que tornam o sistema difícil de resolver numericamente, não é explicitamente gerenciada pelo PySPH. Apesar dessas limitações, é importante destacar que o PySPH permite a solução iterativa de sistemas utilizando formulações sem matriz, o que pode ser vantajoso em simulações de grande escala.

3.1.2 PESQUISA DE PARTÍCULAS VIZINHAS MAIS PRÓXIMAS

Para encontrar as partículas que estão sob o suporte de um *kernel*, utiliza-se uma técnica de *nearest neighbor particle searching* (NNPS), conforme descrito em Ramachandran et al. (2021). A estrutura de dados de grade uniforme é a abordagem mais comumente usada para NNPS. Funciona dividindo as partículas em células de tamanho igual ao raio do suporte. Os vizinhos de uma partícula são então pesquisados nas caixas vizinhas da célula que contém a partícula de consulta, conforme mostrado na Figura 2.

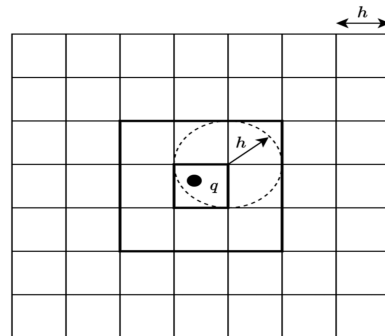


Figura 2 - Os vizinhos mais próximos da partícula q com raio de suporte h podem ser encontrados nas nove caixas vizinhas da célula à qual q pertence.

Nesse modelo bidimensional, a partícula q está localizada em uma célula da grade, e as partículas vizinhas são buscadas dentro de um raio de suporte h centrado nela. Como esse raio pode alcançar partículas em células vizinhas, é necessário considerar as 3^d caixas ao redor da célula onde d é o número de dimensões. Este método funciona eficientemente quando o raio de suporte é constante para todas as partículas, mas quando o raio de suporte é variável, o tamanho da célula precisa ser o raio máximo de suporte, o que torna a abordagem baseada em grade uniforme ineficiente à medida que as células se tornam muito grandes. Para encontrar vizinhos de forma eficiente em casos de raio de suporte variável, é necessária uma estrutura de dados com tamanho de célula adaptativo. A abordagem mais comum nesses casos é usar um Octree (Hernquist e Katz, 1989; Springel, 2005). Como parte do PySPH, fornecemos os seguintes algoritmos para NNPS na CPU:

— **Spatial Hash.** Neste método, cada célula da grade uniforme é mapeada para um índice em uma tabela *hash*. O uso de uma tabela *hash* permite o armazenamento apenas das células não vazias, reduzindo o uso de memória. Contudo, como os dados ficam distribuídos de forma não contígua na memória, essa dispersão prejudica a localidade espacial e temporal de acesso, diminuindo a eficiência do cache da CPU, que é otimizado para acessar dados sequenciais ou próximos entre si. Por esse motivo, essa abordagem pode apresentar um desempenho inferior em comparação a estruturas com melhor aproveitamento de cache.

— **Linked List. (Domínguez et. al., 2011).** Este método mapeia cada célula para um ID de célula 1D achatado, que é mapeado em uma matriz principal, que armazena a primeira partícula dessa célula. Em seguida, outra matriz mapeia cada partícula para a próxima partícula naquela célula, formando uma lista ligada. Este algoritmo oferece dados organizados de forma contígua na memória, isso melhora o acesso pela CPU e resulta em um melhor desempenho de cache tornando-o mais eficiente. No entanto, essa abordagem exige a alocação de um *array* com tamanho igual ao número total de células no domínio, inclusive as células vazias.

— **Z-Order Space Filling Curve.** Curvas de preenchimento de espaço são usadas para mapear dados 3D e 2D em uma dimensão, preservando a localidade espacial. Um SFC de ordem Z é construído classificando os IDs das partículas por chaves que

são geradas usando intercalação de bits (Morton, 1966). Este método oferece o melhor desempenho de cache em grades uniformes, mas requer uma classificação na etapa de construção, o que o torna geralmente mais lento do que o NNPS de lista vinculada.

— **Octree.** Neste método, dividimos recursivamente o espaço em oito octantes até que um nó tenha menos que um número especificado de partículas. Ao consultar vizinhos, a árvore é percorrida de cima para baixo e apenas os octantes que cruzam com a caixa delimitadora da região de consulta são percorridos.

Além destes, existem alguns algoritmos NNPS experimentais implementados em PySPH. Devido a uma série de restrições, nem todos os algoritmos acima podem ser implementados com eficiência em uma GPU. PySPH fornece o Z-Order SFC e os algoritmos Octree para NNPS na GPU.

3.2 O PROBLEMA DE OTIMIZAÇÃO

Independentemente do trabalho envolver situações simples ou complexas, o processo de otimização é muito eficaz na busca de soluções nos mais diversos problemas, e em diferentes áreas, como na ciência da computação, engenharia, economia e etc. Em geral, o processo de otimização consiste em maximizar ou minimizar uma função, buscando-se os melhores valores, ou valores ótimos, de uma função de interesse do problema em análise. Por exemplo, Azrag et al. (2020) em seu artigo fala que os algoritmos de otimização são aplicados em muitas situações, como atividades de negócios, engenharia e projetos industriais, com o objetivo de minimizar ou maximizar uma função objetivo. Ultimamente, algoritmos de otimização inspirados na natureza têm ganhado cada vez mais popularidade no campo da ciência e da engenharia. O desenvolvimento de tais algoritmo vêm cativando muitos pesquisadores, ora pelo fato da inspiração na natureza que vários algoritmos em destaque apresentam, ora pelos resultados de desempenho e autonomia, ou inteligência, de tais algoritmos. Desta forma, a questão da escolha de qual algoritmo aplicar à situação de estudo vem se tornando um tema de difícil resposta, pois além da grande quantidade de opções disponíveis, há a necessidade do entendimento dos pontos positivos e negativos que cada algoritmo apresenta em uma dada aplicação específica.

Nesse contexto, a seção buscou evidenciar a importância dos algoritmos de otimização como ferramentas versáteis e eficazes na resolução de problemas complexos. Essa discussão serve como base conceitual para o desenvolvimento das próximas etapas deste trabalho.

3.2.1 FUNÇÃO DE FITNESS

A função de *fitness*, ou função de aptidão, é uma função matemática utilizada como ferramenta para guiar o processo de otimização. Esta nomenclatura é oriunda dos algoritmos genéticos e outros algoritmos evolutivos. A função de *fitness* avalia a qualidade de uma solução (ou indivíduo/partícula em um algoritmo evolutivo) para um problema específico, indicando o quão bem essa solução se adapta ao ambiente ou atende aos critérios do problema.

Neste sentido, utilizamos a função de *fitness* para realizar otimizações, para encontrar um ponto de extremo, no nosso caso um ponto de mínimo da função objetivo. Então é desejável ter um esquema de otimização que leve ao ótimo global e não dependa diretamente da derivada da função.

Esse tipo de abordagem se mostra especialmente relevante em problemas de otimização do mundo real, nos quais o gradiente da função objetivo pode não estar disponível ou ser difícil de estimar com precisão. Conforme destacado por De Anda-Suárez et al. (2023), nessas situações, métodos tradicionais baseados em derivadas perdem eficácia, tornando necessário o uso de estratégias alternativas que utilizem informações da população de forma eficiente.

Uma possível melhoria na convergência dos algoritmos evolutivos reside na utilização das informações disponibilizadas pelos indivíduos para saber a melhor direção de busca pela solução ótima. Contudo, determinar tal direção implica em estimar o gradiente da função de *fitness*, mas a função de *fitness* não tem necessariamente uma derivada bem definida. Assim, é possível utilizar SPH com um gradiente gaussiano como uma aproximação do gradiente da função de aptidão (De Anda-Suárez et al., 2023).

A expressão da Equação (15) estima o gradiente interpolado da função de aptidão (ou *fitness*) f .

$$\langle \vec{\nabla} f(\vec{r}) \rangle = \int \int \dots \int_{\Omega} \vec{\nabla} f(\vec{r}') W(\|\vec{r} - \vec{r}'\|, h) d^n \vec{r} \quad (15)$$

Seguindo a propriedade $\nabla[a \cdot b] = \nabla a \cdot b + a \cdot \nabla b$, encontramos a Equação(16):

$$\begin{aligned} \vec{\nabla} f(\vec{r}') W(\|\vec{r} - \vec{r}'\|, h) &= \int_{\Omega} \vec{\nabla} \cdot [f(\vec{r}') W(\|\vec{r} - \vec{r}'\|, h)] d^n \vec{r} - \\ &\int_{\Omega} f(\vec{r}') \vec{\nabla} W(\|\vec{r} - \vec{r}', h\|) d^n \vec{r} \end{aligned} \quad (16)$$

a qual pode ser reescrita como,

$$\int_{\Omega} \vec{\nabla} \cdot [f(\vec{r}') W(\|\vec{r} - \vec{r}'\|, h)] d^n \vec{r} = \int_S f(\vec{r}') W(\|\vec{r} - \vec{r}'\|, h) \cdot \hat{n} ds \quad (17)$$

Esse termo considera que o *kernel* é uma função compacta e a integral de superfície é igual a zero, implicando na Equação (18):

$$\langle \vec{\nabla} f(\vec{r}) \rangle = - \int \int \dots \int_{\Omega} f(\vec{r}') \vec{\nabla} W(\|\vec{r} - \vec{r}'\|, h) d^n \vec{r} \quad (18)$$

Em resumo da seção, apresentamos uma derivação analítica do gradiente da função de *fitness* envolvendo apenas as avaliações da referida função no espaço de busca. Consequentemente, a Equação (18) permite saber a direção ótima que deve-se tomar para a otimização da função de *fitness* do problema, sem a necessidade de uma derivação explícita da função de *fitness*.

