



Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências

Faculdade de Engenharia

Lucas Carvalho de Sousa

**Simulação Numérica De Escoamentos Dispersos em
Turbomáquinas Utilizando o Método De Elementos Finitos**

Rio de Janeiro

2019

Lucas Carvalho de Sousa

**Simulação Numérica De Escoamentos Dispersos em Turbomáquinas
Utilizando o Método De Elementos Finitos**

Projeto Final apresentado a Faculdade de Engenharia da Universidade do Estado do Rio de Janeiro, para obtenção do grau de bacharel em Engenharia Mecânica.

Orientador: Prof. D.Sc. Gustavo R. Anjos

Rio de Janeiro

2019

Ficha elaborada pelo autor através do
Sistema para Geração Automática de Ficha Catalográfica da Rede Sirius - UERJ

Sousa, Lucas Carvalho de
S725 Simulação Numérica De Escoamentos Dispersos em Turbo-
máquinas Utilizando o Método De Elementos Finitos / Lucas
Carvalho de Sousa. – 2019.
 109 f.

Orientador: Gustavo Rabello dos Santos
Trabalho de Conclusão de Curso apresentado à Universidade do Estado do Rio de Janeiro, Faculdade de Engenharia, para obtenção do grau de bacharel em Engenharia Mecânica.

1. Método de Elementos Finitos - Monografias. 2. Formulação Corrente-Vorticidade - Monografias. 3. Escoamento Multifásico - Monografias. 4. Escoamento Particulado - Monografias. 5. Programação Orientada a Objetos - Monografias. I. Anjos, Gustavo Rabello dos. II. Universidade do Estado do Rio de Janeiro. Faculdade de Engenharia. III. Título.

CDU 621

Lucas Carvalho de Sousa

**Simulação Numérica De Escoamentos Dispersos em Turbomáquinas
Utilizando o Método De Elementos Finitos**

Projeto Final apresentado a Faculdade de Engenharia da Universidade do Estado do Rio de Janeiro, para obtenção do grau de bacharel em Engenharia Mecânica.

Aprovado em: 25 de Junho de 2019

Banca Examinadora:

Prof. D.Sc. Gustavo R. Anjos - Orientador
Universidade Federal do Rio de Janeiro - UFRJ - COPPE

Prof. D.Sc. Norberto Mangiavacchi
Departamento de Engenharia Mecânica - UERJ

Prof. D.Sc. Fábio Pereira dos Santos
Universidade Federal do Rio de Janeiro - UFRJ - COPPE

D.Sc. Leon Matos Ribeiro de Lima
Eletrouclear

Rio de Janeiro

2019

DEDICATÓRIA

Dedicado à Luis, Beth, Juliana e Jorge.

AGRADECIMENTO

Agradeço aos meus pais, Luis Sousa e Elizabeth Carvalho, que sempre me apoiaram, serviram de inspiração e como excelentes exemplos que sempre segui com muito orgulho.

A minha companheira, Juliana Marques, pelo carinho e apoio sempre que precisei.

Aos meus orientadores, Gustavo Rabello dos Anjos, pelo apoio e incentivo, e Leandro Marques, pela ajuda na organização, planejamento e revisão do texto.

A meu avô Mario Carvalho, pela inspiração e por me mostrar a engenharia de perto e a minha tia Tathiana Carvalho, pela ajuda com o texto.

Finalmente, aos meus amigos Daniel Coelho, Luís Carnevale, Douglas Lopes, Thiago Cabral e todos os outros que me ajudaram no período da graduação, dentro e fora da sala de aula.

RESUMO

SOUSA, Lucas Carvalho de. *Simulação Numérica De Escoamentos Dispersos em Turbomáquinas Utilizando o Método De Elementos Finitos.* 109 f. Monografia (Projeto Final para Bacharelado em Engenharia Mecânica) - Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro (UERJ), Rio de Janeiro, 2019.

Este trabalho tem como objetivo criar uma biblioteca de simulação numérica de escoamentos particulados bidimensionais na linguagem Python e desenvolvida com o paradigma de programação orientada a objeto e analisar os resultados de casos aplicados à turbomáquinas. Foi utilizado o Método de Elementos Finitos para resolver o sistema de equações obtido da formulação de corrente-vorticidade da equação de Navier-Stokes, tomada para um fluido incompressível e newtoniano. Para a modelagem do comportamento das partículas, foi empregado o Método de Diferenças Finitas para solucionar a equação de Basset–Boussinesq–Oseen, considerando o tipo de interação *one-way*, sem efeito das partículas sobre o escoamento.

Palavras-chave: Método de Elementos Finitos, Formulação Corrente-Vorticidade, Escoamento Multifásico, Escoamento Particulado.

ABSTRACT

This work aims to create a Python library for two-dimensional particle flow simulations that is developed using object oriented programming and analyze the results of cases applied to turbomachines. The Finite Element Method was used to solve the system of equations derived from the stream function-vorticity formulation of the Navier-Stokes equation for a incompressible and newtonian fluid. For the particle behavior modeling, the Finite Difference Method was applied to solve the Basset–Boussinesq–Oseen equation, considering one-way interactions, without taking to account the effect of the particles in the flow.

Keywords: Finite Element Method, Stream Function-Vorticity, Multiphase Flow, Particle Flow.

LISTA DE FIGURAS

Figura 1	Erosão em um impelidor causada pela cavitação.	22
Figura 2	Partícula em um escoamento com um gradiente de velocidade não nulo.	36
Figura 3	Elemento triangular linear.	46
Figura 4	Condições de contorno da placa para o problema de Laplace 5.2.1.	64
Figura 5	Distribuição de temperaturas na placa da solução permanente da equação de Laplace 5.2.1.	65
Figura 6	Comparação de resultado das soluções numérica e analítica do caso de transporte de temperatura em sólidos no regime permanente.	66
Figura 7	Comparação de resultado das soluções numéricas e analítica de transporte em sólidos no regime transiente.	67
Figura 8	Condições de contorno da placa para o problema de Poisson 5.2.2.	68
Figura 9	Distribuição de temperaturas na placa da solução do problema permanente de Poisson 5.2.2.	69
Figura 10	Comparação de resultado das soluções numérica e analítica do problema de transporte de temperatura em sólidos no regime permanente com geração de calor.	70
Figura 11	Comparação de resultado das soluções numéricas e analítica do problema de transporte de temperatura em sólidos no regime transiente com geração de calor.	71
Figura 12	Condições de contorno da placa para o problema de Poisson 5.2.3.	72
Figura 13	Distribuição de temperaturas na placa da solução do problema permanente de Poisson 5.2.3.	73

Figura 14	Comparação de resultado das soluções numérica e analítica do problema de transporte de temperatura em sólidos no regime permanente com geração e fluxo de calor.	74
Figura 15	Comparação de resultado das soluções numéricas e analítica do problema de transporte de temperatura em sólidos no regime permanente com geração e fluxo de calor.	75
Figura 16	Condições de contorno de um escoamento entre placas paralelas de Poiseuille 5.3.1.	76
Figura 17	Comparação de resultado das soluções numéricas e analítica 5.10 do problema de corrente-vorticidade no regime permanente. . .	77
Figura 18	Condições de contorno de um escoamento entre placas paralelas de Couette 5.3.2.	78
Figura 19	Comparação de resultado das soluções numéricas e analítica 5.11 do problema de corrente-vorticidade no regime permanente. . .	79
Figura 20	Condições de contorno de uma partícula isolada sob efeito de uma força.	80
Figura 21	Comparação de resultado da solução numérica e analítica 5.12 do percurso de uma partícula em queda livre.	82
Figura 22	Comparação de resultado da solução numérica e analítica 5.13 do percurso de uma partícula em movimento de arrasto em um escoamento.	84
Figura 23	Condições de contorno de uma partícula isolada sob efeito da força de sustentação.	86
Figura 24	Comparação de resultado da solução numérica e analítica 5.15 do percurso de uma partícula em movimento de sustentação em um escoamento.	87

Figura 25	Comparação de resultado da solução numérica e analítica 5.17 do percurso de uma partícula em movimento de aceleração em um escoamento.	88
Figura 26	Malha da simulação em um canal reto.	89
Figura 27	Malha da simulação em um canal com um obstáculo.	89
Figura 28	Malha da simulação em um canal com degrau.	90
Figura 29	Malha da simulação em um canal com restrição.	90
Figura 30	Malha da simulação em uma palheta de um Impelidor.	91
Figura 31	Campo de velocidades de um escoamento em um canal reto. . .	92
Figura 32	Trajeto de partículas inseridas em um escoamento em um canal reto.	93
Figura 33	Campo de velocidades de um escoamento com obstáculo.	94
Figura 34	Trajeto de partículas inseridas em um escoamento com obstáculo. .	94
Figura 35	Campo de velocidades de um escoamento com degrau.	95
Figura 36	Vetores de velocidades de um escoamento com degrau.	96
Figura 37	Trajeto de partículas inseridas em um escoamento com degrau..	96
Figura 38	Campo de velocidades de um escoamento com restrição.	97
Figura 39	Vetores de velocidades de um escoamento com restrição.	98
Figura 40	Trajeto de partículas inseridas em um escoamento com restrição. .	98
Figura 41	Campo de velocidades de um escoamento em uma palheta de um impelidor.	100
Figura 42	Vetores de velocidades de um escoamento em uma palheta de um impelidor.	100
Figura 43	Trajeto de partículas de ouro inseridas em um escoamento em uma palheta de um impelidor.	101
Figura 44	Trajeto de partículas de ferro inseridas em um escoamento em uma palheta de um impelidor.	102

LISTA DE ABREVIATURAS E SIGLAS

MEF	Método de Elementos Finitos
MDF	Método das Diferenças Finitas
POO	Programação Orientada a Objeto
BBO	Equação de Basset–Boussinesq–Oseen

SUMÁRIO

INTRODUÇÃO	15
1 REVISÃO BIBLIOGRÁFICA	17
1.1 Introdução	17
1.2 Método de Elementos Finitos	17
1.3 Escoamentos Multifásicos	20
1.4 Escoamentos Particulados em Turbomáquinas	21
1.5 Programação Orientada a Objetos	23
2 EQUAÇÕES DE GOVERNO	25
2.1 Introdução	25
2.2 Conservação de Massa	25
2.3 Conservação de Quantidade de Movimento	27
2.4 Formulação Corrente-Vorticidade	31
2.5 Número de Reynolds	33
2.6 Forças Exercidas em Partículas	34
3 MODELAGEM NUMÉRICA	38
3.1 Introdução	38
3.2 Método de Elementos Finitos	38
3.2.1 Formulação Forte	38
3.2.2 Formulação Fraca	39
3.2.3 Discretização Espacial	42
3.2.4 Malha Computacional	45
3.3 Método de Diferenças Finitas	49

3.3.1	Discretização da Corrente-Vorticidade no Tempo	50
3.3.2	Discretização da Forças Aplicadas às Partículas	51
4	CÓDIGO NUMÉRICO	53
4.1	Introdução	53
4.2	Estrutura do Código	53
4.2.1	Importação da Malha	55
4.2.2	Definição das Condições de Contorno	55
4.2.3	Funções de Solução	56
4.2.4	Movimentação das Partículas	58
5	VALIDAÇÃO DO CÓDIGO NÚMÉRICO	61
5.1	Introdução	61
5.2	Validações de Problemas em Sólidos	63
5.2.1	Equação de Laplace com Condições de Contorno de Dirichlet	63
5.2.2	Equação de Poisson com Condições de Contorno de Dirichlet	67
5.2.3	Equação de Poisson com Condições de Contorno de Dirichlet e Neumann	71
5.3	Validações de Problemas em Fluídos	75
5.3.1	Escoamento de Poiseuille	75
5.3.2	Escoamento de Couette	77
5.4	Validações de Problemas em Partículas	79
5.4.1	Força Gravitacional	80
5.4.2	Força de Arrasto	82
5.4.3	Força de Sustentação	84
5.4.4	Força de Massa Virtual (<i>Added Mass</i>)	87

6 RESULTADOS DE SIMULAÇÕES	89
6.1 Introdução	89
6.2 Simulação Em Um Canal Reto	92
6.3 Simulação Em Um Canal Com Obstáculo	93
6.4 Simulação Em Um Canal Com Degrau	95
6.5 Simulação Em Um Canal Com Restrição	97
6.6 Simulação Em Uma Palheta de Impelidor	99
CONCLUSÃO	104
REFERÊNCIAS	106

INTRODUÇÃO

As turbomáquinas são equipamentos utilizados em larga escala por diversas áreas da engenharia para a movimentação e transporte de fluidos. Por isso, é de grande interesse estudar o comportamento dos escoamentos gerados por estas máquinas. Neste trabalho serão analisados escoamentos multifásicos que ocorrem nestes casos, do tipo de interação fluido-sólido. Em especial, busca-se estudar a trajetória de partículas presentes nestes escoamentos durante seu funcionamento para a previsão de possíveis efeitos ou consequências causadas.

Com o aumento do poder de computação das máquinas atuais e a democratização das ferramentas de criação de código, tornou-se mais fácil o desenvolvimento independente de *softwares* de simulação por indivíduos. Utilizando estas ferramentas e o conhecimento de cálculo numérico, este trabalho visa criar uma biblioteca na linguagem de programação *Python* de simulação de escoamentos particulados em diversas geometrias bidimensionais. Esta biblioteca é estruturada utilizando os princípios do paradigma de programação orientada a objetos, para auxiliar o uso intuitivo do código. A solução dos escoamentos é calculada pela formulação da corrente-vorticidade utilizando o Método de Elementos Finitos na forma Euleriana para as variáveis espaciais e o Método de Diferenças Finitas para o termo temporal das equações. Para a simulação do movimento das partículas também será usado o Método de Diferenças Finitas, aplicado a equação Basset–Boussinesq–Oseen para solução das forças aplicadas às mesmas.

Serão gerados resultados para problemas com soluções analíticas conhecidas na literatura e comparados com os valores numéricos para verificação dos cálculos. Em seguida, serão realizadas simulações em diversas geometrias de interesse, e finalmente na geometria da palheta de um rotor, para estudar o comportamento do escoamento e das partículas em turbomáquinas.

Este trabalho será apresentado da seguinte forma:

- Introdução
- Capítulo 1: Revisão bibliográfica dos temas discutidos
- Capítulo 2: Desenvolvimento das equações matemáticas utilizadas
- Capítulo 3: Aplicação dos métodos numéricos às equações do modelo
- Capítulo 4: Descrição da estruturação do código e metodologia de solução
- Capítulo 5: Validação dos resultados com soluções conhecidas
- Capítulo 6: Resultados das simulações dos tópicos de interesse
- Conclusão

1 REVISÃO BIBLIOGRÁFICA

1.1 Introdução

Nesta seção é apresentada a literatura utilizada, analisando-se as partes pertinentes ao trabalho realizado. Os principais tópicos de estudo foram sobre os temas de Método de Elementos Finitos, Escoamentos Multifásicos, Escoamentos Particulados em Turbomáquinas e a Programação Orientada a Objeto.

1.2 Método de Elementos Finitos

A representação matemática de fenômenos físicos é feita por equações, para certos casos esta representação pode ser realizada por uma equação linear simples, em outras situações ela é composta por equações diferenciais.

Infelizmente, nem todas equações possuem soluções analíticas, isto é, soluções contínuas em formas de equações. Nestes casos, uma forma de encontrar algo que represente bem as observações reais realizadas é a solução numérica. A solução numérica é uma aproximação do resultado real feito solucionando o problema em trechos discretos, de forma que quanto menor forem, maior será a precisão do resultado.

Os métodos utilizados neste trabalho são conhecidos como: o Método das Diferenças Finitas (**MDF**) e o Método dos Elementos Finitos (**MEF**).

O Método de Diferenças Finitas foi um dos primeiros métodos numéricos a serem desenvolvidos, apresentado por Courant et. al. (1928) [1], sua metodologia é realizada a partir da discretização da equação de governo do problema em diferenças discretas, dividindo-se o domínio contínuo em regiões pontuais. De forma que quanto menor for o espaçamento entre as regiões do domínio mais preciso o método se tornará. A razão matemática deste mecanismo pode ser interpretada

com a definição da derivada, que busca representar a taxa de variação contínua em um ponto tomando-se o limite da diferença entre o valor da função em dois pontos sobre a distância entre os mesmos, quando esta distância se aproxima a zero. As fórmulas de discretização das equações são baseadas na expansão da Série de Taylor como aplicado por George Boole (1859) [2]. O operador diferencial é substituído por uma soma aproximadamente equivalente, com a ordem de seu erro sendo relativo ao número de termos. Dentro desta substituição podem ser escolhidos vários esquemas, cada um com suas aplicações, vantagens e desvantagens. Os esquemas mais usuais são os de diferenças atrasadas (*Backwards Differences*), diferenças adiantadas (*Forward Differences*) e diferenças centradas (*Central Differences*), onde, em geral, as diferenças centradas possuem melhor aproximação.

Em seguida, os estudos sobre o Método de Elementos Finitos foram iniciados posteriormente e no início de seu desenvolvimento foi voltado para a solução de sistemas mais complexos, que incluíam equações diferenciais parciais. Seus primeiros trabalhos na área da engenharia estudavam as soluções de problemas em sólidos e estruturas em geral como estudado por Turner (1956) [3]. Sua metodologia funciona de forma similar porém, é um processo mais complexo pois trata-se em dividir o domínio em regiões pontuais interligadas, chamadas de nós. As equações do sistema são aplicadas a cada nó e sua influência é transmitida para seus nós vizinhos. Um grupo de nós gera o que é chamado de elemento, estes elementos podem estar organizados com diversas formas geométricas: triângulos, quadriláteros, pentágonos, etc. O espaçamento entre os nós pode ser constante ou não, estruturada e não-estruturada, respectivamente. O domínio onde estes elementos são definidos e estão contidos é chamado de malha.

A metodologia deste método, de acordo com Lewis et. al. (2004) [4], consiste na reformulação da equação de governo no que é chamada de forma fraca. Esta forma é obtida realizando-se primeiro a conversão do problema para a forma

forte, que se trata da avaliação das equações com funções peso atribuídas. Estas funções peso são arbitrárias e são escolhidas de acordo com o método de discretização utilizado. Durante a conversão das equações para a forma forte, surgem as chamadas funções base, de forma ou interpoladoras, estas servem como determinadores das componentes do valor da função em cada nó dentro do elemento. Existem diversos esquemas para a escolha das funções de peso, os mais comuns na literatura são os esquemas de: Galerkin, Taylor-Galerkin e Petrov-Galerkin. Neste trabalho será utilizado o método de Galerkin, que define as funções de peso com mesmo valor das funções de base. Em seguida, a formulação forte é integrada sobre o domínio para se encontrar a forma fraca do problema.

Após encontrar a forma fraca, ela é transformada em sua versão matricial para cada elemento, onde cada índice corresponde a um nó local presente no elemento. Em seguida estas matrizes são adicionadas a uma matriz global que representa a componente de cada nó presente no domínio. Então são aplicadas as condições de contorno e o sistema é resolvido, obtendo-se o valor da variável da equação governante em cada nó da malha.

Dentro da seleção do comportamento da malha pode-se escolher entre esquemas que consideram a malha fixa, chamada de formulação Euleriana, esquemas com a malha móvel, chamada de formulação Lagrangiana, e esquemas mistos como o chamado Lagrangiano-Euleriano Arbitrário (ALE) como explicado por Anjos et. al. (2015) [5]. Neste trabalho será utilizado o esquema de Euler, onde a malha é considerada estacionária e o valor das variáveis de campo de interesse são calculados na posição dos nós da malha. Para a geração dos pontos e elementos da malha será utilizado o *software open source Gmsh*, como sugerido por Geuzaine e Remacle (2009) [22].

1.3 Escoamentos Multifásicos

Escoamentos multifásicos são utilizados largamente na área da engenharia para uma diversidade de aplicações. Estes ocorrem quando há o transporte de mais de uma substância em fases não miscigenadas. Estas fases podem estar ou não no mesmo estado, subdividindo os tipos de escoamento multifásico no tipo de interação entre as fases, líquido-líquido, gás-líquido e sólido-líquido.

Dentro da engenharia mecânica pode-se verificar a grande importância destes escoamentos em casos como a extração de petróleo, onde é injetado um fluido e é captada uma mistura deste com o óleo bruto, e em trocadores de calor que possuem interação entre os fluidos.

Para os escoamentos particulados, sólido-líquido ou sólido-gás com pequenos sólidos chamados de partículas, pode-se notar sua importância até mesmo no transporte de dejetos, na área de saneamento. Alguns exemplos na seção de mecânica incluem o transporte de vapor com condensado, formação de bolhas em bombas e sólidos precipitados.

Um dos primeiros trabalhos sobre este tipo de escoamento foi Baker et al. (1965) [6], sobre o comportamento de escoamentos multifásicos em transportes verticais. Estes usados bastante em trocadores de calor, para melhorar sua eficiência de troca térmica.

Elghobashi et al. (1991) [7] estuda o comportamento de escoamentos particulados demonstrando o efeito da turbulência na simulação de escoamentos multifásicos.

Como apresentado por Balachandar et al. (2010) [8], os valores da fração de volume ocupada pela fase dispersa e a razão entre a massa da fase dispersa e a massa da fase líquida servem como indicadores do nível de interação entre as fases. Para valores muito pequenos, o efeito dominante é do escoamento, portanto, neste caso pode-se considerar apenas os efeitos do fluido sobre as partículas, chamado

de *one-way flow*. Para casos com valores maiores, as partículas tomam um papel mais significativo no escoamento e é preciso fazer uma ligação recíproca entre os mesmos. Portanto, recalcula-se o escoamento considerando os efeitos das partículas no fluido, conhecido como de *two-way flow*. Finalmente, quando estes valores forem mais elevados a fase particulada toma um papel importante no comportamento e torna-se necessário considerar até os efeitos de outras partículas sobre cada uma delas, como colisão, aglomeração e quebra, denominado por Elghobashi et al. (1994) [9] de *four-way flow*.

Para a modelagem das forças atuando sobre cada partícula no escoamento foi utilizada a equação **Basset–Boussinesq–Oseen** (BBO), apresentada por Shao-Lee Soo et al. (1999) [10]. Esta equação é subdividida em várias forças atuantes, como a gravidade, arrasto, massa virtual, entre outras. Entretanto, as equações das forças possuem uma restrição para sua validade, podendo apenas serem aplicadas para casos com baixo número de Reynolds.

1.4 Escoamentos Particulados em Turbomáquinas

Dentro do ciclo de vida de uma turbomáquina, pode-se esperar um certo desgaste devido a pequenas partículas que se chocam contra as paredes durante o movimento do rotor. Este desgaste está ligado as propriedades do escoamento assim como das partículas.

Este efeito está presente até em turbinas de aeronaves, como estudado por Hussein et al. (1973) [11]. Este tipo de escoamento sólido-gás é verificado em locais com altos níveis de poluição. Pode-se encontrar pequenas partículas sólidas presentes no ar ingerido por turbinas industriais e de aeronaves. A sua presença causa um desgaste acelerado nas regiões radiais da turbina, como verificado também por Tabakoff et al. (1986) [12], que apresentou as zonas de maior colisão das partículas. Estes trabalhos lidam diretamente com os efeitos da colisão das partículas,

utilizando modelos que representam o comportamento delas após o choque.

Entrando na área de sólido-líquido, Uzol et al. (2002) [13] fez um estudo mostrando o escoamento de uma turbomáquina utilizando partículas inseridas e um velocímetro de imagem para acompanhar seu trajeto. Este trabalho não estuda o especificamente um escoamento particulado experimental, porém, utiliza alguns exemplos como ferramenta para visualizar seu comportamento para casos em geral.

Em Ghenaiet et al. (2005) [14] é estudado com mais profundidade os efeitos dos escoamentos particulados em turbomáquinas com líquidos. Neste caso, é estudado os efeitos da degradação causada por partículas de areia ingeridas pelo escoamento no desempenho de uma turbomáquina axial. Foi criado um modelo preditivo e demonstrada uma correlação entre o tamanho das partículas e a velocidade da degradação causada.

A causa mais comum de erosão em impelidores de turbomáquinas trabalhando com líquidos é a cavitação. A cavitação é a formação de bolhas de ar devido a uma queda local na pressão que logo após serem geradas implodem, gerando ondas de vibração e podendo danificar locais próximos. A Figura 1 mostra os efeitos do desgaste causado pelo efeito da cavitação em uma bomba.



Fonte: John Anspach Consulting [15].

Figura 1: Erosão em um impelidor causada pela cavitação.

Blake et al. (1987) [16] estudou o comportamento da cavitação tomando o ponto de vista das bolhas criadas como partículas no escoamento. Através desta interpretação, pode-se realizar simulações dos efeitos da cavitação com modelos de escoamentos particulados.

1.5 Programação Orientada a Objetos

A Programação Orientada a Objetos (**POO**), termo foi popularizado por Alan Kay, originou na década de 60 como forma de representar as informações interpretadas pelo sistema de maneira análoga às experiências de vida humana. Ela é um paradigma, isto é, uma forma diferente de se interpretar a programação.

Como apresentado por Snyder (1986) [17], esta metodologia de programação visa representar estruturas complexas através de classes, também conhecidas como tipos, de variáveis. Cada classe representa a ideia de um objeto, suas características, mas não representa o objeto em si, como uma receita de comportamentos do mesmo. As classes podem possuir funções que são executadas por um elemento dela, estas funções são chamadas de métodos. Um método especial, presente em todas as classes, é chamado de construtor, este método inicializa os objetos criados desta classe com valores padrões para suas propriedades ou com os valores definidos, caso tenham sido especificados. As representações dos objetos são chamadas de instâncias.

A principal vantagem da POO é a facilidade que esta proporciona na abstração de problemas em analogias do mundo físico. Neste trabalho, por exemplo, são criadas classes para a malha, a partícula, as condições de contorno, entre outras. Desta forma, pode-se visualizar o comportamento esperado entre as partículas e a malha apenas se lendo o código. Este paradigma permite maior legibilidade de código e facilita a aplicação do mesmo como biblioteca quando importado.

Outra grande otimização trazida pela POO é a facilidade de reaproveita-

mento de código pois, cada instância pode chamar o mesmo método definido no mesmo lugar. Além disso, ela fortalece o uso da tipagem na programação. Linguagens tipadas, fraca ou fortemente, são linguagens onde seus métodos e funções que requerem que variáveis coincidam com os tipos declarados para que sejam executadas. Linguagens fortemente tipadas tendem a nem compilar caso alguma variável não corresponda ao tipo correto em algum local no código.

Neste trabalho foi utilizado a linguagem *Python*, criada por Rossum (1995) [18]. Python é uma linguagem fracamente tipada, ou seja, há erros devido a tipos não correspondentes, entretanto, ela aceita que funções chamem variáveis de qualquer tipo, até que ocorra um erro. Um dos motivos por ter sido escolhida é sua natureza *Open Source*, de uso livre e sem restrição de licença. Com isso, há um grande número e variedade de bibliotecas livres para serem utilizadas. Desta forma, facilita-se severamente o desenvolvimento do código.

2 EQUAÇÕES DE GOVERNO

2.1 Introdução

Este trabalho apresenta a modelagem de fluidos e partículas em um sistema de escoamento multifásico, portanto, é preciso definir o que é um escoamento e quais as suas restrições para esta situação. Um escoamento é o movimento das moléculas de um fluido em conjunto. As moléculas são tomadas como elementos infinitesimais, porém, são tratadas de forma que não haja espaços vazios entre elas. Isto permite que as propriedades do fluido sejam tratadas pontualmente, podendo variar, por exemplo, sua densidade ou velocidade de nó a nó. Pode-se então modelar o comportamento destes escoamentos seguindo as equações de conservação:

- Conservação de Massa
- Conservação de Quantidade de Movimento

2.2 Conservação de Massa

O princípio de conservação de massa sem geração descreve que, dentro de um volume de controle, a soma da **taxa de acúmulo de massa dentro do volume com o fluxo de massa que atravessa a fronteira do volume** é nula como demonstrado por Pontes e Norberto (2010) [19].

O acúmulo de massa dentro do volume de controle é definido como:

$$\int_{V_c} \frac{\partial}{\partial t} dm = \int_{V_c} \frac{\partial}{\partial t} (\rho_f dV_c) \quad (2.1)$$

$$m = \rho_f V_c \quad (2.2)$$

onde V_c é o volume de controle, dm é o elemento infinitesimal de massa e ρ_f é a massa específica do fluido.

Simplificando a equação de acúmulo de massa (Eq. 2.1), tomada para um volume de controle permanente, que não varia no tempo, tem-se:

$$\int_{V_c} \frac{\partial}{\partial t} (\rho_f dV_c) = \int_{V_c} \frac{\partial \rho_f}{\partial t} dV_c + \int_{V_c} \rho_f \frac{\partial dV_c}{\partial t} = \int_{V_c} \frac{\partial \rho_f}{\partial t} dV_c \quad (2.3)$$

E o fluxo de massa que atravessa a fronteira é retratado como:

$$\oint_S \rho_f \vec{v}_f \cdot \vec{n} dA \quad (2.4)$$

onde S é a curva de contorno da fronteira do volume de controle, \vec{v}_f é o campo de velocidades do fluido, dA é o elemento infinitesimal de área da superfície de contorno do volume de controle e \vec{n} é um vetor normal unitário orientado para fora do contorno S .

A conservação de massa é então representada como:

$$\int_{V_c} \frac{\partial \rho_f}{\partial t} dV_c + \oint_S \rho_f \vec{v}_f \cdot \vec{n} dA = 0 \quad (2.5)$$

Pode-se rescrever a equação de conservação de massa aplicando-se o *Theorema de Gauss*, apresentado por Pontes e Norberto (2010) [19], na integral de superfície:

$$\int_{V_c} \frac{\partial \rho_f}{\partial t} dV_c + \int_{V_c} \vec{\nabla} \cdot (\rho_f \vec{v}_f) dV_c = \int_{V_c} \left(\frac{\partial \rho_f}{\partial t} + \vec{\nabla} \cdot (\rho_f \vec{v}_f) \right) dV_c = 0 \quad (2.6)$$

obtendo-se a equação integral da conservação de massa, onde $\vec{\nabla}$ é o operador diferencial gradiente de componentes $\vec{\nabla} = \left(\frac{\partial}{\partial i}, \frac{\partial}{\partial j}, \frac{\partial}{\partial k} \right)$.

Ao considerar a conservação do ponto de vista local, pode-se remover o

termo integral e escrever a forma diferencial da conservação de massa:

$$\frac{\partial \rho_f}{\partial t} + \vec{\nabla} \cdot (\rho_f \vec{v}_f) = 0 \quad (2.7)$$

Esta equação é denominada *Equação da Continuidade*, e pode ser reescrita como:

$$\frac{\partial \rho_f}{\partial t} + \vec{v}_f \cdot \vec{\nabla} \rho_f + \rho_f \vec{\nabla} \cdot \vec{v}_f = 0 \quad (2.8)$$

Tomando-se algumas hipóteses, é possível simplificar mais a equação. Para um fluido incompressível, com massa específica invariante na posição e no tempo, a equação da continuidade pode ser escrita como:

$$\rho_f \vec{\nabla} \cdot \vec{v}_f = 0 \quad (2.9)$$

Como a massa específica do fluido não pode ser nula, tem-se:

$$\vec{\nabla} \cdot \vec{v}_f = 0 \quad (2.10)$$

tomada para um **fluido incompressível**.

2.3 Conservação de Quantidade de Movimento

A conservação de quantidade de movimento é similar a conservação de massa, porém, é um termo vetorial. Para o caso deste trabalho, é utilizada a versão linear da conservação de quantidade de movimento. Portanto, tem-se que a conservação da quantidade de movimento determina que a **taxa de acúmulo de quantidade de movimento linear dentro do volume de controle** mais **o fluxo de quantidade de movimento linear que atravessa a fronteira do volume de controle** é igual a **soma das forças aplicadas à superfície da**

fronteira do volume de controle e as forças do volume.

A definição da taxa de acúmulo de quantidade de movimento linear dentro do volume de controle é:

$$\int_{V_c} \frac{\partial}{\partial t} (\rho_f \vec{v}_f) dV_c \quad (2.11)$$

E o fluxo de quantidade de movimento que atravessa a fronteira é retratado como:

$$\oint_S \rho_f \vec{v}_f \vec{v}_f \cdot \vec{n} dA \quad (2.12)$$

As forças aplicadas à superfície da fronteira do volume de controle é:

$$\oint_S \boldsymbol{\sigma} \cdot \vec{n} dA \quad (2.13)$$

onde $\boldsymbol{\sigma}$ é o tensor de tensões.

E as forças de volume são, tomada apenas a força gravitacional:

$$\int_{V_c} \rho_f \vec{g} dV_c \quad (2.14)$$

onde \vec{g} é a aceleração gravitacional presente.