Do resultado da Equação (18), projetamos um *kernel* baseado na distribuição gaussiana, G , devido à sua aproximação direta da função delta de Dirac, onde quanto menor h (parâmetro de suavização do *kernel* gaussiano), melhor será a aproximação. Utilizamos o modelo gaussiano apresentado na Equação (19):

$$G(\|\vec{r} - \vec{r}'\|, h) = A_n \cdot \exp\left(-\frac{\|\vec{r} - \vec{r}'\|^2}{h^2}\right) \quad (19)$$

onde a constante de normalização A_n é descrita na Equação (20):

$$A_n = \frac{1}{\int \int \dots \int_{-\infty}^{\infty} f(\vec{r}') W(\|\vec{r} - \vec{r}'\|, h) d^n \vec{r}} = \frac{1}{\left(\pi^{\frac{n}{2}} h^n\right)} \quad (20)$$

Neste ponto, temos os elementos para expressar a interpolação do gradiente da função de *fitness* a partir dos resultados das Equações (17), (19) e (20). Portanto, a Equação (21) expressa o gradiente da função a ser otimizada em termos do gradiente gaussiano. Daqui em diante, chamaremos este gradiente de *Smoothed Gradient* (GS).

$$\langle \vec{\nabla} f(\vec{r}) \rangle = - \int \int \dots \int_{\Omega} f(\vec{r}') \vec{\nabla} \left[\frac{1}{\pi^{\frac{n}{2}} h^n} \exp\left(-\frac{\|\vec{r} - \vec{r}'\|^2}{h^2}\right) \right] d^n \vec{r} \quad (21)$$

Por fim, destaca-se que, para avaliação do desempenho de métodos de otimização, é comum o uso de funções de *benchmark*, especialmente aquelas caracterizadas por múltiplos mínimos locais e apenas um mínimo global. Essas funções são amplamente adotadas na literatura para testar a robustez e a eficiência de algoritmos numéricos. No presente trabalho, esse conjunto de funções será explorado como parte da metodologia experimental, conforme apresentado no capítulo seguinte.

4 METODOLOGIA

Em nossa metodologia, buscamos testar uma proposta de determinação numérica de mínimos de funções matemáticas com o uso da técnica de SPH. Para tanto, elegemos algumas funções de *benchmarks* para a realização de testes de desempenho em acurácia na determinação dos valores de mínimos.

Inicializamos este capítulo apresentando as funções de *benchmark* eleitas para os testes e, em seguida, apresentamos o esquema empregado nas simulações com a técnica SPH para a definição de mínimos globais de funções matemáticas.

4.1 FUNÇÕES DE BENCHMARK

Neste trabalho foram utilizadas algumas funções matemáticas com vários mínimos locais e apenas um mínimo global para testar a eficiência do SPH na busca por mínimos de funções. Neste sentido, as funções aqui escolhidas servem como funções de *benchmark*, as quais poderão ser utilizadas para uma análise comparativa de desempenho.

De acordo com a necessidade do projeto, pode-se escolher o algoritmo que melhor se adapte ao problema. As funções de *benchmark* são amplamente utilizadas para testar diversos algoritmos numéricos de Otimização Global, sendo aplicadas especialmente a funções de teste multimodais e multidimensionais difíceis.

Algumas dessas funções foram retiradas da literatura, enquanto outras foram geradas com o auxílio de ferramentas específicas para criação de funções de teste.

Essas funções de teste permitem avaliar a robustez e a capacidade de generalização dos algoritmos, fornecendo cenários controlados com características bem definidas. Assim, torna-se possível analisar o desempenho do SPH frente a diferentes níveis de complexidade e topologias de busca (Arora et al., 2020).

Nas subseções subsequentes estão sendo apresentadas as funções de *benchmark* que foram eleitas a partir de uma busca na literatura e suas respectivas caracterizações.

4.1.1 FUNÇÃO DE PERM

A função de Perm escolhida é uma das funções *benchmarks* eleitas para testar o processo de otimização realizado pelo SPH. Kashyap et al. (2020) também aplicaram esta função para a análise de técnica de *Differential Evolution* (DE), uma técnica utilizada na computação evolutiva.

A função de Perm é exibida na Equação (22),

$$f(x) = \sum_{i=1}^d \left(\sum_{j=1}^d (j + \beta) \left(x_j^i - \frac{1}{j^i} \right) \right)^2 \quad \begin{matrix} x_i \in [-d, d], \forall i= \\ 1, 2, 3, \dots, d \end{matrix} \quad (22)$$

4.1.2 FUNÇÃO DE GRIEWANK

A função de Griewank também foi uma das escolhidas para os teste e experimentos de otimização com o SPH, sendo esta função uma combinação de termos de diferenças quadráticas e funções trigonométricas que cria um cenário complexo com vários mínimos locais. Gao et al. (2020), em sua pesquisa sobre Algoritmos Robustos de Forrageamento Bacteriano Baseado na Estratégia de Orientação de Poucos Indivíduos Excelentes, aplicaram a Função de Griewank, demonstrando a usabilidade desta função com processo de *benchmark*.

A descrição matemática da função de Griewank é mostrada na Equação (23). Esta função é uma função multimodal complexa com pontos mínimos locais e grandes obstáculos. A alta correlação das variáveis induz aos algoritmos de otimização um ótimo local (mínimo local). O ponto ideal global da função está em $x_i = \{0, \dots, 0\}$, com o valor da função correspondente de $f(x_i) = 0$.

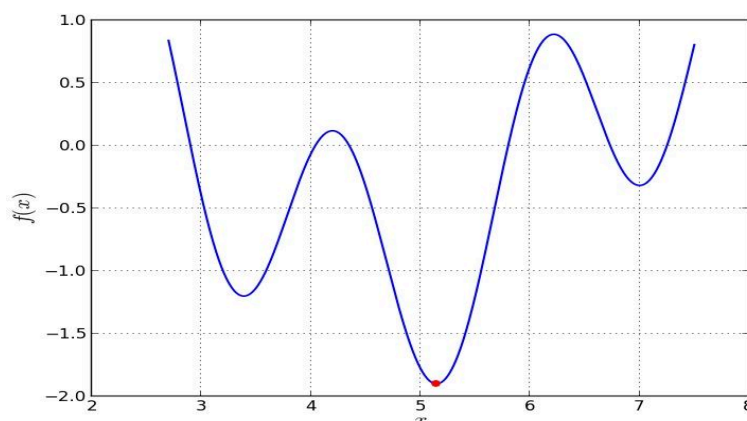
$$f(x) = \frac{1}{4000} \sum_{i=1}^n (x_i - 100)^2 - \prod_{i=1}^n \cos \left(\frac{x_i - 100}{\sqrt{i}} \right) + 1 \quad (23)$$

4.1.3 FUNÇÕES DE TESTE UNIVARIADO

Para o nosso problema escolhemos as funções de testes univariados com características de problemas de testes multimodais.

Na busca de enriquecer mais esse trabalho, seguindo o trabalho de Grant-Peters e Hauser (2020), e mais funções de *benchmark* foram eleitas a partir do site infinity77 (2023), que contém várias funções de *benchmarks* para problemas de otimização, foram utilizadas quatro funções de Teste Univariado deste site que representam problemas de minimização multimodal definido das seguintes formas:

- **Função Problema 02 – Teste Univariado.**

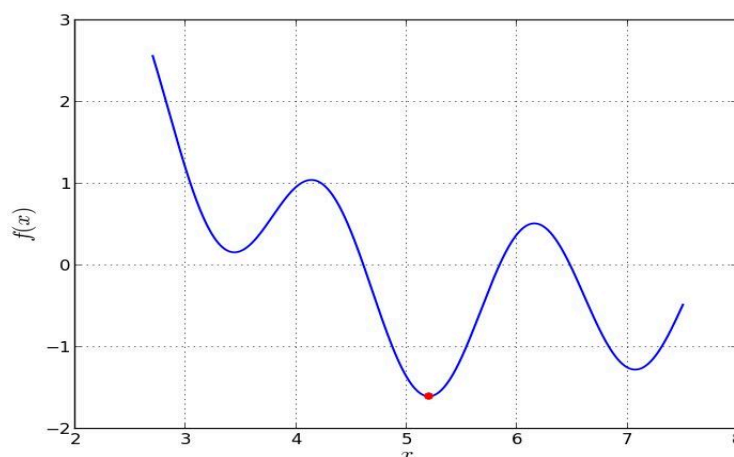


$$f_{\text{problema 02}}(x) = \sin(x) + \sin\left(\frac{10}{3}x\right) \quad (24)$$

As restrições vinculadas do eixo x , tem que obedecer a seguinte condição: $x \in [2.7, 7.5]$, para que a função possa ser exibida corretamente.