Montando-se a equação, a conservação de quantidade de movimento é representada como:

$$\int_{V_c} \frac{\partial}{\partial t} (\rho_f \vec{v}_f) dV_c + \oint_S \rho_f \vec{v}_f \vec{v}_f \cdot \vec{n} dA = \oint_S \boldsymbol{\sigma} \cdot \vec{n} dA + \int_{V_c} \rho_f \vec{g} dV_c \quad (2.15)$$

Novamente, aplica-se o *Teorema de Gauss* nas integrais de superfície e extrai-se a forma integral da equação da conservação de quantidade de movimento:

$$\int_{V_c} \frac{\partial}{\partial t} (\rho_f \vec{v}_f) dV_c + \int_{V_c} \vec{\nabla} \cdot (\rho_f \vec{v}_f \vec{v}_f) dV_c = \int_{V_c} \vec{\nabla} \cdot \boldsymbol{\sigma} dV_c + \int_{V_c} \rho_f \vec{g} dV_c \quad (2.16)$$

Simplificando:

$$\int_{V_c} \left(\frac{\partial}{\partial t} (\rho_f \vec{v}_f) + \vec{\nabla} \cdot (\rho_f \vec{v}_f \cdot \vec{v}_f) - \vec{\nabla} \cdot \boldsymbol{\sigma} - \rho_f \vec{g} \right) = 0 \quad (2.17)$$

Novamente, considerando-se a conservação do ponto de vista pontual, remove-se o termo integral para escrever a forma diferencial da conservação de quantidade de movimento:

$$\frac{\partial}{\partial t} (\rho_f \vec{v}_f) + \vec{\nabla} \cdot (\rho_f \vec{v}_f \cdot \vec{v}_f) = \vec{\nabla} \cdot \boldsymbol{\sigma} + \rho_f \vec{g} \quad (2.18)$$

Continuando a desenvolver a equação:

$$\frac{\partial}{\partial t} (\rho_f \vec{v}_f) + \vec{\nabla} \cdot (\rho_f \vec{v}_f \cdot \vec{v}_f) = \rho_f \frac{\partial \vec{v}_f}{\partial t} + \vec{v}_f \frac{\partial \rho_f}{\partial t} + \rho_f \vec{v}_f \cdot \vec{\nabla} \cdot \vec{v}_f + \vec{v}_f \cdot \vec{\nabla} \cdot (\rho_f \vec{v}_f) \quad (2.19)$$

$$\rho_f \left(\frac{\partial \vec{v}_f}{\partial t} + \vec{v}_f \cdot \vec{\nabla} \cdot \vec{v}_f \right) + \vec{v}_f \left(\frac{\partial \rho_f}{\partial t} + \vec{\nabla} \cdot (\rho_f \vec{v}_f) \right) \quad (2.20)$$

Novamente, é tomada a hipótese de um fluido incompressível. Portanto, a equação pode ser simplificada para ser escrita como:

$$\rho_f \left(\frac{\partial \vec{v}_f}{\partial t} + \vec{v}_f \cdot \vec{\nabla} \cdot \vec{v}_f \right) = \vec{\nabla} \cdot \boldsymbol{\sigma} + \rho_f \vec{g} \quad (2.21)$$

Reescrevendo o tensor de tensões como uma soma de dois tensores, e o substituindo na equação, tem-se:

$$\boldsymbol{\sigma} = -p \mathbf{I} + \boldsymbol{\tau} \quad (2.22)$$

$$\rho_f \left(\frac{\partial \vec{v}_f}{\partial t} + \vec{v}_f \cdot \vec{\nabla} \cdot \vec{v}_f \right) = -\vec{\nabla} p + \vec{\nabla} \cdot \boldsymbol{\tau} + \rho_f \vec{g} \quad (2.23)$$

onde p é o campo de pressões no fluido, \mathbf{I} é a matriz de identidade e $\boldsymbol{\tau}$ é o tensor de tensões viscosas.

Novamente é necessário fazer uma hipótese para este escoamento, para que

seja possível definir as forças atuantes no fluido. O tensor de tensões viscosas $\boldsymbol{\tau}$ está relacionado as propriedades do fluido, podendo ser definido matematicamente para um fluido homogêneo, isotrópico e newtoniano, como:

$$\boldsymbol{\tau} = \mu_f \left(\vec{\nabla} \cdot \vec{v}_f + (\vec{\nabla} \cdot \vec{v}_f)^T \right) \quad (2.24)$$

onde μ_f é a viscosidade dinâmica do fluido.

Substituindo a definição de $\boldsymbol{\tau}$ na Eq. 2.23, obtém-se:

$$\rho_f \left(\frac{\partial \vec{v}_f}{\partial t} + \vec{v}_f \cdot \vec{\nabla} \cdot \vec{v}_f \right) = -\vec{\nabla} p + \vec{\nabla} \cdot \left(\mu_f \left(\vec{\nabla} \cdot \vec{v}_f + (\vec{\nabla} \cdot \vec{v}_f)^T \right) \right) + \rho_f \vec{g} \quad (2.25)$$

Assumindo-se que a viscosidade dinâmica é constante para todo o fluido:

$$\rho_f \left(\frac{\partial \vec{v}_f}{\partial t} + \vec{v}_f \cdot \vec{\nabla} \cdot \vec{v}_f \right) = -\vec{\nabla} p + \mu_f \vec{\nabla} \cdot \left(\vec{\nabla} \cdot \vec{v}_f + (\vec{\nabla} \cdot \vec{v}_f)^T \right) + \rho_f \vec{g} \quad (2.26)$$

$$\rho_f \left(\frac{\partial \vec{v}_f}{\partial t} + \vec{v}_f \cdot \vec{\nabla} \cdot \vec{v}_f \right) = -\vec{\nabla} p + \mu_f \left(\nabla^2 \vec{v}_f + \vec{\nabla} \cdot (\vec{\nabla} \cdot \vec{v}_f)^T \right) + \rho_f \vec{g} \quad (2.27)$$

Utilizando o que foi obtido na Equação de Continuidade Eq. 2.10 pode-se substituir na equação Eq. 2.27:

$$\rho_f \left(\frac{\partial \vec{v}_f}{\partial t} + \vec{v}_f \cdot \vec{\nabla} \cdot \vec{v}_f \right) = -\vec{\nabla} p + \mu_f \nabla^2 \vec{v}_f + \rho_f \vec{g} \quad (2.28)$$

Dividindo-se todos os termos pela massa específica pode-se reescrever a equação Eq. 2.28, resultando-se na forma simplificada da *Equação de Navier-Stokes* para fluidos newtonianos incompressíveis e com viscosidade constante:

$$\frac{\partial \vec{v}_f}{\partial t} + \vec{v}_f \cdot \vec{\nabla} \cdot \vec{v}_f = -\frac{1}{\rho_f} \vec{\nabla} p + \frac{\mu_f}{\rho_f} \nabla^2 \vec{v}_f + \vec{g} \quad (2.29)$$

2.4 Formulação Corrente-Vorticidade

A formulação de corrente-vorticidade é um sistema de equações que provê um método alternativo de se calcular as propriedades de um escoamento sem solucionar diretamente a equação de Navier-Stokes (Eq. 2.29). Isto permite simplificar a solução do problema, pois a Equação de Navier-Stokes possui um forte acoplamento entre o campo de pressões e o campo de velocidades.

Para isso, é utilizada a seguinte identidade vetorial:

$$\vec{v} \cdot \vec{\nabla} \cdot \vec{v} = \vec{\nabla} \frac{v^2}{2} - \vec{v} \times \vec{\nabla} \times \vec{v} \quad (2.30)$$

Substituindo na Eq. 2.29:

$$\frac{\partial \vec{v}_f}{\partial t} + \vec{\nabla} \frac{v_f^2}{2} - \vec{v}_f \times \vec{\nabla} \times \vec{v}_f = -\frac{1}{\rho_f} \vec{\nabla} p + \frac{\mu_f}{\rho_f} \nabla^2 \vec{v}_f + \vec{g} \quad (2.31)$$

Em seguida, aplica-se o operador rotacional nos dois lados da equação:

$$\vec{\nabla} \times \frac{\partial \vec{v}_f}{\partial t} + \vec{\nabla} \times \vec{\nabla} \frac{v_f^2}{2} - \vec{\nabla} \times \vec{v}_f \times \vec{\nabla} \times \vec{v}_f = -\vec{\nabla} \times \frac{1}{\rho_f} \vec{\nabla} p + \vec{\nabla} \times \frac{\mu_f}{\rho_f} \nabla^2 \vec{v}_f + \vec{\nabla} \times \vec{g} \quad (2.32)$$

Simplifica-se então a equação, onde os termos que possuem o operador gradiente são anulados, pois o rotacional do gradiente de um escalar é zero. O termo gravitacional também é anulado já que a derivada da constante \vec{g} é zero. Fica-se com a seguinte equação:

$$\frac{\partial}{\partial t} (\vec{\nabla} \times \vec{v}_f) - \vec{\nabla} \times \vec{v}_f \times \vec{\nabla} \times \vec{v}_f = \frac{\mu_f}{\rho_f} \nabla^2 (\vec{\nabla} \times \vec{v}_f) \quad (2.33)$$

Define-se o vetor $\vec{\omega}$ como *vorticidade*, onde $\vec{\omega} = \vec{\nabla} \times \vec{v}_f$. Substitui-se na

equação:

$$\frac{\partial \vec{\omega}}{\partial t} - \vec{\nabla} \times \vec{v}_f \times \vec{\omega} = \frac{\mu_f}{\rho_f} \nabla^2 \vec{\omega} \quad (2.34)$$

Rearranjando-se as operações vetoriais, tem-se:

$$\frac{\partial \vec{\omega}}{\partial t} + \vec{v}_f \cdot \vec{\nabla} \cdot \vec{\omega} - \vec{\omega} \cdot \vec{\nabla} \cdot \vec{v}_f = \frac{\mu_f}{\rho_f} \nabla^2 \vec{\omega} \quad (2.35)$$

Como a vorticidade é perpendicular ao vetor velocidade, para escoamentos bidimensionais pode-se anular o produto $\vec{\omega} \cdot \vec{\nabla} \cdot \vec{v}_f$ como demonstrado por Pontes e Norberto (2010) [19]. Então encontra-se a *Equação da Vorticidade* para escoamentos de fluidos newtonianos incompressíveis:

$$\frac{\partial \vec{\omega}}{\partial t} + \vec{v}_f \cdot \vec{\nabla} \cdot \vec{\omega} = \frac{\mu_f}{\rho_f} \nabla^2 \vec{\omega} \quad (2.36)$$

Para escoamentos permanentes bidimensionais de fluidos incompressíveis, a velocidade é calculada pela vazão volumétrica. Portanto, a velocidade pode ser substituída por um escalar ψ , conhecido como *função corrente*. A relação entre a função corrente e o campo de velocidades do fluido é obtida através da manipulação da equação da continuidade (Eq. 2.8), dada como:

$$\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} = 0 \quad (2.37)$$

onde v_x é a componente do campo de velocidades do fluido no eixo x e v_y é a componente do campo de velocidades do fluido no eixo y .

E a relação entre elas é apresentada em seguida:

$$v_x = \frac{\partial \psi}{\partial y} \quad (2.38)$$

$$v_y = -\frac{\partial \psi}{\partial x} \quad (2.39)$$

Enquanto a relação entre a função corrente e a vorticidade é:

$$(\vec{\nabla} \times \vec{v}_f)_z = \frac{\partial v_y}{\partial x} - \frac{\partial v_x}{\partial y} \quad (2.40)$$

Substituindo os termos de velocidade pelas equações Eq. 2.38 e Eq. 2.39:

$$\omega_z = -\frac{\partial}{\partial x} \frac{\partial \psi}{\partial x} - \frac{\partial}{\partial y} \frac{\partial \psi}{\partial y} \quad (2.41)$$

$$\omega_z = -\nabla^2 \psi \quad (2.42)$$

Então o sistema de corrente-vorticidade é apresentado como:

$$\frac{\partial \vec{\omega}}{\partial t} + \vec{v}_f \cdot \vec{\nabla} \cdot \vec{\omega} = \frac{\mu_f}{\rho_f} \nabla^2 \vec{\omega} \quad (2.43)$$

$$\nabla^2 \psi = -\omega_z \quad (2.44)$$

$$\vec{v}_f = \left(\frac{\partial \psi}{\partial y}, -\frac{\partial \psi}{\partial x} \right) \quad (2.45)$$

2.5 Número de Reynolds

Para analisar as características de escoamentos com diversas variáveis, utiliza-se Re , uma constante adimensional conhecida como **Número de Reynolds**.

O número de Reynolds permite reconhecer rapidamente as qualidades de um escoamento apenas por seu valor. Ele estabelece uma relação entre as forças viscósas e inerciais de um fluido. Pode-se declarar então qual força possui maior componente no comportamento do escoamento.

O número de Reynolds de um escoamento é calculado por:

$$Re = \frac{\rho_f |\vec{v}_f|_{max} D}{\mu_f} \quad (2.46)$$

onde v_{fmax} é a maior velocidade do campo de velocidades do fluido, e D é o comprimento característico de problema.

2.6 Forças Exercidas em Partículas

A principal equação que governa o comportamento do movimento das partículas é obtida a partir da 2^a Lei de Newton como demonstrado por Crowe et al. (2011) [20].

$$\sum \vec{F}_p = \frac{d(m_p \vec{v}_p)}{dt} \quad (2.47)$$

onde $\sum \vec{F}_p$ é o somatório de forças aplicadas a partícula, m_p é a massa e \vec{v}_p a velocidade da partícula.

As forças aplicadas às partículas foram determinadas a partir da equação de **Basset–Boussinesq–Oseen** (BBO), que descreve as forças sofridas por partículas sob efeito de escoamentos com número de Reynolds 2.5 pequeno. Porém, o número de Reynolds utilizados é diferente do calculado para escoamentos, utiliza-se um valor específico para cada partícula.

$$Re_p = \frac{\rho_p |(\vec{v}_f - \vec{v}_p)|_{max} d_p}{\mu_f} \quad (2.48)$$

onde ρ_p é a densidade, d_p o diâmetro e \vec{v}_p a velocidade da partícula, e \vec{v}_f é a velocidade do escoamento na mesma posição da partícula.

Como mencionado, as forças da equação de BBO representam fielmente o comportamento real de partículas para valores de Reynolds menores que 1, ou seja, $Re_p < 1$.

Nem todas as forças apresentadas na equação BBO foram inseridas no código de simulação. Elas foram inseridas sequencialmente durante o desenvolvimento do código, seguidas de testes para verificar sua implementação. Foram

escolhidas apenas as principais forças, apresentadas a seguir, por Crowe et al. (2011) [20]:

- **Força Gravitacional:** Também conhecida como força Peso, criada pelo campo gravitacional do planeta aplicado aos corpos com massa. Seu valor é calculado através das massas dos corpos, a distância entre seus centros de massa e uma constante universal. Porém, para simplificar o problema, pode-se assumir que a distância entre eles é constante e igual ao raio do planeta Terra. Portanto, pode-se assumir que o valor da força será sempre constante e proporcional a uma aceleração constante aplicada a massa da partícula, o que torna esta a força mais simples de se calcular:

$$\vec{F}_{grav} = m_p \vec{g} \quad (2.49)$$

onde \vec{g} é tomada como a aceleração constante da gravidade na superfície terrestre com valor definido de $\vec{g} = (0, -9.80665)m/s^2$.

- **Força de Arrasto:** Força causada pelo atrito do fluido no contorno da partícula, que cria um efeito cisalhante e tende a deixar a partícula com mesma velocidade que o fluido.

Sua equação é determinada como:

$$\vec{F}_{drag} = 3\pi\mu_f d_p (\vec{v}_f - \vec{v}_p) \quad (2.50)$$

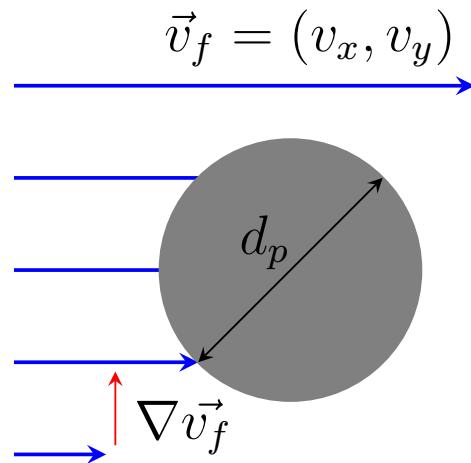
Pode-se observar seu comportamento presente no termo de velocidade relativa $(\vec{v}_f - \vec{v}_p)$ da equação Eq. 2.50. Este termo causa o efeito de aproximar a velocidade da partícula à velocidade do fluido já que torna a força proporcionalmente maior quanto mais afastadas elas se encontrarem. E o sinal permite que caso a partícula esteja mais rápida ela seja desacelerada e o

inverso se ela se encontrar mais lenta.

- **Força de Sustentação:** A força de sustentação é uma componente que tende a mover a partícula em direções perpendiculares a seu movimento principal. Ela possui várias formas de ser calculada, sendo nomeada para cada caso.

Por exemplo, a **Força de Magnus** está relacionada a rotação, ou velocidade angular, presente na partícula. Neste trabalho as partículas são assumidas como corpos rígidos sem rotação, então esta força não será considerada.

Porém, outro tipo de força de sustentação é a **Força de Saffman** explorada por Crowe et al. (2011) [20], que é gerada a partir do gradiente de velocidades presente no escoamento. Para que haja o efeito da força de sustentação de Saffman deve existir um gradiente no campo de velocidades do fluido diferente de nulo, como pode se notar na Figura 2.



Fonte: Autor.

Figura 2: Partícula em um escoamento com um gradiente de velocidade não nulo.

A força é aplicada em uma direção perpendicular ao movimento do escoa-

mento e proporcional ao crescimento da velocidade no escoamento. Ela pode ser calculada através da equação:

$$\vec{F}_{lift} = 1.61 \mu_f d_p (\vec{v}_f - \vec{v}_p) \sqrt{Re_G} \quad (2.51)$$

onde Re_G é conhecido como número de Reynolds de cisalhamento, e é calculado através da equação:

$$Re_G = \frac{d_p^2 \rho_f}{\mu_f} \nabla \vec{v}_f \quad (2.52)$$

- **Força de Massa Virtual (*Added Mass*):** A força de massa virtual está relacionada ao trabalho realizado pelo fluido para acelerar um corpo submerso. Isto pode ser interpretado como a energia que seria utilizada para mover uma mesma quantidade de fluido no lugar do corpo presente. O valor desta força pode ser calculado através da equação:

$$\vec{F}_{mass} = \frac{1}{2} \rho_f V_p \frac{d}{dt} (\vec{v}_f - \vec{v}_p) \quad (2.53)$$

onde V_p é o volume da partícula inserida no escoamento.

3 MODELAGEM NUMÉRICA

3.1 Introdução

Neste trabalho foram utilizados dois tipos de modelagem numéricas para simular o comportamento do sistema multifásico. Na fase dos fluidos, foi utilizado o *Método de Elementos Finitos* (MEF) para solucionar as equações de governo, pois ele proporciona uma forma eficiente de solucionar as equações com rápida convergência. Enquanto para a fase sólida das partículas, e o termo temporal das equações dos fluidos, foi utilizado o *Método de Diferenças Finitas* (MDF), o qual foi escolhido por sua simplicidade de implementação tomando-se cuidado com suas restrições de convergência.

São tomadas as hipóteses do fluido como um meio contínuo, fluido incompressível, e das partículas perfeitamente rígidas.

3.2 Método de Elementos Finitos

3.2.1 Formulação Forte

A formulação forte são as equações de governo do problema na sua forma diferencial, com as condições de contorno definidas. As equações do fluido, definidas em Eq. 2.43, Eq. 2.44 e Eq. 2.45 são tomadas no domínio $\Omega \subset \mathbb{R}^2$ com condições de contorno definidas em:

$$\omega = \omega_\Gamma \text{ em } \Gamma_1 \quad (3.1)$$

$$\psi = \psi_\Gamma \text{ em } \Gamma_2 \quad (3.2)$$

$$\vec{v}_f = \vec{v}_{f\Gamma} \text{ em } \Gamma_3 \quad (3.3)$$

3.2.2 Formulação Fraca

A formulação fraca é o resultado da ponderação da equação da forma forte integrada sobre o domínio. Para o encontrar as formas fracas das equações de governo tomadas neste trabalho, inicialmente são estabelecidos resíduos \vec{R}_i nas equações de forma forte:

$$\frac{\partial \vec{\omega}}{\partial t} + \vec{v}_f \cdot \vec{\nabla} \cdot \vec{\omega} - \frac{\mu_f}{\rho_f} \nabla^2 \vec{\omega} = \vec{R}_1 \quad (3.4)$$

$$\nabla^2 \psi + \omega_z = \vec{R}_2 \quad (3.5)$$

$$\vec{v}_f - \left(\frac{\partial \psi}{\partial y}, - \frac{\partial \psi}{\partial x} \right) = \vec{R}_3 \quad (3.6)$$

Em seguida, busca-se impor o valor médio de cada resíduo como nulo, de forma que:

$$\int_{\Omega} \vec{R}_1 \cdot \vec{\delta} d\Omega = 0 \quad (3.7)$$

$$\int_{\Omega} \vec{R}_2 \cdot \vec{\phi} d\Omega = 0 \quad (3.8)$$

$$\int_{\Omega} \vec{R}_3 \cdot \vec{\xi} d\Omega = 0 \quad (3.9)$$

onde $\vec{\delta}$, $\vec{\phi}$ e $\vec{\xi}$ são as funções de peso de cada equação, respectivamente. As funções peso são funções arbitrárias utilizadas para obter as componentes de contribuição de cada nó.

Substituindo-se os resíduos nas integrais, tem-se:

$$\int_{\Omega} \left(\frac{\partial \vec{\omega}}{\partial t} + \vec{v}_f \cdot \vec{\nabla} \cdot \vec{\omega} - \frac{\mu_f}{\rho_f} \nabla^2 \vec{\omega} \right) \cdot \vec{\delta} d\Omega = 0 \quad (3.10)$$

$$\int_{\Omega} (\nabla^2 \psi + \omega_z) \cdot \vec{\phi} d\Omega = 0 \quad (3.11)$$

$$\int_{\Omega} \left(\vec{v}_f - \left(\frac{\partial \psi}{\partial y}, -\frac{\partial \psi}{\partial x} \right) \right) \cdot \vec{\xi} d\Omega = 0 \quad (3.12)$$

Reorganizam-se as integrais:

$$\int_{\Omega} \frac{\partial \vec{\omega}}{\partial t} \cdot \vec{\delta} d\Omega + \int_{\Omega} \vec{v}_f \cdot \vec{\nabla} \cdot \vec{\omega} \cdot \vec{\delta} d\Omega - \int_{\Omega} \frac{\mu_f}{\rho_f} \nabla^2 \vec{\omega} \cdot \vec{\delta} d\Omega = 0 \quad (3.13)$$

$$\int_{\Omega} \nabla^2 \psi \cdot \vec{\phi} d\Omega + \int_{\Omega} \omega_z \cdot \vec{\phi} d\Omega = 0 \quad (3.14)$$

$$\int_{\Omega} \vec{v}_f \cdot \vec{\xi} d\Omega - \int_{\Omega} \left(\frac{\partial \psi}{\partial y}, -\frac{\partial \psi}{\partial x} \right) \cdot \vec{\xi} d\Omega = 0 \quad (3.15)$$

Aplica-se agora o Teorema de Green nos termos difusivos das equações:

$$-\int_{\Omega} \frac{\mu_f}{\rho_f} \nabla^2 \vec{\omega} \cdot \vec{\delta} d\Omega = \int_{\Omega} \frac{\mu_f}{\rho_f} \vec{\nabla} \cdot \vec{\omega} \cdot \vec{\nabla} \cdot \vec{\delta} d\Omega - \int_{\Gamma} \frac{\mu_f}{\rho_f} \vec{\delta} \cdot \vec{\nabla} \cdot \vec{\omega} \cdot \vec{n} d\Gamma \quad (3.16)$$

$$\int_{\Omega} \nabla^2 \psi \cdot \vec{\phi} d\Omega = - \int_{\Omega} \vec{\nabla} \cdot \psi \cdot \vec{\nabla} \cdot \vec{\phi} d\Omega + \int_{\Gamma} \vec{\phi} \cdot \vec{\nabla} \cdot \psi \cdot \vec{n} d\Gamma \quad (3.17)$$

onde \vec{n} é um vetor normal unitário, orientado para o exterior do contorno Γ . Como as condições de contorno definidas para o problema em 3.2.1 apontam apenas condições de Dirichlet, isto é, valores fixos no contorno, pode-se assumir como hipótese que $\delta = 0$ e $\phi = 0$ em todo o contorno Γ . Assim, a integral em Γ é nula e os termos difusivos são anulados:

$$-\int_{\Omega} \frac{\mu_f}{\rho_f} \nabla^2 \vec{\omega} \cdot \vec{\delta} d\Omega = \int_{\Omega} \frac{\mu_f}{\rho_f} \vec{\nabla} \cdot \vec{\omega} \cdot \vec{\nabla} \cdot \vec{\delta} d\Omega \quad (3.18)$$

$$\int_{\Omega} \nabla^2 \psi \cdot \vec{\phi} d\Omega = - \int_{\Omega} \vec{\nabla} \cdot \psi \cdot \vec{\nabla} \cdot \vec{\phi} d\Omega \quad (3.19)$$

As equações resultantes são:

$$\int_{\Omega} \frac{\partial \vec{\omega}}{\partial t} \cdot \vec{\delta} d\Omega + \int_{\Omega} \vec{v}_f \cdot \vec{\nabla} \cdot \vec{\omega} \cdot \vec{\delta} d\Omega + \int_{\Omega} \frac{\mu_f}{\rho_f} \vec{\nabla} \cdot \vec{\omega} \cdot \vec{\nabla} \cdot \vec{\delta} d\Omega = 0 \quad (3.20)$$

$$-\int_{\Omega} \vec{\nabla} \cdot \psi \cdot \vec{\nabla} \cdot \vec{\phi} d\Omega + \int_{\Omega} \omega_z \cdot \vec{\phi} d\Omega = 0 \quad (3.21)$$

$$\int_{\Omega} \vec{v}_f \cdot \vec{\xi} d\Omega - \int_{\Omega} \left(\frac{\partial \psi}{\partial y}, -\frac{\partial \psi}{\partial x} \right) \cdot \vec{\xi} d\Omega = 0 \quad (3.22)$$

Se assumir-se que:

$$m_1 \left(\frac{\partial \vec{\omega}}{\partial t}, \delta \right) = \int_{\Omega} \frac{\partial \vec{\omega}}{\partial t} \cdot \vec{\delta} d\Omega \quad (3.23)$$

$$g_1(\vec{v}_f, \vec{\delta}) = \int_{\Omega} \vec{v}_f \cdot \vec{\nabla} \cdot \vec{\omega} \cdot \vec{\delta} d\Omega \quad (3.24)$$

$$k_1(\vec{\omega}, \vec{\delta}) = \int_{\Omega} \vec{\nabla} \cdot \vec{\omega} \cdot \vec{\nabla} \cdot \vec{\delta} d\Omega \quad (3.25)$$

$$k_2(\psi, \vec{\phi}) = \int_{\Omega} \vec{\nabla} \cdot \psi \cdot \vec{\nabla} \cdot \vec{\phi} d\Omega \quad (3.26)$$

$$m_2(\omega_z, \vec{\phi}) = \int_{\Omega} \omega_z \cdot \vec{\phi} d\Omega \quad (3.27)$$

$$m_3(\vec{v}_f, \vec{\xi}) = \int_{\Omega} \vec{v}_f \cdot \vec{\xi} d\Omega \quad (3.28)$$

$$g_3(\psi, \vec{\xi}) = \int_{\Omega} \left(\frac{\partial \psi}{\partial y}, -\frac{\partial \psi}{\partial x} \right) \cdot \vec{\xi} d\Omega \quad (3.29)$$

então as equações na forma fraca são:

$$m_1 \left(\frac{\partial \vec{\omega}}{\partial t}, \delta \right) + g_1(\vec{v}_f, \vec{\delta}) + \frac{\mu_f}{\rho_f} k_1(\vec{\omega}, \vec{\delta}) = 0 \quad (3.30)$$

$$-k_2(\psi, \vec{\phi}) + m_2(\omega_z, \vec{\phi}) = 0 \quad (3.31)$$

$$m_3(\vec{v}_f, \vec{\xi}) - g_3(\psi, \vec{\xi}) = 0 \quad (3.32)$$

Para os seguintes conjuntos de funções bases:

$$\mathbb{W} = \left\{ \omega \in \Omega \rightarrow \mathbb{R}^2 : \int_{\Omega} \omega^2 d\Omega < \infty; \omega \in \omega_{\Gamma} \right\} \quad (3.33)$$

$$\mathbb{P} = \left\{ \psi \in \Omega \rightarrow \mathbb{R}^2 : \int_{\Omega} \psi^2 d\Omega < \infty; \psi \in \psi_{\Gamma} \right\} \quad (3.34)$$

$$\mathbb{V} = \left\{ v_f \in \Omega \rightarrow \mathbb{R}^2 : \int_{\Omega} v_f^2 d\Omega < \infty; v_f \in v_{f\Gamma} \right\} \quad (3.35)$$

3.2.3 Discretização Espacial

A escolha das funções peso pode ser realizada de várias formas, por simplicidade este trabalho utiliza a **Formulação de Galerkin**. Neste método, as funções peso são utilizadas com o mesmo valor da função interpoladora de cada variável. Substituindo-se nas equações:

$$\begin{aligned} \int_{\Omega} \frac{\partial \omega}{\partial t} \delta d\Omega + \int_{\Omega} v_{fx} \frac{\partial \omega}{\partial x} \delta d\Omega + \int_{\Omega} v_{fy} \frac{\partial \omega}{\partial y} \delta d\Omega \\ + \int_{\Omega} \frac{\mu_f}{\rho_f} \left(\frac{\partial \omega}{\partial x} \frac{\partial \delta}{\partial x} + \frac{\partial \omega}{\partial y} \frac{\partial \delta}{\partial y} \right) d\Omega = 0 \end{aligned} \quad (3.36)$$

$$- \int_{\Omega} \left(\frac{\partial \psi}{\partial x} \frac{\partial \phi}{\partial x} + \frac{\partial \psi}{\partial y} \frac{\partial \phi}{\partial y} \right) d\Omega + \int_{\Omega} \omega_z \phi d\Omega = 0 \quad (3.37)$$

$$\int_{\Omega} v_{fx} \xi d\Omega - \int_{\Omega} \frac{\partial \psi}{\partial y} \xi d\Omega = 0 \quad (3.38)$$

$$\int_{\Omega} v_{fy} \xi d\Omega + \int_{\Omega} \frac{\partial \psi}{\partial x} \xi d\Omega = 0 \quad (3.39)$$

As discretizações são aplicadas sobre um domínio com n_e elementos e n_p nós. Este domínio é determinado por uma malha computacional criada. As variáveis

ficam então:

$$\omega(\vec{x}, t) = \sum_{i=1}^{n_p} \omega_i(t) N_i(\vec{x}) \quad (3.40)$$

$$\psi(\vec{x}, t) = \sum_{i=1}^{n_p} \psi_i(t) N_i(\vec{x}) \quad (3.41)$$

$$v_{fx}(\vec{x}, t) = \sum_{i=1}^{n_p} v_{fxi}(t) N_i(\vec{x}) \quad (3.42)$$

$$v_{fy}(\vec{x}, t) = \sum_{i=1}^{n_p} v_{fyi}(t) N_i(\vec{x}) \quad (3.43)$$

onde os valores das funções em cada nó $\omega_i = [\omega_1, \dots, \omega_{n_p}]$, $\psi_i = [\psi_1, \dots, \psi_{n_p}]$, $v_{fxi} = [v_{fx1}, \dots, v_{fxn_p}]$ e $v_{fyi} = [v_{fy1}, \dots, v_{fyn_p}]$ são as incógnitas desejadas. Como estas funções são independentes do tempo, elas são retiradas dos termos de integração sobre o domínio Ω . Enquanto isso, as funções de aproximação $N_i = [N_1, \dots, N_{n_p}]$, também chamadas de funções base, são escolhidas arbitrariamente.