Ótimo global: $f(x) = -1,899599$, para $x = 5,145735$.

- **Função Problema 07 – Teste Univariado.**

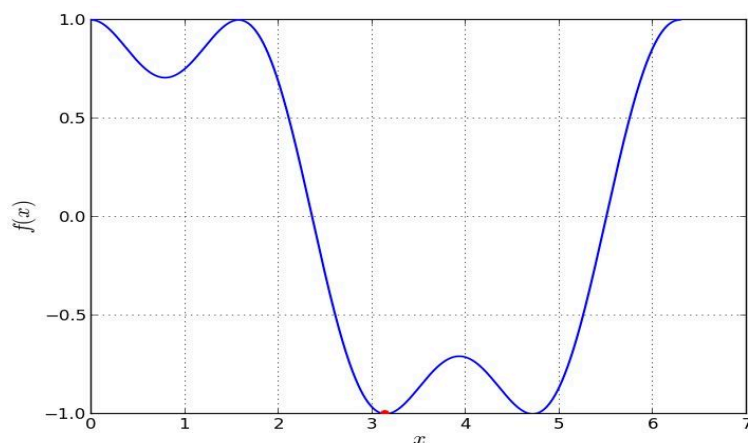


$$f_{\text{problema 07}}(x) = \sin(x) + \sin\left(\frac{10}{3}x\right) + \log(x) - 0,84x + 3 \quad (25)$$

As restrições vinculadas do eixo x , tem que obedecer a seguinte condição: $x \in [2.7, 7.5]$, para que a função possa ser exibida corretamente.

Ótimo global: $f(x) = -1,6013$, para $x = 5,19978$.

- **Função Problema 12 – Teste Univariado.**

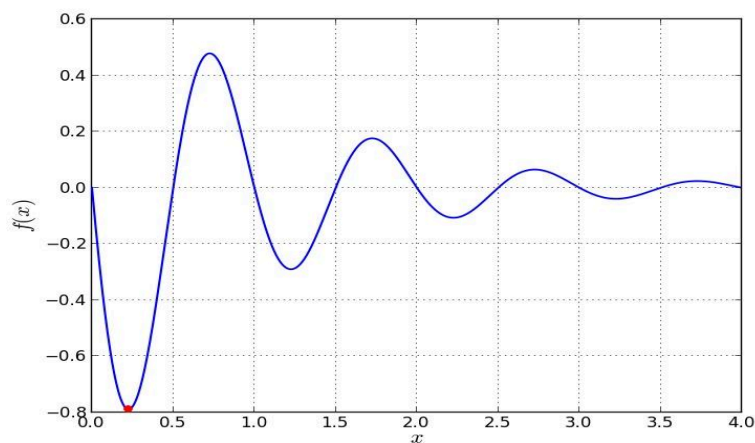


$$f_{problema\ 12}(x) = \sin^3(x) + \cos^3(x) \quad (26)$$

As restrições vinculadas do eixo x , tem que obedecer a seguinte condição: $x \in [0, 2\pi]$, para que a função possa ser exibida corretamente.

Ótimo global: $f(x) = -1$, para $x = \pi$ (3,14159).

- **Função Problema 14 – Teste Univariado.**



$$f_{problema\ 14}(x) = -e^{-x} \sin(2\pi x) \quad (27)$$

As restrições vinculadas do eixo x , tem que obedecer a seguinte condição: $x \in [0, 4]$, para que a função possa ser exibida corretamente.

Ótimo global: $f(x) = -0,788685$, para $x = 0,224885$.

4.1.3.1 FUNÇÕES DE TESTE UNIVARIADO ADAPTADAS

Para obter os resultados próximos do 0, foi preciso adaptar as funções de otimização de Teste Univariado, do site infinity77, mencionado na Seção 4.1.3, para atender ao propósito deste trabalho, pois os seus resultados ótimos mínimos são:

- **Função Problema 02 – Teste Univariado**

Ótimo global: $f(x) = -1,899599$.

- **Função Problema 07 – Teste Univariado**

Ótimo global: $f(x) = -1,6013$.

- **Função Problema 12 – Teste Univariado**

Ótimo global: $f(x) = -1$.

- **Função Problema 14 – Teste Univariado**

Ótimo global: $f(x) = -0,788685$.

Uma pequena adaptação dessas funções, foi adicionado ao final dos cálculos a soma desses valores, só que ao invés de negativos, os tornando positivos, obtendo assim o valor = 0, ficando as funções reescritas da seguinte forma:

- **Função Problema 02 – Teste Univariado.**

$$f_{problema\ 02}(x) = \sin(x) + \sin\left(\frac{10}{3}x\right) + 1,89959 \quad (28)$$

Ótimo global: $f(x) = 0$, para $x = 5,145735$.

- **Função Problema 07 – Teste Univariado.**

$$f_{problema\ 07}(x) = \sin(x) + \sin\left(\frac{10}{3}x\right) + \log(x) - 0,84x + 3 + 1,6013 \quad (29)$$

Ótimo global: $f(x) = 0$, para $x = 5,19978$.

- **Função Problema 12 – Teste Univariado.**

$$f_{problema\ 12}(x) = \sin^3(x) + \cos^3(x) + 1 \quad (30)$$

Ótimo global: $f(x) = 0$, para $x = \pi (3,14159)$.

- **Função Problema 14 – Teste Univariado.**

$$f_{problema\ 14}(x) = -e^{-x} \sin(2\pi x) + 0,788685 \quad (31)$$

Ótimo global: $f(x) = 0$, para $x = 0,224885$.

4.4 SIMULAÇÃO

O fluxograma, ilustrado na Figura 3, apresenta o funcionamento do PySPH em tempo real, onde são realizadas várias etapas: a 1ª é a inicialização da simulação; a 2ª etapa é a codificação de todas as funções implementadas na simulação, no caso foi utilizado o código quebra de barragem em 2D como base para o desenvolvimento e implementação dos experimentos realizados; a 3ª etapa são as partículas SPH trabalhando em conjunto com as funções em estudo, calculando e guardando os mínimos e as interações; a 4ª etapa é a evolução desse processo para encontrar a interação e os mínimos; a 5ª etapa é quando se encontra o mínimo de x em relação a y , tudo isso sendo executado na função escolhida que é o nosso sólido sob o qual as partículas de fluido escorrem livremente em busca do ponto mínimo global. Após o final da simulação, já com a interação e os pontos mínimos de x e y , chega-se à última etapa, a visualização da dinâmica simulada. No Framework PySPH existe um visualizador chamado *Particle View*, sendo possível visualizar toda a dinâmica executada com o escoamento do fluido sobre a superfície estudada, no caso as funções de *benchmark*.



Figura 3 - Fluxograma da simulação PySPH.

As simulações realizadas demonstraram uma forma numérica para a determinação do mínimo da função estudada. Estas simulações podem ser executadas tanto em 2D como em 3D, mas escolhemos realizar os experimentos em 2D, porque é mais leve e rápido computacionalmente, apresentando um conceito de prova para demonstrarmos a possibilidade de se buscar por mínimos de funções. Nos experimentos realizados, criamos uma linha de partículas fluidas que vai cair sobre uma superfície sólida (em um certo domínio), sendo tais superfícies as funções de *benchmark* escolhidas. Assim, as partículas irão cair por gravidade, escorrendo e acomodando-se sobre toda a superfície do sólido. Como são conhecidas as coordenadas de todas as partículas, é fácil descobrir qual a partícula que encontrou o menor valor de “altura” sobre a superfície sólida, descobrindo o ponto referente ao mínimo global da função para o dado domínio de busca.

A metodologia adotada neste trabalho visa aplicar o método SPH à resolução de problemas de otimização de funções matemáticas. A técnica foi escolhida por sua flexibilidade, ausência de malha e potencial de adaptação a diferentes tipos de problemas, mesmo aqueles em que o espaço de busca apresenta regiões complexas ou descontinuidades.

No entanto, para validar os resultados obtidos com o SPH, é necessário dispor de uma técnica de referência, amplamente estudada e reconhecida pela literatura, que permita estabelecer uma base de comparação justa e confiável. Essa comparação torna-se ainda mais relevante diante da ausência de trabalhos anteriores que explorem o uso do SPH para este tipo de aplicação.

Nesse contexto, optou-se pela utilização do algoritmo *Particle Swarm Optimization*, uma técnica clássica da Computação Evolutiva, com forte presença em estudos de otimização global. A seguir, apresenta-se a estrutura metodológica adotada para a implementação do PSO neste estudo.

4.5 BASE COMPARATIVA: O PSO

Para compor uma base comparativa que permita avaliar a eficácia do método SPH na resolução de problemas de otimização, este trabalho adota o algoritmo *Particle Swarm Optimization* como técnica de referência. O PSO é um método amplamente utilizado na literatura científica e reconhecido por sua capacidade de

encontrar soluções próximas dos mínimos de funções matemáticas em diversos contextos.

Sua implementação é simples, leve e de baixo custo computacional, além de apresentar bons resultados mesmo com um número limitado de iterações. Por essa razão, o PSO foi selecionado para realizar os mesmos testes aplicados ao SPH, permitindo uma comparação direta entre os desempenhos.

Ambos os métodos foram aplicados às mesmas funções de *benchmark*, utilizando condições similares de execução, como número de partículas e limites do domínio de busca. Dessa forma, é possível realizar uma análise equilibrada e fundamentada sobre os pontos fortes e limitações de cada abordagem. Os detalhes sobre a implementação, os parâmetros utilizados e os resultados obtidos com o PSO são apresentados no capítulo seguinte.

5 EXPERIMENTOS E RESULTADOS

Este capítulo vem demonstrar todos os procedimentos experimentais e resultados obtidos da determinação dos mínimos das funções de *benchmarks* realizadas tanto com a proposta do PySPH como a criação de uma base comparativa de resultados encontrados pela técnica do PSO. Este capítulo foi dividido da seguinte forma: 5.1 - Funções de Otimização antes da simulação no PySPH, 5.2 - Funções de Otimização após a simulação no PySPH, 5.2.1 - Funções de Otimização após a simulação no PSO, 5.3 - Resultados: 5.3.1 - Análise dos resultados de cada Função no PySPH, 5.3.2 - Análise dos resultados de cada Função no PSO e 5.3.3 - Análise de todos os resultados encontrados.

5.1 FUNÇÕES DE OTIMIZAÇÃO ANTES DA SIMULAÇÃO NO PYSPH

As partículas sólidas, exibidas nas Figuras: 4-9, utilizando o PySPH *Particle View*, formaram as funções: Perm, Griewank, Problema 02 - Teste Univariado, Problema 07 - Teste Univariado, Problema 12 - Teste Univariado e por fim o Problema 14 - Teste Univariado, acompanhadas de 2 (duas) paredes para evitar que o fluido das partículas que estão aguardando o início da simulação em formato de bloco retangular, preenchendo todo o domínio estudado, não saiam do espaço

reservado. Em todas as funções estudadas o mesmo objetivo foi buscado, obter os resultados mais próximos possíveis do mínimo, como já descrito no Capítulo 4.

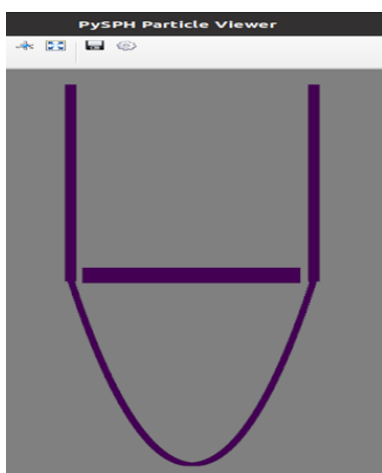


Figura 4
Função de Perm

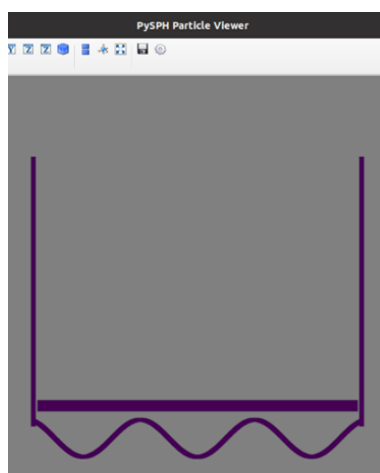


Figura 5
Função de Griewank

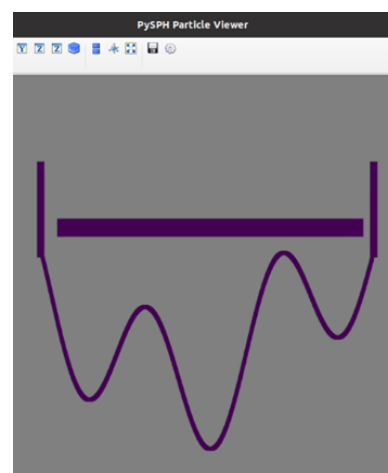


Figura 6
Função do Problema 02
Teste Univariado

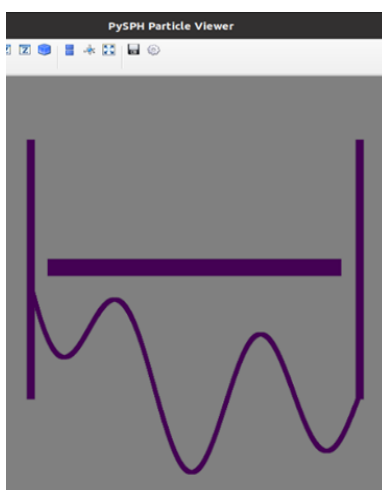


Figura 7
Função do Problema 07
Teste Univariado

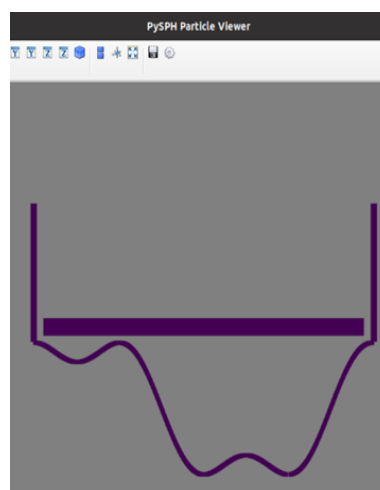


Figura 8
Função do Problema 12
Teste Univariado

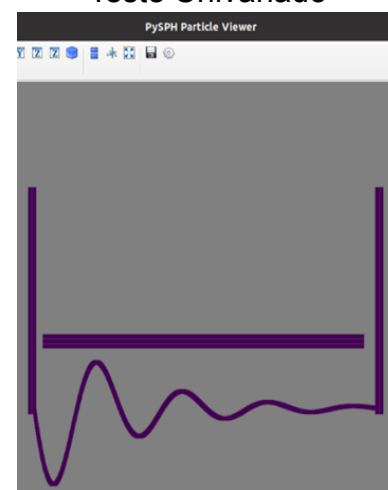


Figura 9
Função do Problema 14
Teste Univariado

5.2 FUNÇÕES DE OTIMIZAÇÃO APÓS A SIMULAÇÃO NO PYSPH

Após iniciar os processos utilizando o visualizador do PySPH, o *Particle View*, os fluidos das partículas percorreram todas as funções: Perm, Griewank, Problema 02 - Teste Univariado, Problema 07 - Teste Univariado, Problema 12 - Teste Univariado e por fim o Problema 14 - Teste Univariado, até o final da simulação. As Figuras 10-15, apresentam a função de *fitness* realizando as otimizações, juntamente com as funções, para encontrar os valores mínimos de x em relação a y , próximo de 0, como já mencionado. A função de Perm e de Testes Univariados, que o valor ideal de x são aleatórios e para y , em todas as funções o valor ideal é 0.