Na formulação de Galerkin, as funções de base são iguais as suas respectivas funções peso:

$$\delta(\vec{x}, t) = \sum_{j=1}^{n_p} \delta_j(t) N_j(\vec{x}) \quad (3.44)$$

$$\phi(\vec{x}, t) = \sum_{j=1}^{n_p} \phi_j(t) N_j(\vec{x}) \quad (3.45)$$

$$\xi(\vec{x}, t) = \sum_{j=1}^{n_p} \xi_j(t) N_j(\vec{x}) \quad (3.46)$$

Então as equações do sistema em suas formas variacionais discretizadas no

espaço ficam como:

$$\begin{aligned} & \int_{\Omega} \sum_{i=1}^{n_p} \frac{\partial \omega_i}{\partial t} N_i \sum_{j=1}^{n_p} \delta_j N_j d\Omega \\ & + \int_{\Omega} v_{fx} \sum_{i=1}^{n_p} \frac{\partial \omega_i N_i}{\partial x} \sum_{j=1}^{n_p} \delta_j N_j d\Omega + \int_{\Omega} v_{fy} \sum_{i=1}^{n_p} \frac{\partial \omega_i N_i}{\partial y} \sum_{j=1}^{n_p} \delta_j N_j d\Omega \\ & + \int_{\Omega} \frac{\mu_f}{\rho_f} \left(\sum_{i=1}^{n_p} \frac{\partial \omega_i N_i}{\partial x} \sum_{j=1}^{n_p} \frac{\partial \delta_j N_j}{\partial x} + \sum_{i=1}^{n_p} \frac{\partial \omega_i N_i}{\partial y} \sum_{j=1}^{n_p} \frac{\partial \delta_j N_j}{\partial y} \right) d\Omega = 0 \end{aligned} \quad (3.47)$$

$$\begin{aligned} & - \int_{\Omega} \left(\sum_{i=1}^{n_p} \frac{\partial \psi_i N_i}{\partial x} \sum_{j=1}^{n_p} \frac{\partial \phi_j N_j}{\partial x} + \sum_{i=1}^{n_p} \frac{\partial \psi_i N_i}{\partial y} \sum_{j=1}^{n_p} \frac{\partial \phi_j N_j}{\partial y} \right) d\Omega \\ & + \int_{\Omega} \sum_{i=1}^{n_p} \omega_{zi} N_i \sum_{j=1}^{n_p} \phi_j N_j d\Omega = 0 \end{aligned} \quad (3.48)$$

$$\int_{\Omega} \sum_{i=1}^{n_p} v_{fxi} N_i \sum_{j=1}^{n_p} \xi_j N_j d\Omega - \int_{\Omega} \sum_{i=1}^{n_p} \frac{\partial \psi_i N_i}{\partial y} \sum_{j=1}^{n_p} \xi_j N_j d\Omega = 0 \quad (3.49)$$

$$\int_{\Omega} \sum_{i=1}^{n_p} v_{fyi} N_i \sum_{j=1}^{n_p} \xi_j N_j d\Omega + \int_{\Omega} \sum_{i=1}^{n_p} \frac{\partial \psi_i N_i}{\partial x} \sum_{j=1}^{n_p} \xi_j N_j d\Omega = 0 \quad (3.50)$$

Pode-se remover as componentes da velocidade do fluido v_{fx} e v_{fy} da equação Eq. 3.47, pois serão utilizados os componentes da velocidade no último passo de tempo para estes valores, tornando-se a equação linear.

Retiram-se então os somatórios das funções interpoladoras das integrais, e

como $\sum_{j=1}^{n_p} \delta_j \neq 0$, $\sum_{j=1}^{n_p} \phi_j \neq 0$ e $\sum_{j=1}^{n_p} \xi_j \neq 0$ as equações de governo serão:

$$\begin{aligned} & \sum_{j=1}^{n_p} \sum_{i=1}^{n_p} \frac{\partial \omega_i}{\partial t} \int_{\Omega} N_i N_j d\Omega \\ & + \sum_{j=1}^{n_p} \sum_{i=1}^{n_p} \omega_i \left(v_{fx} \int_{\Omega} \frac{\partial N_i}{\partial x} N_j d\Omega + v_{fy} \int_{\Omega} \frac{\partial N_i}{\partial y} N_j d\Omega \right. \\ & \left. + \frac{\mu_f}{\rho_f} \int_{\Omega} \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} d\Omega \right) = 0 \end{aligned} \quad (3.51)$$

$$\sum_{j=1}^{n_p} \sum_{i=1}^{n_p} \psi_i \left(- \int_{\Omega} \left(\frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} \right) d\Omega + \omega_i \int_{\Omega} N_i N_j d\Omega + \right) = 0 \quad (3.52)$$

$$\sum_{j=1}^{n_p} \sum_{i=1}^{n_p} v_{fxi} \left(\int_{\Omega} N_i N_j d\Omega - \psi_i \int_{\Omega} \frac{\partial N_i}{\partial y} N_j d\Omega + \right) = 0 \quad (3.53)$$

$$\sum_{j=1}^{n_p} \sum_{i=1}^{n_p} v_{fyi} \left(\int_{\Omega} N_i N_j d\Omega + \psi_i \int_{\Omega} \frac{\partial N_i}{\partial x} N_j d\Omega + \right) = 0 \quad (3.54)$$

3.2.4 Malha Computacional

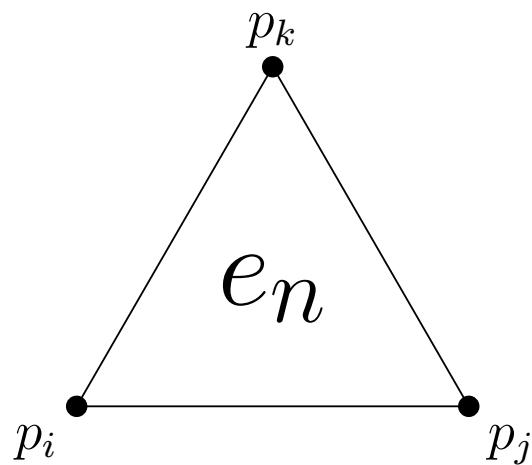
A malha utilizada pode ser estruturada, com nós equidistantes, ou não estruturada, com nós posicionados a critério do criador da malha. É comum utilizar malhas não estruturadas com zonas de maior interesse com elementos menores, para se obter informações mais precisas nestes locais. Em certos casos de equações com acoplamentos fortemente ligados, se torna necessário utilizar elementos com mais informações atreladas, como elementos quadráticos ou cúbicos. Este é o caso da solução da equação de Navier-Stokes. Porém, para evitar esta restrição, este trabalho utilizou a formulação de corrente-vorticidade, que desvia deste problema. Podendo então utilizar elementos lineares e simplificando sua implementação.

Como apresentado por Anjos (2007) [21], os tipos de elementos mais comuns

aplicados a aos diferentes tipos de simulações são:

- Dimensão do problema
 - Caso unidimensional: Retas
 - Caso bidimensional: Triângulos ou Retângulos
 - Caso tridimensional: Tetraedros ou Prismas
- Ordem dos polinômios interpoladores
 - Primeiro grau: Lineares
 - Segundo grau: Quadráticos
 - Terceiro grau: Cúbicos

Para este trabalho foi escolhido o elemento do tipo triangular com interpolação linear, pois não há restrições no problema escolhido, e este é o caso mais aplicado na literatura. A Figura 3 mostra como são os elementos utilizados:



Fonte: Autor.

Figura 3: Elemento triangular linear.

Pode-se notar na Figura 3 que o elemento de índice e_n é delimitado pelos nós p_i , p_j e p_k . A função de interpolação aplicada entre os nós é uma função linear entre suas posições.

Então pode-se escrever as equações do problema em sua forma matricial como:

$$\mathbf{M} \frac{\partial \omega}{\partial t} + v_{fx} \mathbf{G}_x \omega + v_{fy} \mathbf{G}_y \omega + \frac{\mu_f}{\rho_f} (\mathbf{K}_{xx} + \mathbf{K}_{yy}) \omega = 0 \quad (3.55)$$

$$- (\mathbf{K}_{xx} + \mathbf{K}_{yy}) \psi + \mathbf{M} \omega = 0 \quad (3.56)$$

$$\mathbf{M} v_{fx} \omega - \mathbf{G}_y \psi = 0 \quad (3.57)$$

$$\mathbf{M} v_{fy} \omega + \mathbf{G}_x \psi = 0 \quad (3.58)$$

onde \mathbf{M} , \mathbf{G}_x , \mathbf{G}_y , \mathbf{K}_{xx} e \mathbf{K}_{yy} são as matrizes elementares de dimensão $n_p \times n_p$.

Estas matrizes são montadas a partir da contribuição local de cada nó em um elemento e unidas na matriz geral, onde cada posição da matriz é o valor correspondente ao nó de mesmo índice. As matrizes locais \mathbf{m}^e , \mathbf{g}_x^e , \mathbf{g}_y^e , \mathbf{k}_{xx}^e e \mathbf{k}_{yy}^e , para cada elemento e são definidas como:

$$\mathbf{m}^e = \int_{\Omega^e} N_i^e N_j^e d\Omega^e \quad (3.59)$$

$$\mathbf{g}_x^e = \int_{\Omega^e} \frac{\partial N_i^e}{\partial x} N_j^e d\Omega^e \quad (3.60)$$

$$\mathbf{g}_y^e = \int_{\Omega^e} \frac{\partial N_i^e}{\partial y} N_j^e d\Omega^e \quad (3.61)$$

$$\mathbf{k}_{xx}^e = \int_{\Omega^e} \frac{\partial N_i^e}{\partial x} \frac{\partial N_j^e}{\partial x} d\Omega^e \quad (3.62)$$

$$\mathbf{k}_{yy}^e = \int_{\Omega^e} \frac{\partial N_i^e}{\partial y} \frac{\partial N_j^e}{\partial y} d\Omega^e \quad (3.63)$$

As matrizes utilizadas foram obtidas por Lewis, Nithiarasu e Seetharamu

(2004) [4], e são escritas em suas formas matriciais:

$$\mathbf{m}^e = \frac{A^e}{12} \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix} \quad (3.64)$$

$$\mathbf{g}_x^e = \frac{1}{6} \begin{bmatrix} b_i & b_j & b_k \\ b_i & b_j & b_k \\ b_i & b_j & b_k \end{bmatrix} \quad (3.65)$$

$$\mathbf{g}_y^e = \frac{1}{6} \begin{bmatrix} c_i & c_j & c_k \\ c_i & c_j & c_k \\ c_i & c_j & c_k \end{bmatrix} \quad (3.66)$$

$$\mathbf{k}_{xx}^e = \frac{t_h}{4A} \begin{bmatrix} b_i b_i & b_j b_i & b_k b_i \\ b_i b_j & b_j b_j & b_k b_j \\ b_i b_k & b_j b_k & b_k b_k \end{bmatrix} \quad (3.67)$$

$$\mathbf{k}_{yy}^e = \frac{t_h}{4A} \begin{bmatrix} c_i c_i & c_j c_i & c_k c_i \\ c_i c_j & c_j c_j & c_k c_j \\ c_i c_k & c_j c_k & c_k c_k \end{bmatrix} \quad (3.68)$$

onde A^e é a área, t_h a espessura e $b_i, b_j, b_k, c_i, c_j, c_k$ as coordenadas relativas do elemento. Estas coordenadas relativas são definidas para os nós p_i, p_j e p_k de um elemento qualquer como:

$$b_i = y_j - y_k \quad (3.69)$$

$$b_j = y_k - y_i \quad (3.70)$$

$$b_k = y_i - y_j \quad (3.71)$$

$$c_i = x_k - x_j \quad (3.72)$$

$$c_j = x_i - x_k \quad (3.73)$$

$$c_k = x_j - x_i \quad (3.74)$$

onde x_i e y_i são as coordenadas do nó p_i , x_j e y_j são as coordenadas do nó p_j e x_k e y_k são as coordenadas do nó p_k .

E a área A pode ser calculada através das coordenadas dos nós pela equação:

$$A^e = \frac{1}{2}((x_i y_j - x_j y_i) + (x_j y_k - x_k y_j) + (x_k y_i - x_i y_k)) \quad (3.75)$$

3.3 Método de Diferenças Finitas

O Método de Diferenças Finitas é um das mais antigas e simples formas de se calcular o valor de um diferencial numericamente. A base do método é tomar diferenças pequenas o suficiente para replicar o valor tomado no meio contínuo.

Imagina-se uma função $f(x)$ tomada no domínio $x \in \mathbb{R}$ e deseja-se obter a derivada $\frac{df}{dx}(x)$. Tomando-se definição da derivada, pode-se escrever:

$$\frac{df}{dx}(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h} \quad (3.76)$$

onde h é a diferença entre os nós tomados na variável x . Ou seja, caso deseja-se obter o valor da derivada desta função numericamente basta aplicar a diferença entre dois nós com afastamento suficientemente pequeno.

Outra forma de interpretar-se o MDF seria através da aplicação da *Série de Taylor*. Tomando-se novamente a função $f(x)$ e a série de Taylor aplicada a ela

ao redor de um nó na posição x qualquer, tem-se:

$$f(x+h) \approx f(x) + \frac{df(x)}{dx}(x+h-x) + \frac{1}{2!} \frac{d^2 f(x)}{dx^2}(x+h-x)^2 + \frac{1}{3!} \frac{d^3 f(x)}{dx^3}(x+h-x)^3 + \dots \quad (3.77)$$

Como deseja-se obter o valor da derivada de primeira ordem, pode-se reorganizar a equação para extrair o termo $\frac{df(x)}{dx}$:

$$\frac{df(x)}{dx} \approx \frac{f(x+h) - f(x) - \frac{1}{2!} \frac{d^2 f(x)}{dx^2}(h)^2 - \frac{1}{3!} \frac{d^3 f(x)}{dx^3}(h)^3 - \dots}{h} \quad (3.78)$$

Em certos casos não se tem nenhuma informação sobre as derivadas de ordens superiores da função, portanto, pode-se aproximar o valor da derivada ao remover os termos com ordens superiores e substituí-los por um termo de erro $O(h^2)$.

$$\frac{df(x)}{dx} \approx \frac{f(x+h) - f(x)}{h} + O(h^2) \quad (3.79)$$

onde, novamente, quanto menor for o passo h , mais o valor da derivada numérica se aproximará do valor contínuo.

3.3.1 Discretização da Corrente-Vorticidade no Tempo

A equação de governo Eq. 3.55 do problema de corrente-vorticidade ainda possui um termo derivativo não discretizado. Este é o termo $\frac{\partial \omega}{\partial t}$, e ele é formado utilizando a discretização em MDF apresentada em Eq. 3.79.

$$\frac{\partial \omega}{\partial t} \approx \frac{\omega(t+dt) - \omega(t)}{dt} \quad (3.80)$$

onde dt é o passo de tempo da simulação.

Será utilizada a notação de índice de variáveis no tempo como sobrescrito, onde t_n é o índice do passo no tempo, $\omega^{t_{n+1}} = \omega(t+dt)$ e $\omega^{t_n} = \omega(t)$. Então a

discretização final do sistema corrente-vorticidade é:

$$\left(\mathbf{M}v_{fx}\mathbf{G_x} + v_{fy}\mathbf{G_y} + \frac{\mu_f}{\rho_f} (\mathbf{K_{xx}} + \mathbf{K_{yy}}) \right) \omega^{t_{n+1}} = \mathbf{M}\omega^{t_n} \quad (3.81)$$

$$(\mathbf{K_{xx}} + \mathbf{K_{yy}}) \psi = \mathbf{M}\omega^{t_{n+1}} \quad (3.82)$$

$$\mathbf{M}v_{fx}\omega^{t_{n+1}} = \mathbf{G_y}\psi \quad (3.83)$$

$$\mathbf{M}v_{fy}\omega^{t_{n+1}} = -\mathbf{G_x}\psi \quad (3.84)$$

3.3.2 Discretização da Forças Aplicadas às Partículas

As equações das forças aplicadas descritas em 2.6 são discretizadas utilizando MDF com seus parâmetros em cada instante t_n . Suas equações discretizadas são:

- **Força Gravitacional:**

$$\vec{F}_{grav}^{t_n} = m_p \vec{g} \quad (3.85)$$

- **Força de Arrasto:**

$$\vec{F}_{drag}^{t_n} = 3\pi\mu_f d_p \left(\vec{v}_f^{t_n} - \vec{v}_p^{t_{n-1}} \right) \quad (3.86)$$

- **Força de Sustentação:**

$$\vec{F}_{lift}^{t_n} = 1.61\mu_f d_p \left(\vec{v}_f^{t_n} - \vec{v}_p^{t_{n-1}} \right) \sqrt{Re_G^{t_n}} \quad (3.87)$$

$$Re_G^{t_n} = \frac{d_p^2 \rho_f}{\mu_f} \left(\frac{d\vec{v}_f}{d\vec{r}} \right)^{t_n} \quad (3.88)$$

onde $\frac{d\vec{v}_f}{d\vec{r}}$ é a variação do valor da velocidade nas extremidades da partícula, em um eixo perpendicular ao movimento do fluido que cruza o centro de

massa da mesma.