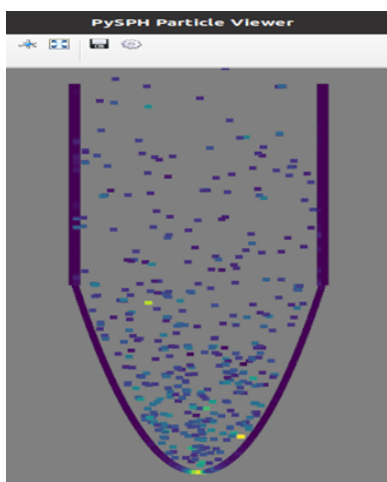


Figura 10
Função de Perm

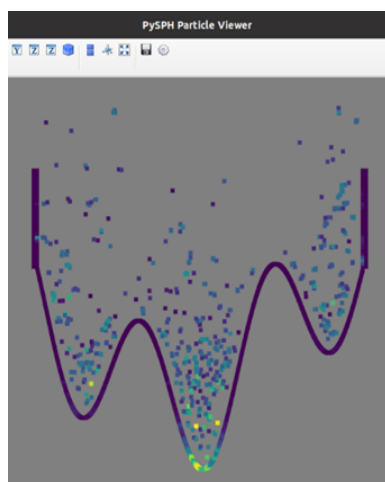


Figura 11
Função de Griewank

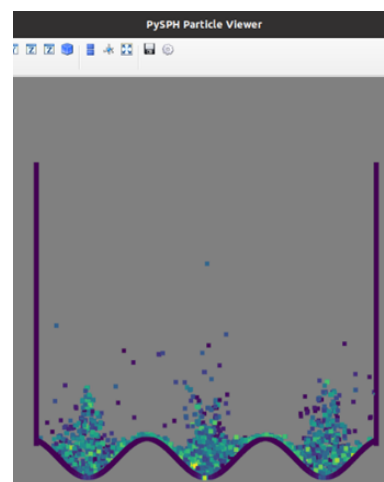


Figura 12
Função do Problema 02
Teste Univariado

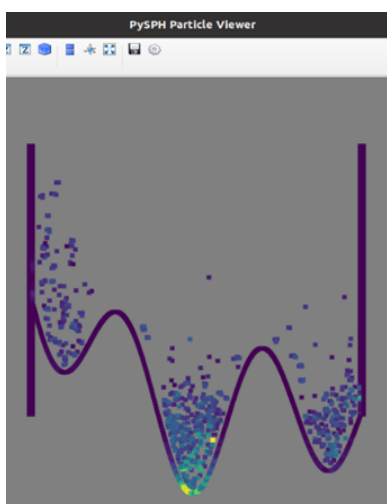


Figura 13
Função do Problema 07
Teste Univariado

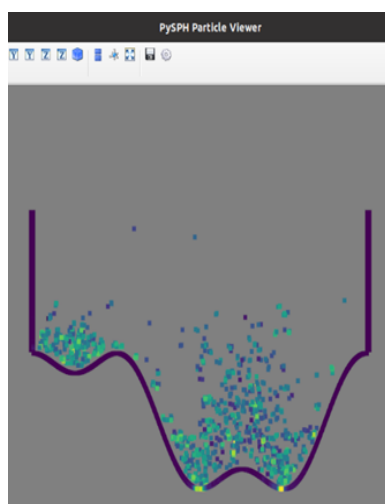


Figura 14
Função do Problema 12
Teste Univariado

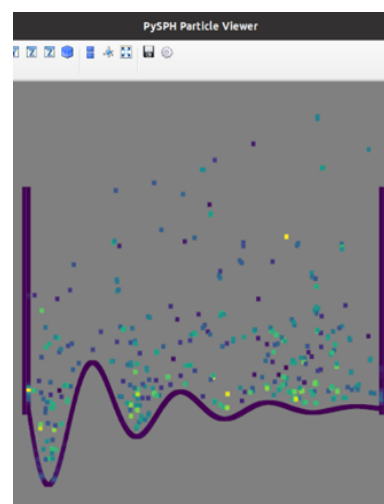


Figura 15
Função do Problema 14
Teste Univariado

5.3 FUNÇÕES DE OTIMIZAÇÃO APÓS A SIMULAÇÃO NO PSO

Com o mesmo objetivo de se encontrar o mínimo das funções com o uso do PySPH e para gerar uma base comparativa, simulamos também no PSO todas as funções: Perm, Griewank, Problema 02 - Teste Univariado, Problema 07 - Teste Univariado, Problema 12 - Teste Univariado e o Problema 14 - Teste Univariado. As simulações com o uso do PSO têm as suas evoluções apresentadas nas Figuras 16-21, onde o eixo x representa a evolução do PSO (ou as iterações) e o eixo y representa o valor de mínimo encontrado para a respectiva função em estudo. Os últimos valores dos gráficos representam os valores mínimos encontrados, que de

forma geral são todos próximos de 0. É possível notar que não é preciso colocar muitas interações para o PSO evoluir, foi utilizado no máximo 50 iterações.

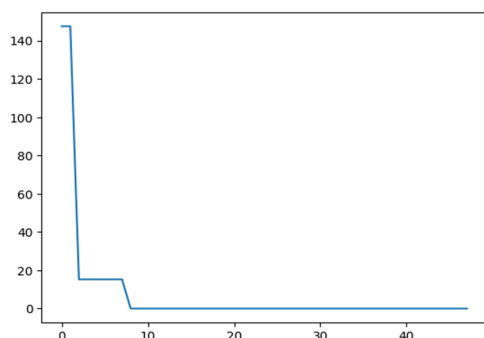


Figura 16
Função de Perm - PSO

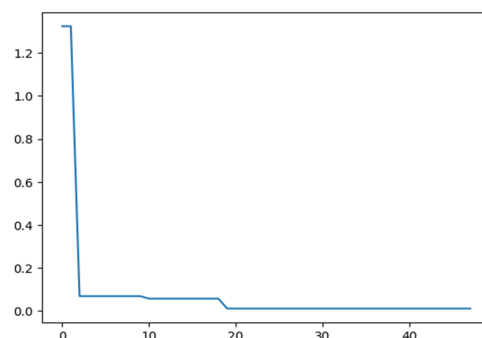


Figura 17
Função de Griewank - PSO

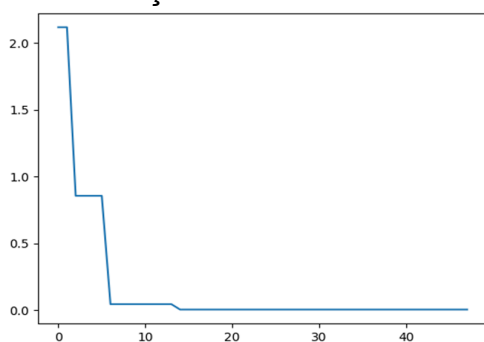


Figura 18
Função do Problema 02
Teste Univariado - PSO

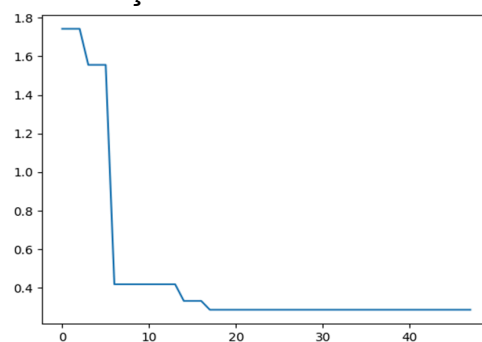


Figura 19
Função do Problema 07
Teste Univariado - PSO

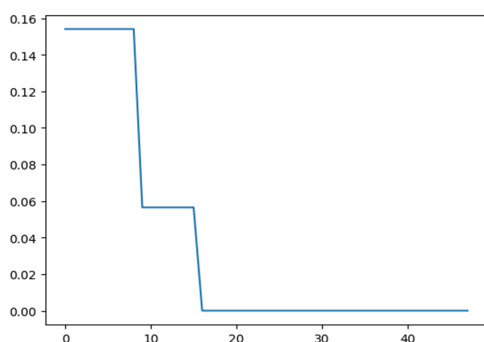


Figura 20
Função do Problema 12
Teste Univariado - PSO

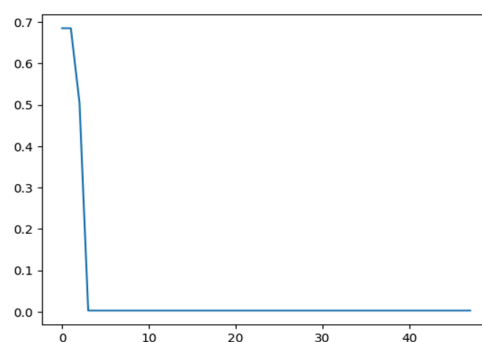


Figura 21
Função do Problema 14
Teste Univariado - PSO

5.4 RESULTADOS

Cada experimento tem distintos tempos de simulação, embora todos os experimentos tenham sido adaptados do mesmo código, o código *dam break* 2D. Este código é um exemplo disponível na biblioteca PySPH, sendo utilizado como referência para realizar as implementações das funções, onde a estrutura original foi mantida, só havendo a inclusão das condições de cada função.

Para as simulações realizadas com o PSO, utilizamos a biblioteca *pyswarm*, que já disponibiliza uma estrutura simples para a implementação do algoritmo, facilitando a adaptação das funções de *benchmark* adotadas neste estudo (Miranda, 2025).

Todos os resultados obtidos apresentaram êxito nas simulações realizadas. A 5.3.1 é dedicada à análise individual de cada função no PySPH, descrevendo os comportamentos observados e destacando as características específicas de cada simulação. Em seguida, a 5.3.2 concentra-se no PSO, possibilitando verificar como as mesmas funções se comportam em um método distinto de otimização e permitindo estabelecer paralelos entre as abordagens. Por fim, a 5.3.3 consolida essas análises, reunindo os resultados obtidos nos dois métodos e discutindo de forma detalhada o desempenho individual de cada função. Essa organização possibilita não apenas uma avaliação criteriosa de cada técnica de forma isolada, mas também uma comparação equilibrada que evidencia as características e particularidades de cada abordagem dentro do contexto do estudo.

5.4.1 ANÁLISE DOS RESULTADOS DE CADA FUNÇÃO NO PYSPH

Os cálculos envolvendo o método SPH e as funções estudadas exigem um nível de complexidade alto, como também um custo computacional alto. O *framework* PySPH, facilita bastante nos processamentos dos cálculos e a visualização através do PySPH *Particle View*.

Ao final das simulações a função de *fitness* realizou as otimizações em conjunto com as funções de *benchmarks*, o PySPH encontrou as interações que informam o tempo que as simulações levaram para encontrar o valor mais próximo possível do mínimo ideal das posições das partículas x e y de cada função. O próximo passo foi a inclusão do valor de x encontrado (posição da partícula), em cada função para a realização do cálculo do seu respectivo valor. O resultado mínimo teórico para todas as funções, como já visto, é igual a 0. Por último, também são apresentados os valores dos erros absolutos das diferenças entre os valores encontrados e os valores teóricos verdadeiros. Na Tabela 1, estão todos os resultados encontrados para as funções estudadas.

Tabela 1 - Resultados das simulações PySPH.

FUNÇÕES	INT.	X ENCONTRADO	Y ENCONTRADO	CÁLCULO DA FUNÇÃO COM O X ENCONTRADO	RESULTADO MÍNIMO TEÓRICO PARA A FUNÇÃO	VALOR DO ERRO ABSOLUTO DA DIFERENÇA
Perm	4146	1,00095	$5,703 \times 10^{-2}$	$2,043 \times 10^{-6}$	0	$2,043 \times 10^{-6}$
Griewank	2821	0,00848	$2,440 \times 10^{-2}$	$1,799 \times 10^{-8}$	0	$1,799 \times 10^{-8}$
Problema 02 - Teste Univariado	3337	5,14589	$8,269 \times 10^{-2}$	$-1,989 \times 10^{-7}$	0	$1,989 \times 10^{-7}$
Problema 07 - Teste Univariado	3229	5,19957	$5,611 \times 10^{-2}$	$-7,282 \times 10^{-6}$	0	$7,282 \times 10^{-6}$
Problema 12 - Teste Univariado	2863	3,14368	$2,463 \times 10^{-2}$	$6,539 \times 10^{-6}$	0	$6,539 \times 10^{-6}$
Problema 14 - Teste Univariado	2306	0,22512	$8,781 \times 10^{-2}$	$5,428 \times 10^{-7}$	0	$5,428 \times 10^{-7}$

5.4.2 ANÁLISE DOS RESULTADOS DE CADA FUNÇÃO NO PSO

O PSO é conhecido por ser estocástico devido à sua natureza probabilística e aleatória. Nas funções: Perm, Griewank, Problema 02 - Teste Univariado, Problema 07 - Teste Univariado, Problema 12 - Teste Univariado e o Problema 14 - Teste Univariado, as partículas se movimentam em busca do melhor valor da solução ideal e melhor valor da solução para o x de cada função.

Como os resultados são aleatórios e também muito próximos, não sendo preciso realizar muitas repetições para se ter uma boa representação dos valores esperados. Foram realizadas 10 repetições do PSO para cada função estudada, sendo calculados a média e o desvio padrão, estando os resultados apresentados nas Tabelas 2 até a 7, sendo estes resultados sumarizados na Tabela 8.

Tabela 2 - Resultados das simulações da função Perm.

TENTATIVAS	INTERAÇÕES	VALOR		
		X ENCONTRADO	SOLUÇÃO IDEAL P/ Y	CALCULADO COM X
1	10	1,00000014	0,00000000000000463	$1,823 \times 10^{-16}$
2	4	1,00000034	0,00000000000002558	
3	42	1,00000009	0,00000000000000176	
4	24	1,00000004	0,00000000000000037	
5	26	0,99999975	0,00000000000001406	
6	18	0,99999990	0,00000000000000219	
7	19	1,00000008	0,00000000000000152	
8	14	0,99999973	0,00000000000001677	
9	46	1,00000009	0,00000000000000180	
10	3	0,99999993	0,00000000000000113	
MÉDIA	20,6	1,000000009	0,00000000000000698	
DESVIO PADRÃO		0,000000185	0,00000000000000871	

Tabela 3 - Resultados das simulações da função Griewank.

TENTATIVAS	INTERAÇÕES	VALOR		CALCULADO COM X
		X ENCONTRADO	SOLUÇÃO IDEAL P/ Y	
1	11	-0,000002964846	0,000000000004397	$1,439 \times 10^{-17}$
2	19	0,000000683942	0,000000000000234	
3	5	-0,000000725661	0,000000000000263	
4	13	-0,000000593040	0,000000000000176	
5	48	0,000000180196	0,000000000000016	
6	13	-0,000000866123	0,000000000000375	
7	35	0,000000677919	0,000000000000230	
8	37	0,000001802971	0,0000000000001626	
9	14	-0,000000036655	0,000000000000001	
10	41	-0,000000557798	0,000000000000156	
MÉDIA	23,60	-0,00000023991	0,000000000000747	
DESVIO PADRÃO		0,00000126343	0,000000000001365	

Tabela 4 - Resultados das simulações da função Problema 02 - Teste Univariado.

TENTATIVAS	INTERAÇÕES	VALOR		CALCULADO COM X
		X ENCONTRADO	SOLUÇÃO IDEAL P/ Y	
1	48	5,14573537	-0,00000034915208	$-3,492 \times 10^{-7}$
2	30	5,14573530	-0,00000034915211	
3	15	5,14573539	-0,00000034915206	
4	32	5,14573486	-0,00000034915100	
5	45	5,14573552	-0,00000034915181	
6	6	5,14573576	-0,00000034915081	
7	18	5,14573532	-0,00000034915211	
8	11	5,14573533	-0,00000034915211	
9	2	5,14573544	-0,00000034915199	
10	36	5,14573450	-0,00000034914838	
MÉDIA	24,30	5,14573527900	-0,00000034915144	
DESVIO PADRÃO		0,00000035325	0,00000000000118	

Tabela 5 - Resultados das simulações da função Problema 07 - Teste Univariado.

TENTATIVAS	INTERAÇÕES	VALOR		CALCULADO COM X
		X ENCONTRADO	SOLUÇÃO IDEAL P/ Y	
1	37	5,19977858	-0,000007546494130	$-7,546 \times 10^{-6}$
2	30	5,19977836	-0,000007546494390	
3	31	5,19977825	-0,000007546494310	
4	22	5,19977841	-0,000007546494390	
5	40	5,19977799	-0,000007546493510	
6	18	5,19977816	-0,000007546494120	
7	17	5,19977877	-0,000007546493450	
8	25	5,19977763	-0,000007546491150	
9	8	5,19977811	-0,000007546493990	
10	19	5,19977810	-0,000007546493960	
MÉDIA	24,70	5,19977823600	-0,000007546493740	
DESVIO PADRÃO		0,00000031924	0,000000000000967	

Tabela 6 - Resultados das simulações da função Problema 12 - Teste Univariado.

TENTATIVAS	INTERAÇÕES	VALOR		
		X ENCONTRADO	SOLUÇÃO IDEAL P/ Y	CALCULADO COM X
1	4	3,14159335	0,000000000000072	$7,760 \times 10^{-14}$
2	32	3,14159262	0,000000000000000	
3	8	3,14159471	0,000000000000636	
4	40	3,14159264	0,000000000000000	
5	12	3,14159263	0,000000000000000	
6	29	3,14159258	0,000000000000001	
7	45	3,14159278	0,000000000000002	
8	38	3,14159249	0,000000000000004	
9	36	3,14159258	0,000000000000001	
10	46	3,14159243	0,000000000000007	
MÉDIA	29,00	3,14159288100	0,000000000000072	
DESVIO PADRÃO		0,00000069138	0,000000000000199	

Tabela 7 - Resultados das simulações da função Problema 14 - Teste Univariado.

TENTATIVAS	INTERAÇÕES	VALOR		
		X ENCONTRADO	SOLUÇÃO IDEAL P/ Y	CALCULADO COM X
1	47	0,22488039	-0,000000387408673	$-3,569 \times 10^{-7}$
2	6	0,22488043	-0,000000387408636	
3	28	0,22488041	-0,000000387408666	
4	30	0,22488035	-0,000000387408647	
5	40	0,22488036	-0,000000387408659	
6	47	0,22488039	-0,000000387408672	
7	35	0,22488043	-0,000000387408645	
8	42	0,22488031	-0,000000387408586	
9	49	0,22488016	-0,000000387407878	
10	17	0,22488032	-0,000000387408613	
MÉDIA	34,10	0,22488035500	-0,000000387408568	
DESVIO PADRÃO		0,00000008031	0,000000000000244	

Tabela 8 - Resultados das simulações do PSO.

FUNÇÕES	INT.	X ENCONTRADO	Y ENCONTRADO	CÁLCULO DA FUNÇÃO COM O X ENCONTRADO	RESULTADO MÍNIMO TEÓRICO PARA A FUNÇÃO	VALOR DO ERRO ABSOLUTO DA DIFERENÇA	DESVIO PADRÃO
Perm	21	1,000000009	$6,981 \times 10^{-14}$	$1,823 \times 10^{-16}$	0	$1,823 \times 10^{-16}$	$1,85 \times 10^{-7}$
Griewank	24	-0,000000240	$7,475 \times 10^{-13}$	$1,439 \times 10^{-17}$	0	$1,439 \times 10^{-17}$	$1,26 \times 10^{-6}$
Problema 02 - Teste Univariado	24	5,145735279	$-3,492 \times 10^{-7}$	$-3,492 \times 10^{-7}$	0	$3,492 \times 10^{-7}$	$3,53 \times 10^{-7}$
Problema 07 - Teste Univariado	25	5,199778236	$-7,546 \times 10^{-6}$	$-7,546 \times 10^{-6}$	0	$7,546 \times 10^{-6}$	$3,19 \times 10^{-7}$
Problema 12 - Teste Univariado	29	3,141592881	$7,234 \times 10^{-13}$	$7,760 \times 10^{-14}$	0	$7,760 \times 10^{-14}$	$6,91 \times 10^{-7}$
Problema 14 - Teste Univariado	34	0,224880355	$-3,874 \times 10^{-7}$	$-3,569 \times 10^{-7}$	0	$3,569 \times 10^{-7}$	$8,03 \times 10^{-8}$

5.4.3 ANÁLISE DE TODOS OS RESULTADOS ENCONTRADOS

Tabela 9 - Resultados de todas as simulações.