- **Força de Massa Virtual:**

$$\vec{F}_{mass}^{t_n} = \frac{1}{2} \rho_f V_p \frac{(\vec{v}_f^{t_n} - \vec{v}_p^{t_{n-1}}) - (\vec{v}_f^{t_{n-1}} - \vec{v}_p^{t_{n-2}})}{dt} \quad (3.89)$$

Finalmente, após calcular-se as forças aplicadas, é necessário determinar a nova velocidade e posição de cada partícula:

$$\vec{v}_p^{t_n} = \frac{dt}{m_p} \left(\vec{F}_{grav}^{t_n} + \vec{F}_{drag}^{t_n} + \vec{F}_{lift}^{t_n} + \vec{F}_{mass}^{t_n} \right) \quad (3.90)$$

$$(p_x, p_y)^{t_n} = \vec{v}_p^{t_n} dt \quad (3.91)$$

4 CÓDIGO NUMÉRICO

4.1 Introdução

O código deste trabalho foi feito completamente pelo autor na linguagem ***Python 3.5***, com auxílio das bibliotecas: *NumPy*, para manipulação de dados e matrizes, *SciPy* para solução de sistemas e matrizes esparsas, *Matplotlib* para a visualização de gráficos e *pygmsh* para importação da malha no formato `.msh`.

Para a geração de malhas foi utilizado o *software open source Gmsh*, como demonstrado por Geuzaine e Remacle (2009) [22]. Nele é possível desenhar o perfil do domínio e gerar os nós e elementos de acordo com o refino desejado. Também pode-se forçar diferentes tamanhos de elementos em regiões de maior interesse.

Para a visualização de resultados foi utilizado o *software open source Paraview*, pois este permite uma verificação visual da evolução dos resultados, assim como diversas ferramentas para avaliação dos dados.

Todos os softwares de terceiros foram usados com permissão e licenças apropriadas.

4.2 Estrutura do Código

O código foi planejado e desenvolvido com o paradigma de *Orientação em Objeto* em mente. Isto é, o código é centralizado em classes que possuem propriedades e métodos referentes a modelagem física do problema estudado. Por exemplo, as classes `TccLib.Mesh` e `TccLib.Particle` representam os conceitos físicos de malha e partícula respectivamente. Estas classes criam objetos que contém suas propriedades individuais, como as densidades ρ_f e ρ_p , e permitem um melhor entendimento do fenômeno.

Este código foi pensado em ser utilizado como uma biblioteca pública com

acesso liberado a qualquer um interessado (link do projeto no GitHub). Por isso, as funções e métodos possuem diversas opções de chamadas. Mais informações estão disponíveis através do comando interno:

```
help(arg)
```

substituindo-se o argumento arg pela função desejada.

Demonstra-se a seguir um exemplo de uso da biblioteca:

```
# Importação da biblioteca
import TccLib

# Importação da malha ou coordenadas de uma nova
malha = TccLib.Mesh("arquivo_da_malha.msh")
# ou malha = TccLib.Mesh([coordenadas (x, y)])

# Definição das condições de contorno
malha.new_boundary_condition("nome da propriedade",
                             [índices dos nós],
                             [valor da condição no nó],
                             [1 para Dirichlet ou 0 para Neumann])

# Chamada para a função de solução
v_x, v_y = TccLib.solve_velocity_field(malha)

# Adição de partículas
malha.add_particle(propriedades da partícula)
# Loop de movimentação das partículas
```

```
for t in time_list:
    TccLib.move_particles(malha, (v_x, v_y))
```

Agora serão explorados cada etapa separadamente.

4.2.1 Importação da Malha

O objeto de malha `TccLib.Mesh` possui um construtor que aceita como parâmetros uma lista de coordenadas dos nós da malha ou o nome de um arquivo na extensão `.msh` para importação. Após a obtenção dos nós é utilizada a função `scipy.spatial.Delaunay()`, que retorna uma lista com os conjuntos de índices dos nós por elemento.

São cadastradas as informações da malha criada no objeto, como as coordenadas dos nós nos eixos x e y , o número de elementos e o número de nós da malha. É criado um dicionário vazio para receber as condições de contorno, onde suas chaves são nomes fixos para cada problema.

4.2.2 Definição das Condições de Contorno

A classe de condições de contorno `TccLib.BoundaryConditions` é uma classe privada, ou seja, o usuário não interage com ela diretamente. Para cadastrar novas condições de contorno é preciso utilizar o método dentro da classe `TccLib.Mesh.new_boundary_condition()`.

O nome da propriedade varia de acordo com a função de solução. Para o propósito deste trabalho, a solução do sistema corrente-vorticidade, são registradas as condições de contorno com nomes: "`psi`" para os valores da corrente, "`vel_x`" para os valores da velocidade do fluido no eixo x e "`vel_y`" para os valores da velocidade do fluido no eixo y.

Os valores da condição de contorno são passados em três listas de mesmo

tamanho, onde a primeira contém os índices dos elementos, a segunda contém os valores da condição correspondente ao índice associado e a terceira contém um valor lógico para a condição correspondentes ao índice, onde verdadeiro representa uma condição do tipo Dirichlet e falso representa o tipo Neumann. Caso seja fornecido um valor numérico, ou *booleano*, ao invés da segunda e terceira lista respectivamente, será interpretado que todos os valores da lista substituída possuem este valor.

4.2.3 Funções de Solução

Esta biblioteca foi projetada para adaptar diversas funções de solução para cada modelo físico desejado. No momento da escrita, há duas funções de soluções implementadas no código:

- Equação de Poisson (MEF) - `TccLib.solve_poisson()`
- Sistema Corrente-Vorticidade (MEF) - `TccLib.solve_velocity_field()`

A solução dos problemas é realizada de forma semelhante. Primeiramente são geradas as malhas globais do domínio da malha em uma função auxiliar `TccLib.get_matrices()`:

```
# Loop em cada elemento na lista da malha
for elem in malha.ien:
    x = malha.x[elem] # = [x_i, x_j, x_k]
    y = malha.y[elem] # = [y_i, y_j, y_k]

    # Criação das matrizes locais
    ...
    ...
```

```

# Registro das matrizes locais nas matrizes globais
for i in range(3):
    for j in range(3):
        kx_global[elem[i], elem[j]] += k_x[i][j]
        ky_global[elem[i], elem[j]] += k_y[i][j]
        m_global[elem[i], elem[j]] += m[i][j]
        gx_global[elem[i], elem[j]] += g_x[i][j]
        gy_global[elem[i], elem[j]] += g_y[i][j]

```

que retorna as matrizes globais na ordem apresentada. As matrizes globais são construídas dentro de um loop que percorre os elementos e monta as matrizes locais. Elas são então adicionadas a uma matriz esparsa global, que é um objeto que contém somente os elementos não nulos da matriz, referenciados por seus índices. Desta forma, otimiza-se o armazenamento de memória durante a simulação, pois permite representar uma matriz com tamanho $N \times N$ com apenas elementos existentes na memória, com a ordem computacional de N .

Em seguida, são montadas as matrizes principais para a solução algébrica:

$$A_{n \times n} \cdot x_n = b_n \quad (4.1)$$

onde n é o número de nós da malha.

Então são aplicadas as condições de contorno, utilizando a função privada `TccLib.apply_boundary_conditions()`:

```

# Iteração sobre a lista de nós com condição de contorno
for coluna in no_list:
    if condição é Dirichlet:
        # Dirichlet
        for linha in lista_de_linhas:

```

```

# Compensar linhas anuladas
b[linha] -= A[linha, coluna] * condicao[coluna]
# Zerar termos na matriz principal
A[linha, coluna] = 0.
A[coluna, linha] = 0.

# Igualar valor da condição no nó a ela mesma
A[coluna, coluna] = 1.

b[coluna, 0] = condicao[coluna]

else:
    # Neumann Treatment
    b[coluna, 0] += condicao[coluna]

```

Finalmente, o sistema algébrico é solucionado pela função terceirizada da biblioteca de álgebra linear, otimizada, `scipy.sparse.linalg.spsolve()`.

As funções de solução retornam listas com o valor da solução calculada para cada nó na ordem de índices registrada no objeto de malha. Há métodos criados para auxiliar a visualização dos resultados no objeto de malha, como o método `TccLib.Mesh.show_velocity_quiver()` que apresenta um gráfico com flechas apontando a direção do fluxo em cada nó e com tamanho proporcional a seu módulo.

4.2.4 Movimentação das Partículas

No caso do problema multifásico de partículas inseridas em um fluido em movimento, há uma etapa de movimentação das partículas separada da função de solução principal. Isto é devido ao fato do método de solução escolhido utilizar a implementação *one-way* como apresentado por Crowe et al. (2011) [20]. Neste tipo de solução o movimento do fluido, meio principal, afeta as partículas, meio

secundário, porém elas não causam reflexos no fluido. Esta implementação foi escolhida devida a demanda computacional elevada da implementação *two-way*, que requereria solucionar o campo de velocidades (MEF) mais vezes por iteração. Então as partículas são simuladas percorrendo um escoamento permanente, com o valor do campo de velocidades sido calculado anteriormente.

As partículas são adicionadas a malha através de um método próprio, que leva como parâmetros as propriedades físicas da partícula, sua posição e condições iniciais e um nome para registro, `TccLib.Mesh.add_particle()`.

Em cada chamada da função `TccLib.move_particles()` são calculadas as forças aplicadas a cada partícula ainda presente no domínio da malha e as forças são adicionadas a um dicionário com seus respectivos nomes como chaves.

```
# Iteração sobre a lista partículas,
# checando se cada uma está presente no domínio
for particula in particulas if particula in malha:
    # Inicialização do dicionário
    forcas = dict()

    # Adição das forças
    forcas["gravitational"] = ...
    forcas["arrasto"] = ...
    forcas["sustentacao"] = ...
    forcas["massa_virtual"] = ...

    # Função que altera as propriedades da partícula
    particula.apply_forces(...)
```

Para se calcular estas forças são necessárias diversas informações que são captadas nesta função, como a velocidade do escoamento na posição da partícula.

E o valor do número de Reynolds (Eq. 2.48) é calculado em cada chamada, para verificar se a simulação respeita a restrição apresentada em 2.6. Em seguida é chamada uma função interna da classe `TccLib.Particle` para executar o reposicionamento das partículas, esta calcula as novas posições das partículas e as move, caso respeitem as condições de colisão.

Portanto, a função de movimentação das partículas é aplicada depois da solução do sistema. A função de movimentação `TccLib.move_particles()` não possui retorno, porém pode-se analisar seu resultado observando-se a propriedade `TccLib.Particle.position_history` das partículas presentes na malha. Esta propriedade é uma lista que contém as posições percorridas pela partícula, o que permite a visualização gráfica do percurso da mesma.

5 VALIDAÇÃO DO CÓDIGO NÚMERO

5.1 Introdução

Neste capítulo serão apresentados as comparações realizadas entre os resultados numéricos obtidos para x casos amplamente conhecidos na literatura e a solução analítica unidimensional dos mesmos. Como a solução numérica é obtida para problemas bidimensionais, é preciso pegar uma seção transversal do domínio e interpolar resultado, produzindo, assim, uma aproximação. Dessa forma, a quantificação do erro relativo médio se faz necessária, com o objetivo de apresentar a acurácia do código numérico. O erro relativo entre a solução numérica e a solução analítica é calculado pela equação (Eq. 5.1):

$$er_i = \frac{|(val_a)_i - (val_n)_i|}{(val_a)_i} \quad (5.1)$$

onde $(val_a)_i$ é o valor encontrado pela solução analítica e $(val_n)_i$ é o valor encontrado pela solução numérica, ambos encontrados no nó i .

São calculados também a média e o desvio padrão dos erros relativos pelas equações (Eq. 5.2) e (Eq. 5.2), respectivamente:

$$er_{mean} = \frac{1}{N} \sum_{i=0}^N er_i \quad (5.2)$$

$$er_{std} = \left\| \frac{1}{N} \sum_{i=0}^N er_i - er_{mean} \right\|_2 \quad (5.3)$$

As validações foram organizadas em três seções que representam diferentes etapas da implementação do modelo matemático, sendo assim espera-se mostrar a aplicabilidade do código numérico desenvolvido, além do histórico de aprendizado obtido. Na seção 5.2, os problemas de transferência de calor em sólidos com

condutividade térmica constante são apresentados. Os casos propostos nesta seção buscam confirmar a importação correta da malha e a montagem das matrizes globais, como também a aplicação das condições de contorno de Dirichlet e de Neumann.

Já na seção 5.3, os problemas com o termo convectivo presente são analisados. Foi considerado o fluido como incompressível e newtoniano, dessa forma a equação de Navier-Stokes pode ser aplicada segundo a formulação corrente-vorticidade. A estrutura do algoritmo de solução nos casos propostos dessa seção é a mesma utilizada na resolução do problema proposto neste trabalho. Dessa forma, podemos confirmar a correta aplicação das condições de contorno da vorticidade que deve ser calculada em cada passo de tempo.

Finalmente, na seção 5.4, é apresentado os clássicos casos de dinâmica em partículas com o intuito de validar as forças de gravidade, arrasto e sustentação isoladamente e, com isso, possibilitar uma correção pontual no modelo quando necessário, além de permitir observar com maior precisão a influência de cada uma das forças sobre as partículas.

Para casos com variáveis temporais, foi utilizado um critério de parada de 10^{-5} de variação de valores entre dois intervalos de tempo consecutivos. Desta maneira espera-se que o sistema já tenha entrado na situação de convergência e esteja próximo o suficiente de seus valores finais. Isto foi feito para poupar tempo de computação, para casos que possuem um limite de tempo elevado e convergem rapidamente, fazendo com que o código continuasse desnecessariamente.

A execução do código e a computação dos resultados foram realizados em um computador de uso pessoal com as seguintes especificações:

- Dell Latitude E6410 com processador Intel® Core™ i5 CPU M 520 2.40GHz com 4 núcleos e 4Gb de memória RAM. O sistema operacional Ubuntu 16.04 LTS e compilador Python 3.5.

5.2 Validações de Problemas em Sólidos

5.2.1 Equação de Laplace com Condições de Contorno de Dirichlet

O problema de troca de calor em uma placa é um dos exemplos clássicos utilizados para estudar as equações de transmissão de calor em sólidos. O mais simples destes é uma barra unidimensional com condutividade térmica constante e sem geração de calor onde a temperatura é conhecida nas extremidades. Como a malha do código foi desenvolvida para solução de problemas bidimensionais, crie-se um problema bidimensional com condições de contorno equivalentes e extraí-se uma seção para que se possa comparar os resultados.

Para a solução dos problemas em sólidos e fluidos no espaço foi utilizado o Método de Elementos Finitos, apresentado na seção 3.2. Enquanto para a aproximação da dinâmica das partículas e solução dos sólidos e fluidos no tempo utilizou-se o Método de Diferenças Finitas, explicado na seção 3.3.

A equação de governo deste fenômeno é conhecida como a equação de Laplace (Eq. 5.4) para sólidos em estado permanente e é apresentada a seguir:

$$\nabla^2 T = 0 \quad (5.4)$$

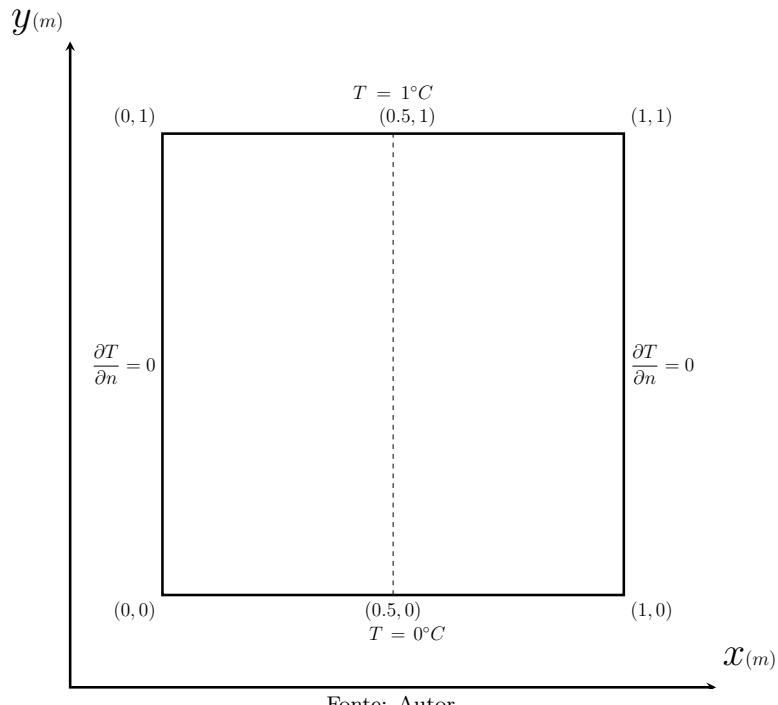
onde T é a temperatura na placa e ∇^2 é o operador diferencial conhecido como laplaciano.