FUNÇÕES	INT.		X ENCONTRADO		Y ENCONTRADO		CÁLCULO DA FUNÇÃO COM O X ENCONTRADO		RESULTADO MÍNIMO TEÓRICO PARA A FUNÇÃO		VALOR DO ERRO ABSOLUTO DA DIFERENÇA.		DESVIO PADRÃO	
	PySPH	PSO	PySPH	PSO	PySPH	PSO	PySPH	PSO	PySPH	PSO	PySPH	PSO	PySPH	PSO
Perm	4146	21	1,00095	1,000000009	$5,703 \times 10^{-2}$	$6,981 \times 10^{-14}$	$2,043 \times 10^{-6}$	$1,823 \times 10^{-16}$	0	0	$2,043 \times 10^{-6}$	$1,823 \times 10^{-16}$	—	$1,85 \times 10^{-7}$
Griewank	2821	24	0,00848	-0,0000000240	$2,440 \times 10^{-2}$	$7,476 \times 10^{-13}$	$1,799 \times 10^{-8}$	$1,439 \times 10^{-17}$	0	0	$1,799 \times 10^{-8}$	$1,439 \times 10^{-17}$	—	$1,26 \times 10^{-6}$
Problema 02 - Teste Univariado	3337	24	5,14589	6,145735279	$8,289 \times 10^{-2}$	$-3,492 \times 10^{-7}$	$-1,989 \times 10^{-7}$	$-3,492 \times 10^{-7}$	0	0	$1,989 \times 10^{-7}$	$3,492 \times 10^{-7}$	—	$3,53 \times 10^{-7}$
Problema 07 - Teste Univariado	3229	25	5,19957	5,199776236	$5,611 \times 10^{-2}$	$-7,546 \times 10^{-6}$	$-7,282 \times 10^{-6}$	$-7,546 \times 10^{-6}$	0	0	$7,282 \times 10^{-6}$	$7,546 \times 10^{-6}$	—	$3,19 \times 10^{-7}$
Problema 12 - Teste Univariado	2863	29	3,14368	3,141592681	$2,463 \times 10^{-2}$	$7,234 \times 10^{-13}$	$6,539 \times 10^{-6}$	$7,760 \times 10^{-14}$	0	0	$6,539 \times 10^{-6}$	$7,760 \times 10^{-14}$	—	$6,91 \times 10^{-7}$
Problema 14 - Teste Univariado	2306	34	0,22512	0,224880355	$8,781 \times 10^{-2}$	$-3,874 \times 10^{-7}$	$5,428 \times 10^{-7}$	$-3,569 \times 10^{-7}$	0	0	$5,428 \times 10^{-7}$	$3,569 \times 10^{-7}$	—	$8,03 \times 10^{-8}$

6 CONCLUSÃO E RECOMENDAÇÕES

O propósito desta dissertação foi desenvolver um modelo computacional para fluidos que podem escorrer sobre uma superfície. Tais fluidos são formados por partículas, as quais têm sua dinâmica computada por meio do método numérico SPH. O problema principal abordado foi encontrar mínimos de funções matemáticas complexas, partindo da ideia que um fluido quando jogado em cima de uma superfície em um campo gravitacional irá escorrer e se acomodar nos mínimos da superfície. Numericamente, se tal superfície é uma função de interesse, e se são conhecidas as posições das partículas do fluido, é possível encontrar a partícula na posição de mínimo da função, sendo suas coordenadas a localização do mínimo da função em estudo. Se garantirmos que o fluido é arremessado sobre todo o domínio da superfície, então encontraremos o mínimo global da mesma.

O *framework* PySPH, foi empregado nos experimentos realizados. Tal ferramenta computacional foi primordial para os estudos realizados, pois garantiu o ambiente de desenvolvimento para a realização das simulações dos fluidos sobre as superfícies. De forma geral, os experimentos realizados obtiveram erros absolutos da ordem de 10^{-6} , que numericamente garante um acerto de uma parte em cem mil.

Dada a busca bibliográfica realizada na literatura, não foram encontrados outros trabalhos que utilizam o SPH como método numérico para a determinação de mínimos de funções (ou máximos). Contudo, existem alguns algoritmos que também são baseados em partículas e livres de malhas que também podem ser utilizados para o mesmo fim, como por exemplo, o algoritmo de PSO. Contudo, às interações entre as partículas no PSO é linear, não havendo a ideia de um fluido, mas sim de um bando de pássaros. Desta forma, para a criação de uma base comparativa, o PSO também foi utilizado.

Comparando os valores simulados com o SPH e os calculados pelo PSO é possível verificar que ambos algoritmos são capazes de apontar os mínimos das funções. Em geral os resultados com o PSO foram mais precisos, porém é importante destacar que, embora o custo computacional do PSO seja significativamente menor, o SPH apresenta a vantagem de possuir fundamentação teórica que garante a convergência para o mínimo global no domínio explorado. Assim, mesmo com custos computacionais em tempo e memória mais elevados, o

SPH não deve ser visto apenas como uma limitação, mas como uma alternativa robusta que oferece garantias de otimalidade ausentes em métodos heurísticos como o PSO. Contudo, é possível observar que o conceito de prova que o SPH pode ser empregado para encontrar mínimos de funções é demonstrado, além do fato de que uma vez que existam partículas em todo o domínio de exploração (condição utilizada aqui nos experimentos), há uma justificativa que o SPH irá encontrar o mínimo global da função (no domínio explorado). Tal argumento de otimalidade não pode ser empregado ao algoritmo de PSO, que embora tenha alcançado melhores resultados numéricos em geral é um algoritmo não ótimo por definição, ou seja, não há garantia teórica que o PSO irá encontrar uma solução que represente o mínimo global (dado o domínio de exploração). Além deste fato, a complexidade das funções de *benchmark* utilizadas e a baixa dimensionalidade das mesmas também pode ser um ponto que favoreça um bom desempenho do PSO, novamente devido ao custo computacional. Uma investigação com funções de alta complexidade e alta dimensionalidade necessita ser feita para uma melhor caracterização das condições práticas para a indicação do uso da SPH. Tais investigações são propostas aqui como trabalhos futuros a serem realizados.

Além disso, é importante destacar o ambiente computacional utilizado para a realização dos experimentos. Os testes de maior demanda, que envolveram as funções Perm, Griewank e os Problemas 02, 07, 12 e 14 (todos Testes Univariados), tanto no SPH quanto no PSO, foram executados em dois ambientes distintos. Para experimentos preliminares e de menor complexidade, foi utilizado um computador pessoal equipado com processador AMD Ryzen 5 3500U com Radeon Vega Mobile Gfx de 2,10 GHz e 8 GB de RAM. Já para os experimentos mais pesados, também foi empregado um servidor da universidade, configurado com processador Intel Core i7 (8 núcleos, 3,5 GHz) e 32 GB de RAM. A utilização desses diferentes ambientes reforça que os custos computacionais associados ao SPH tendem a ser mais elevados, principalmente pela necessidade de maior poder de processamento e memória, quando comparado ao PSO.

Assim, além do estudo de funções mais complexas, futuros trabalhos podem ser sugeridos: variações do método SPH, otimizando-o para a dinâmica de escoamento de fluidos em superfícies irregulares e/ou rugosas, propondo mecanismos de auto-adaptação dos parâmetros de *kernel*, por exemplo; aplicações

em problemas dinâmicos, onde a superfície de escoamento não é mais uma constante no tempo. Outra possibilidade é a integração do SPH com redes neurais, utilizando a função de perda da rede como a função objetivo sobre a qual as partículas do SPH interagiriam. Nesse cenário, a evolução da rede seria conduzida pela dinâmica do SPH, tornando o processo de treinamento mais dinâmico e explorando novos caminhos para a busca de mínimos globais.

REFERÊNCIAS

- ADAMI, Stefan; HU, X. Y.; ADAMS, Nikolaus A. A transport-velocity formulation for smoothed particle hydrodynamics. *Journal of Computational Physics*, v. 241, p. 292-307, 2013.
- ANTONY, Justin; MANIYERI, Ranjith. Numerical simulation of fluid flow in a channel using smoothed particle hydrodynamics. In: *Proceedings of 65th Congress of ISTAM*, 2020.
- ANTUONO, M. et al. Smoothed particle hydrodynamics method from a large eddy simulation perspective: generalization to a quasi-Lagrangian model. *Physics of Fluids*, v. 33, n. 1, 2021.
- ARORA, Krishan et al. Optimization methodologies and testing on standard benchmark functions of load frequency control for interconnected multi-area power system in smart grids. *Mathematics*, v. 8, n. 6, p. 980, 2020.
- ASADI, Hossein et al. Investigation of hydrodynamically dominated membrane rupture using smoothed particle hydrodynamics–finite element method. *Fluids*, v. 4, n. 3, p. 149, 2019.
- ATLURI, Satya N.; ZHU, Tulong. A new meshless local Petrov-Galerkin (MLPG) approach in computational mechanics. *Computational mechanics*, v. 22, n. 2, p. 117-127, 1998.
- AZRAG, Mohammed Adam Kunna; KADIR, Tuty Asmawaty Abdul; ALI, Noorlin Mohd. A comparison of particle swarm optimization and global African buffalo optimization. In: *IOP Conference Series: Materials Science and Engineering*. IOP Publishing, 2020. p. 012034.
- BASHIR, Hassan Abdullahi. Diversity control in evolutionary computation using asynchronous dual-populations with search space partitioning. *Nigerian Journal of Technological Development*, v. 17, n. 3, p. 175–188, 2020.
- BELYTSCHKO, Ted; LU, Yun Yun; GU, Lei. Element-free Galerkin methods. *International journal for numerical methods in engineering*, v. 37, n. 2, p. 229-256, 1994.
- CISTERNAS, Luis A.; LUCAY, Freddy A.; BOTERO, Yesica L. Trends in modeling, design, and optimization of multiphase systems in minerals processing. *Minerals*, v. 10, n. 1, p. 22, 2019.
- CLEARY, Paul W. et al. Inclusion of incremental damage breakage of particles and slurry rheology into a particle scale multiphase model of a SAG mill. *Minerals Engineering*, v. 128, p. 92-105, 2018.
- CLEARY, Paul W.; SINNOTT, Matt; MORRISON, Rob. Prediction of slurry transport in SAG mills using SPH fluid flow in a dynamic DEM based porous media. *Minerals engineering*, v. 19, n. 15, p. 1517-1527, 2006.

CUNDALL, P. A.; STRACK, O. D. L. A discrete numerical model for granular assemblies. *Géotechnique*, London, v. 29, n. 1, p. 47–65, 1979.

DE ANDA-SUÁREZ, Juan et al. Evolutionary Gaussian-gradient: a new optimization algorithm for the electromechanical design of gravitational batteries. In: *Hybrid Intelligent Systems Based on Extensions of Fuzzy Logic, Neural Networks and Metaheuristics*. Cham: Springer Nature Switzerland, 2023. p. 347–364.

DOMÍNGUEZ, J. M. et al. Neighbour lists in smoothed particle hydrodynamics. *International Journal for Numerical Methods in Fluids*, v. 67, n. 12, p. 2026–2042, 2011.

DUAN, Shaomi; LUO, Huilong; LIU, Haipeng. A multi-strategy seeker optimization algorithm for optimization constrained engineering problems. *IEEE Access*, v. 10, p. 7165–7195, 2022.

FUCHS, Sebastian L. et al. An SPH framework for fluid–solid and contact interaction problems including thermo-mechanical coupling and reversible phase transitions. *Advanced Modeling and Simulation in Engineering Sciences*, v. 8, n. 1, p. 1–33, 2021.

GAO, Hongwei et al. Robust bacterial foraging algorithms based on few-excellent-individuals guidance strategy. *Sensors & Materials*, v. 32, 2020.

GINGOLD, Robert A.; MONAGHAN, Joseph J. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly notices of the royal astronomical society*, v. 181, n. 3, p. 375–389, 1977.

GOMEZ-GESTEIRA, Moncho et al. State-of-the-art of classical SPH for free-surface flows. *Journal of Hydraulic Research*, v. 48, n. sup1, p. 6–27, 2010.

GRANT-PETERS, Jonathan; HAUSER, Raphael. A seven-point algorithm for piecewise smooth univariate minimization. *arXiv preprint*, arXiv:2012.06553, 2020.

HERNQUIST, Lars; KATZ, Neal. TREESPH-A unification of SPH with the hierarchical tree method. *Astrophysical Journal Supplement Series* (ISSN 0067-0049), vol. 70, June 1989, p. 419–446. Research supported by the San Diego Supercomputer Center, Pittsburgh Supercomputer Center, and New Jersey High Technology Grant., v. 70, p. 419–446, 1989.

HU, D. A. et al. A meshless local Petrov–Galerkin method for large deformation contact analysis of elastomers. *Engineering analysis with boundary elements*, v. 31, n. 7, p. 657–666, 2007.

INFINITY77. A one-dimensional test functions platform with multiple local and global minima. Disponível em: https://infinity77.net/global_optimization/test_functions_1d.html#d-test-functions. Acesso em: 18 dez. 2023.

KASHYAP, Katyayani; SHARMA, Tarun K.; RAJPUROHIT, Jitendra. Logistic map and wavelet transform based differential evolution. *International Journal of System Assurance Engineering and Management*, v. 11, p. 506–514, 2020.

KENNEDY, James; EBERHART, Russell. Particle swarm optimization. In: Proceedings of ICNN'95-international conference on neural networks. iee, 1995. p. 1942-1948.

KOSCHIER, Dan et al. Smoothed particle hydrodynamics techniques for the physics-based simulation of fluids and solids. *arXiv preprint*, arXiv:2009.06944, 2020.

LEVEQUE, Randall J. *Finite Volume Methods for Hyperbolic Problems*. Cambridge: Cambridge University Press, 2002. Disponível em: <https://link.springer.com/book/10.1007/978-0-8176-8394-8>. Acesso em: 22 jun. 2025.

LIND, Steven J.; ROGERS, Benedict D.; STANSBY, Peter K. Review of smoothed particle hydrodynamics: towards converged Lagrangian flow modelling. *Proceedings of the Royal Society A*, v. 476, n. 2241, p. 20190801, 2020.

LIU, Gui-Rong; LIU, Moubin B. Smoothed particle hydrodynamics: a meshfree particle method. World scientific, 2003.

LIU, Wing-Kam; LI, Shaofan; BELYTSCHKO, Ted. Moving least-square reproducing kernel methods (I) methodology and convergence. *Computer methods in applied mechanics and engineering*, v. 143, n. 1-2, p. 113-154, 1997.

LUCY, Leon B. A numerical approach to the testing of the fission hypothesis. *Astronomical Journal*, vol. 82, Dec. 1977, p. 1013-1024., v. 82, p. 1013-1024, 1977.

MIRANDA, J. *pyswarm: Particle Swarm Optimization (PSO) for Python*. Disponível em: <https://github.com/tisimst/pyswarm>. Acesso em: 27 jun. 2025.

MOHAMMADI, Masoud; RIAZI, Masoud. Applicable investigation of SPH in characterization of fluid flow in uniform and non-uniform periodic porous media. *Sustainability*, v. 14, n. 21, p. 14320, 2022.

MONAGHAN, Joe J. Smoothed particle hydrodynamics. In: Annual review of astronomy and astrophysics. Vol. 30 (A93-25826 09-90), p. 543-574., v. 30, p. 543-574, 1992.

MONAGHAN, Joe J. Smoothed particle hydrodynamics. Reports on progress in physics, v. 68, n. 8, p. 1703, 2005.

MORTON, Guy M. A computer oriented geodetic data base and a new technique in file sequencing. 1966.

MOUNTRIS, Konstantinos A. et al. An explicit total Lagrangian fragile points method for finite deformation of hyperelastic materials. *Engineering Analysis with Boundary Elements*, v. 151, p. 255–264, 2023.

MYERS, Conner et al. A hybrid finite volume method and smoothed particle hydrodynamics approach for efficient and accurate blast simulations. *Frontiers in Physics*, v. 11, p. 1325294, 2024.

MYERS, Conner; PALMER, Todd; PALMER, Camille. A hybrid finite volume-smoothed particle hydrodynamics approach for shock capturing applications. *Computer Methods in Applied Mechanics and Engineering*, v. 417, p. 116412, 2023.

PAHLKE, Johannes; SBALZARINI, Ivo F. A unifying mathematical definition of particle methods. *IEEE Open Journal of the Computer Society*, 2023.

RAJA, Vijayakumar et al. Modeling and simulation of 3D food printing systems: scope, advances, and challenges. *Foods*, v. 12, n. 18, p. 3412, 2023.

RAMACHANDRAN, Prabhu et al. PySPH: a Python-based framework for smoothed particle hydrodynamics. *ACM Transactions on Mathematical Software (TOMS)*, v. 47, n. 4, p. 1–38, 2021.

RAMACHANDRAN, Prabhu; PURI, Kunal. Entropically damped artificial compressibility for SPH. *Computers & Fluids*, v. 179, p. 579–594, 2019.

SPRINGEL, Volker. The cosmological simulation code GADGET-2. *Monthly notices of the royal astronomical society*, v. 364, n. 4, p. 1105-1134, 2005.

TSENG, Hsuan-Yu et al. Particle swarm optimization for nonlinear constrained optimization problems. *IEEE Access*, v. 9, p. 124757–124767, 2021.

VACONDIO, Renato et al. Grand challenges for smoothed particle hydrodynamics numerical schemes. *Computational Particle Mechanics*, v. 8, p. 575–588, 2021.

VIOLEAU, Damien. *Fluid mechanics and the SPH method: theory and applications*. 1. ed. Oxford: Oxford University Press, 2012.

WANG, Chau-Shing et al. Process parameter prediction and modeling of laser percussion drilling by artificial neural networks. *Micromachines*, v. 13, n. 4, p. 529, 2022.

WANG, Mengdi et al. Thin-film smoothed particle hydrodynamics fluid. *ACM Transactions on Graphics (TOG)*, v. 40, n. 4, p. 1–16, 2021.

WU, Qiang; PENG, PiaoPiao; CHENG, YuMin. The interpolating element-free Galerkin method for elastic large deformation problems. *Science China Technological Sciences*, v. 64, n. 2, p. 364-374, 2021.

XI, Runping et al. Survey on smoothed particle hydrodynamics and the particle systems. *IEEE Access*, v. 8, p. 3087–3105, 2019.

YE, Ting et al. Smoothed particle hydrodynamics (SPH) for complex fluid flows: recent developments in methodology and applications. *Physics of Fluids*, v. 31, n. 1, 2019.

ZHANG, Tingyu et al. SPH modeling of water-related natural hazards. *Scientific Journal of Technology*, v. 3, n. 9, 2021.