E a solução analítica do problema unidimensional associado foi obtida pelo método dos coeficientes a determinar e é igual a:

$$T(x) = \frac{T_L - T_0}{L}x + T_0 \quad (5.5)$$

onde L é o comprimento da barra, T_0 e T_L são, respectivamente, os valores da temperatura em $x = 0$ e $x = L$.

As condições de contorno e o domínio bidimensional utilizados na simulação são apresentados na Figura 4. A condição de fluxo de calor $\frac{\partial T}{\partial n}$ nulo significa que nenhuma condição é imposta no contorno.



Fonte: Autor.

Figura 4: Condições de contorno da placa para o problema de Laplace 5.2.1.

O domínio foi discretizado utilizando uma malha triangular linear não estruturada com 3162 elementos e 568 nós. A malha foi criada pelo o software GMSH como proposto por Geuzaine e Remacle (2009) [22] e importada ao código numérico. A Figura 5 apresenta o campo de temperatura, onde os eixos x e y representam o domínio e o eixo z é a distribuição de temperatura, e que é possível observar que o campo de temperatura possui um perfil linear variando de 0 (cor azul) a 1 (cor vermelha).

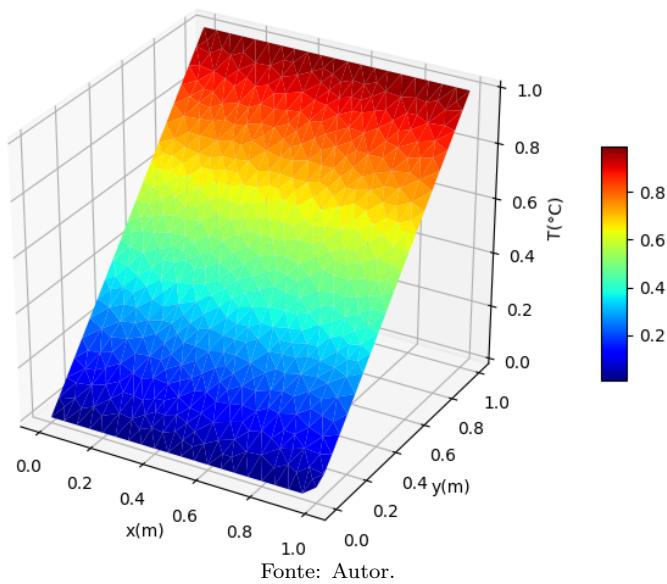
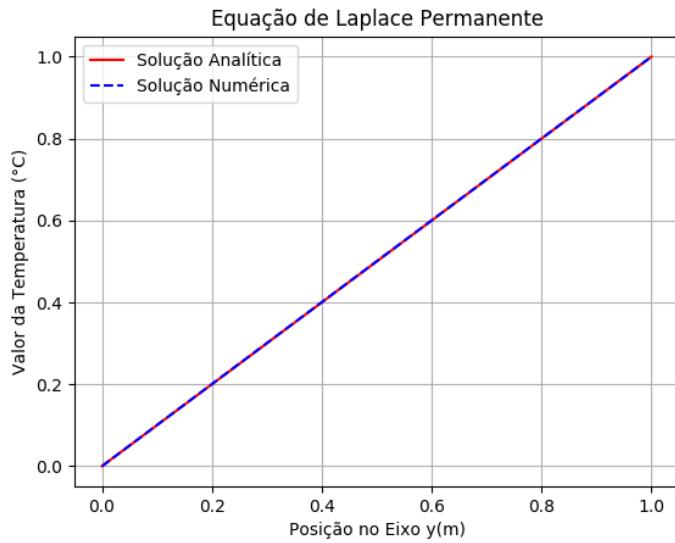


Figura 5: Distribuição de temperaturas na placa da solução permanente da equação de Laplace 5.2.1.

A comparação entre os resultados da solução analítica (Eq. 5.5) e a solução numérica, para a seção $x = 0.5m$, é apresentada na Figura 6, onde é possível observar que ambas possuem o mesmo perfil. Essa proximidade é quantificada pelo erro relativo médio que foi de 0.08929% e com desvio padrão de 0.06698%



Fonte: Autor.

Figura 6: Comparação de resultado das soluções numérica e analítica do caso de transporte de temperatura em sólidos no regime permanente.

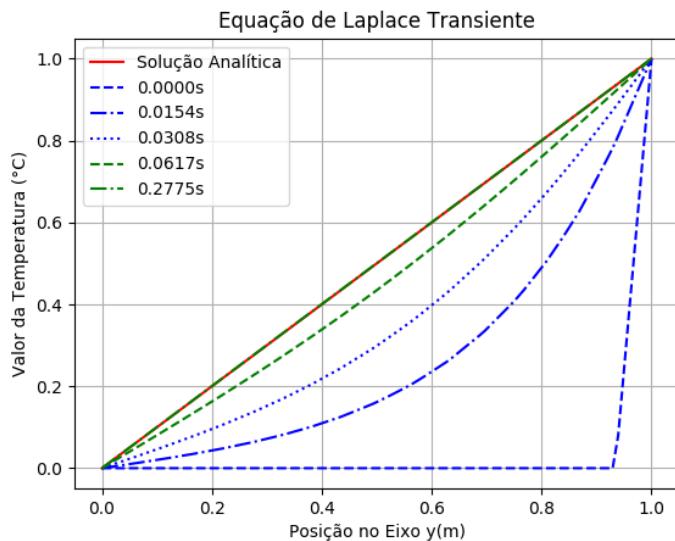
Ao solucionar o mesmo problema introduzindo o termo transiente na equação de governo 5.4, pode-se verificar a evolução de comportamento da temperatura ao longo do tempo. Dessa forma, a equação que representa este caso é:

$$\frac{\partial T}{\partial t} + k\nabla^2 T = 0 \quad (5.6)$$

onde k é o coeficiente de condutividade térmica da placa e t é a variável temporal.

Porém, ao longo dos passos de tempo, a solução se aproxima de um problema permanente, portanto, pode-se fazer a comparação dos resultados obtidos neste exemplo com os valores da solução analítica (Eq. 5.5), tomando-se que $t \rightarrow \infty$. As condições iniciais $t = 0s$ atribuídas aos nós sem condição de contorno foram de um valor inicial de $0^\circ C$. A Figura 7 apresenta a evolução do campo de temperaturas em função do tempo. É possível observar que a solução numérica converge para a solução analítica formando um perfil linear. O erro relativo médio

calculado para este caso foi de 0.08276% e com desvio padrão de 0.06340%.



Fonte: Autor.

Figura 7: Comparação de resultado das soluções numéricas e analítica de transporte em sólidos no regime transiente.

5.2.2 Equação de Poisson com Condições de Contorno de Dirichlet

Neste problema busca-se estudar o comportamento de uma placa com geração de calor em seu domínio e temperaturas fixas nas laterais. Novamente, para permitir a comparação de resultados, é extraída uma seção da placa para observar os resultados como um problema unidimensional.

A equação que governa este caso é denominada equação de Poisson (Eq. 5.7), tomada para um problema permanente, ou seja, sem variação no tempo.

$$-k\nabla^2T = Q \quad (5.7)$$

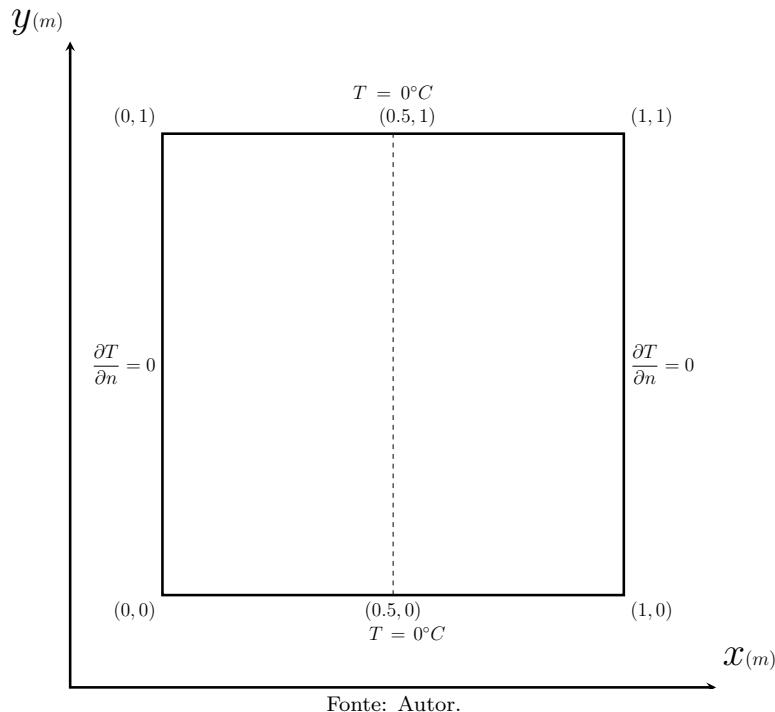
onde Q é a geração de calor na placa.

A solução analítica para o caso de uma barra unidimensional é apresentada

embaixo, obtida novamente pelo método de coeficientes a determinar:

$$T(x) = \frac{Q}{2k} (-x^2 + Lx) + \frac{T_L - T_0}{L} x + T_0 \quad (5.8)$$

As condições de contorno e o domínio bidimensional utilizados na simulação são apresentados na Figura 8. A geração de calor utilizada foi de $Q = 40W/m^3$ e a condutividade térmica foi de $k = 5W/m^{\circ}C$.



Fonte: Autor.

Figura 8: Condições de contorno da placa para o problema de Poisson 5.2.2.

Novamente, foi utilizada uma malha triangular linear não estruturada com 3162 elementos e 568 nós. A Figura 9 apresenta o campo de temperatura, onde os eixos x e y representam o domínio e o eixo z é a distribuição de temperatura, onde é possível observar que o campo de temperatura possui um perfil parabólico variando de 0 (cor azul) a 1 (cor vermelha).

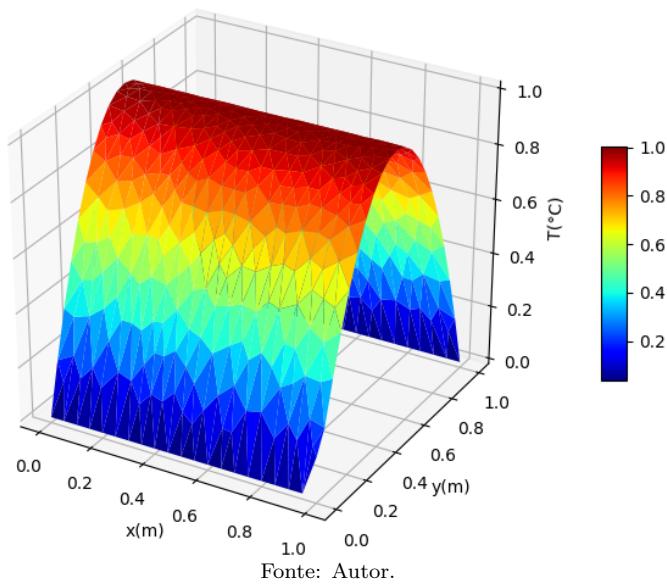
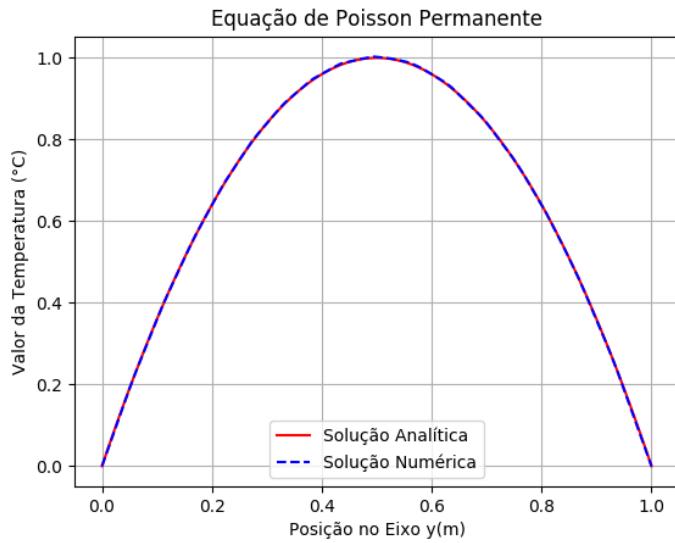


Figura 9: Distribuição de temperaturas na placa da solução do problema permanente de Poisson 5.2.2.

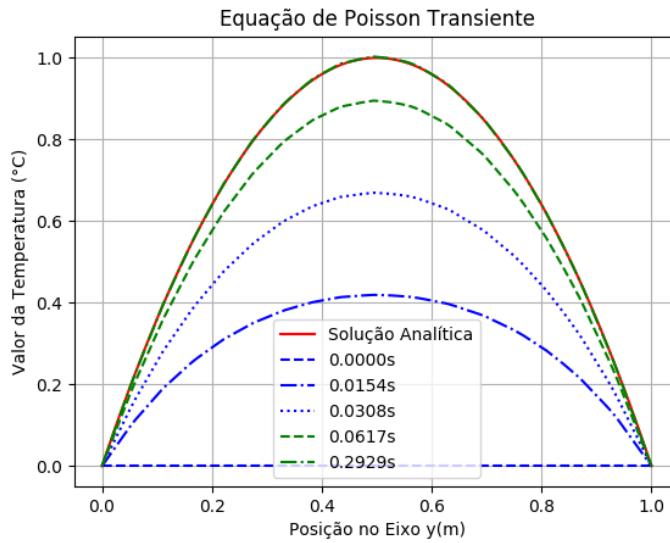
A comparação entre os resultados da solução analítica (Eq. 5.8) e a solução numérica, para a seção $x = 0.5m$, é apresentada na Figura 10, onde é possível observar que ambas possuem o mesmo perfil. Essa proximidade é quantificada pelo erro relativo médio que foi de 0.21048% e com desvio padrão de 0.40921%.



Fonte: Autor.

Figura 10: Comparação de resultado das soluções numérica e analítica do problema de transporte de temperatura em sólidos no regime permanente com geração de calor.

A seguir, é apresentado o resultado com termo transiente $\frac{\partial T}{\partial t}$ tendendo a um estado permanente. As condições iniciais $t = 0s$ atribuídas aos nós sem condição de contorno foram de um valor inicial de $0^{\circ}C$. A Figura 11 apresenta a evolução do campo de temperaturas em função do tempo. É possível observar que a solução numérica converge para a solução analítica formando um perfil parabólico. O erro relativo médio calculado para este caso foi de 0.21101% e com desvio padrão de 0.40994%.



Fonte: Autor.

Figura 11: Comparação de resultado das soluções numéricas e analítica do problema de transporte de temperatura em sólidos no regime transiente com geração de calor.

5.2.3 Equação de Poisson com Condições de Contorno de Dirichlet e Neumann

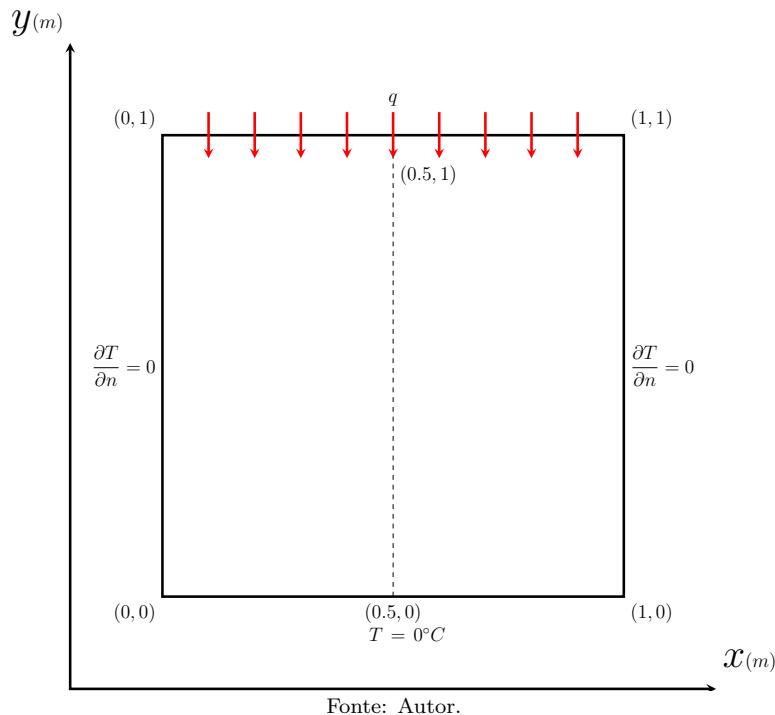
Este caso foi escolhido para validar a solução de problemas com condições de contorno de Neumann. Trata-se de uma placa com temperatura fixa em uma das paredes e no lado oposto é definido um valor para o fluxo de calor presente. Toma-se uma seção da placa para observar os resultados e compará-los com um problema unidimensional de uma barra com as mesmas condições presentes.

A equação de governo é novamente a equação de Poisson (Eq. 5.7), e sua solução analítica para uma barra unidimensional foi calculada pelo método de coeficientes a determinar e é dada por:

$$T(x) = \frac{Q}{k} \left(\frac{-x^2}{2} + Lx \right) - \frac{q}{k}x + T_0 \quad (5.9)$$

onde q é o fluxo de calor na extremidade $x = L$.

As condições de contorno e o domínio bidimensional utilizados na simulação são apresentados na Figura 12. A geração de calor utilizada foi de $Q = -7W/m^3$, a condutividade térmica foi de $k = 5W/m^{\circ}C$ e o fluxo de calor foi de $q = -5W/m^2$.



Fonte: Autor.

Figura 12: Condições de contorno da placa para o problema de Poisson 5.2.3.

Novamente, foi utilizada uma malha triangular linear não estruturada com 3162 elementos e 568 nós. A Figura 13 apresenta o campo de temperatura, onde os eixos x e y representam o domínio e o eixo z é a distribuição de temperatura variando de 0 (cor azul) a 0.35 (vermelha).

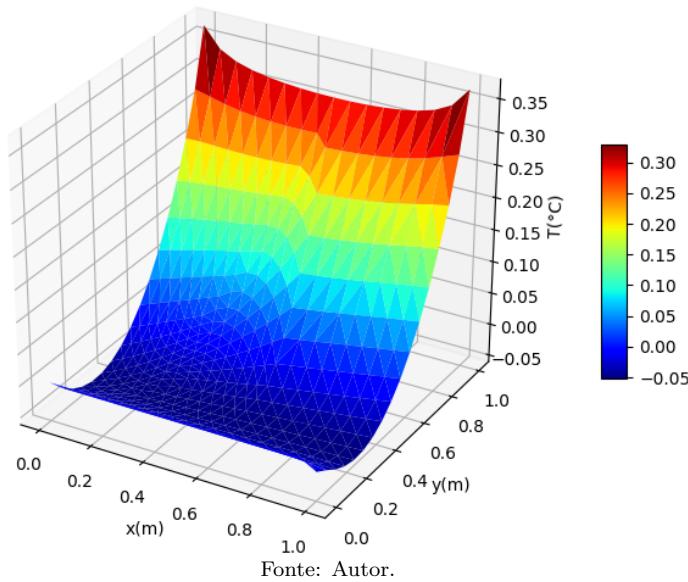
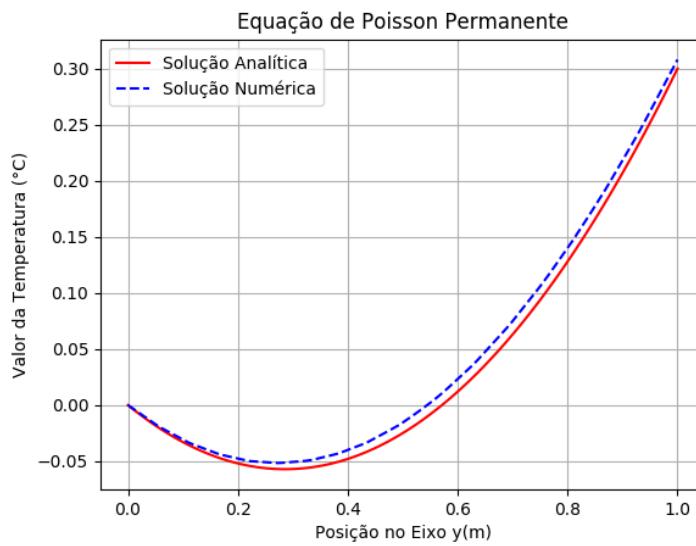


Figura 13: Distribuição de temperaturas na placa da solução do problema permanente de Poisson 5.2.3.

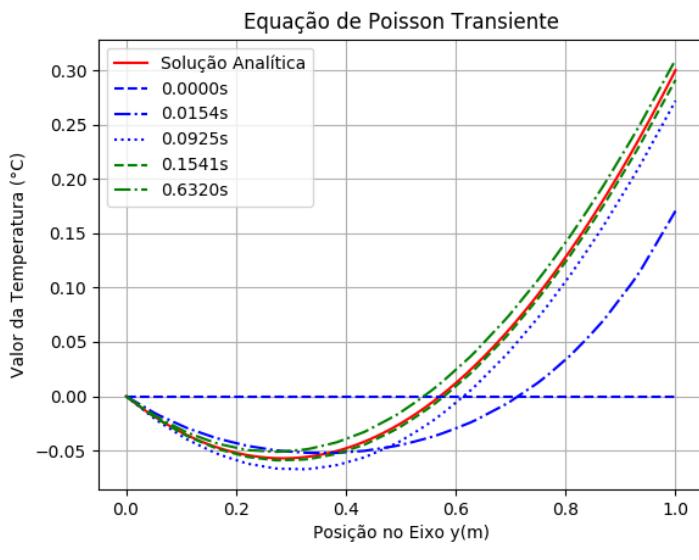
A comparação entre os resultados da solução analítica (Eq. 5.9) e a solução numérica, para a seção $x = 0.5m$, é apresentada na Figura 14, onde pode-se observar que ambas possuem o mesmo perfil. A semelhança entre elas é quantificada pelo erro relativo médio que foi de 0.427% e com desvio padrão de 0.8414%:



Fonte: Autor.

Figura 14: Comparação de resultado das soluções numérica e analítica do problema de transporte de temperatura em sólidos no regime permanente com geração e fluxo de calor.

As condições iniciais $t = 0\text{s}$ atribuídas aos nós sem condição de contorno foram de um valor inicial de 0°C . A Figura 15 apresenta a evolução do campo de temperaturas em função do tempo. É possível observar que a solução numérica converge para a solução analítica. O erro relativo médio calculado para este caso foi de 0.31% e com desvio padrão de 0.9205%.



Fonte: Autor.

Figura 15: Comparação de resultado das soluções numéricas e analítica do problema de transporte de temperatura em sólidos no regime permanente com geração e fluxo de calor.

5.3 Validações de Problemas em Fluídos

5.3.1 Escoamento de Poiseuille

Como indicado anteriormente, foi utilizado o Método de Elementos Finitos para a solução do campo de velocidades no espaço, e o Método de Diferenças Finitas no tempo.

Este é um dos primeiros exemplos dados ao estudar-se a mecânica dos fluidos e a equação de Navier-Stokes(Eq. 2.29), pois trata-se de uma configuração geométrica muito simples. O escoamento de Poiseuille também é conhecido como escoamento entre placas paralelas, já que essa é exatamente a descrição de sua forma e as placas são tomadas como estacionárias em relação ao fluido.

O escoamento ocorre entre duas placas paralelas de comprimento infinito com uma distância constante entre elas. É tomado um fluido ideal, isto é, newto-

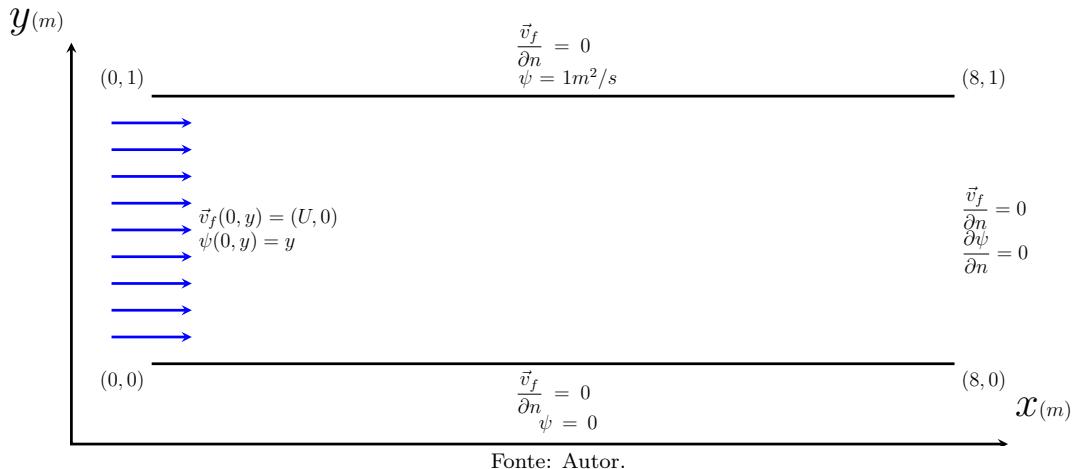
niano, incompressível e em estado permanente com seu perfil plenamente desenvolvido. As condições de contorno são definidas para a função corrente e a velocidade. Para os valores iniciais nos nós sem condição de contorno foram arbitrados como nulos.

Como apresentado por Pontes e Norberto (2010) [19], o sistema de equações de governo deste escoamento é o sistema corrente-vorticidade 2.42, e a solução analítica do perfil de velocidades é dada por:

$$v_x(y) = \frac{4U_{max}}{L^2}y(L-y) \quad (5.10)$$

onde v_x é velocidade do fluido na direção do eixo x , L é o comprimento das placas e U_{max} é a velocidade máxima do escoamento.

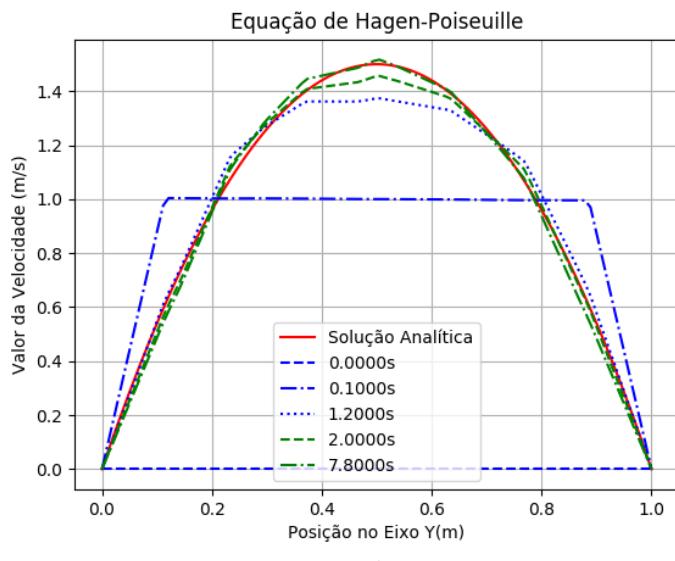
Para esta simulação, foram usadas as condições de contorno e o domínio bidimensional apresentados na Figura 16. A velocidade máxima U_{max} é tomada em função da velocidade de entrada U na relação $U_{max} = 1.5U$. Foi escolhido um valor para o número de Reynolds(Eq. 2.46) de $Re = 1$ e um intervalo de tempo $dt = 0.1s$.



Fonte: Autor.

Figura 16: Condições de contorno de um escoamento entre placas paralelas de Poiseuille 5.3.1.

A Figura 17 apresenta a evolução do perfil de velocidades de uma seção tomada em $x = 0.75L = 6.0m$ em função do tempo. Foi utilizada uma malha triangular linear não estruturada com 4200 elementos e 781 nós. Nota-se que a solução numérica converge para a solução analítica ao se aproximar do estado permanente. O erro relativo médio calculado para este caso foi de 4.587% e com desvio padrão de 5.3501%.



Fonte: Autor.

Figura 17: Comparação de resultado das soluções numéricas e analítica 5.10 do problema de corrente-vorticidade no regime permanente.

5.3.2 Escoamento de Couette

Similar ao escoamento de Poiseuille 5.3.1, entre duas placas paralelas de comprimento infinito, separadas por uma distância constante. Porém, neste caso, as placas possuem uma velocidade relativa entre si, ou seja, estão em movimento. Novamente, utiliza-se a aproximação do escoamento para um fluido newtoniano, incompressível e em estado permanente com seu perfil plenamente desenvolvido. As condições de contorno são definidas para a função corrente e a velocidade, com

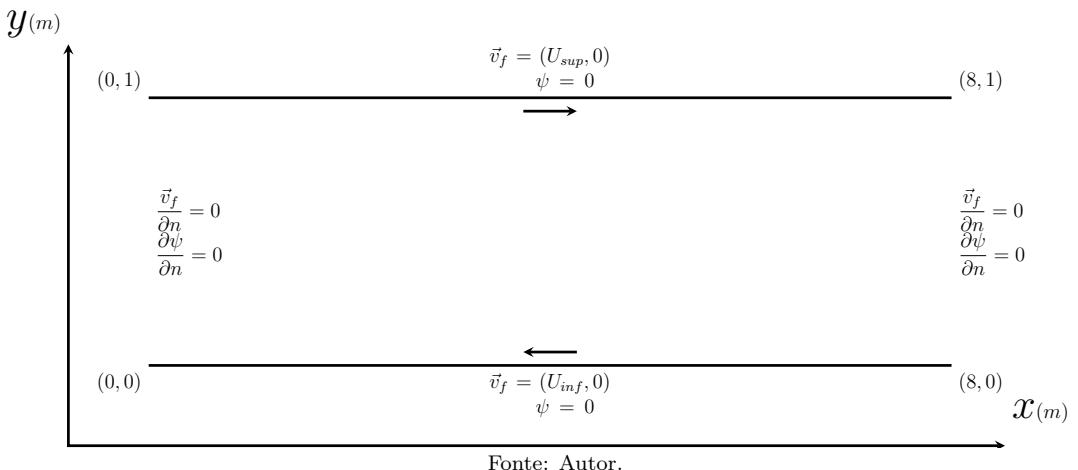
os valores iniciais nos nós sem condição de contorno arbitrados como nulos.

Novamente trabalha-se com o sistema corrente-vorticidade 2.42 e a solução analítica do perfil de velocidades é apresentada por Pontes e Norberto (2010) [19]:

$$v_x(y) = \frac{U_{sup} - U_{inf}}{L} y + U_{inf} \quad (5.11)$$

onde U_{sup} e U_{inf} são as velocidades das placas superior e inferior, respectivamente.

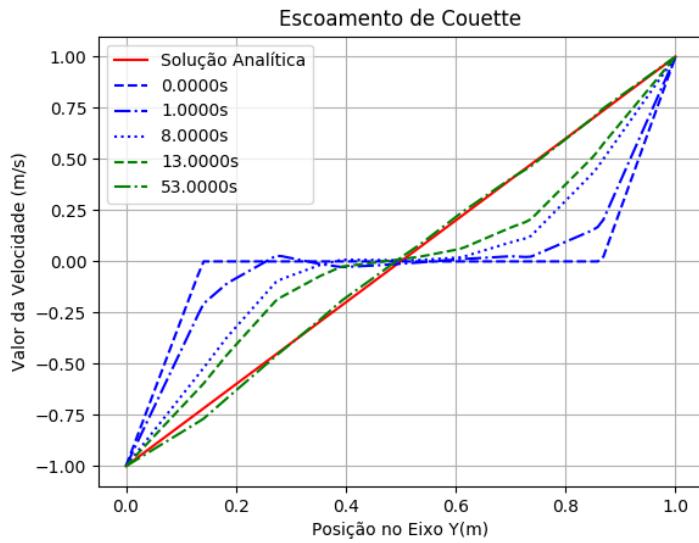
Para esta simulação, foram usadas as condições de contorno e o domínio bidimensional apresentados na Figura 18. Foram tomadas as velocidade superior como $U_{sup} = 1m/s$ e $U_{inf} = -1m/s$ para um escoamento com número de Reynolds de $Re = 1$ e um intervalo de tempo $dt = 1.0s$.



Fonte: Autor.

Figura 18: Condições de contorno de um escoamento entre placas paralelas de Couette 5.3.2.

A Figura 19 apresenta a evolução do perfil de velocidades de uma seção tomada em $x = 0.5L = 4.0m$ em função do tempo. Novamente utilizou-se uma malha triangular linear não estruturada com 4200 elementos e 781 nós. Pode-se observar que a solução numérica converge para a solução analítica ao se aproximar do estado permanente. O erro relativo médio calculado para este caso foi de 1.4251% e com desvio padrão de 27.934%.



Fonte: Autor.

Figura 19: Comparação de resultado das soluções numéricas e analítica 5.11 do problema de corrente-vorticidade no regime permanente.

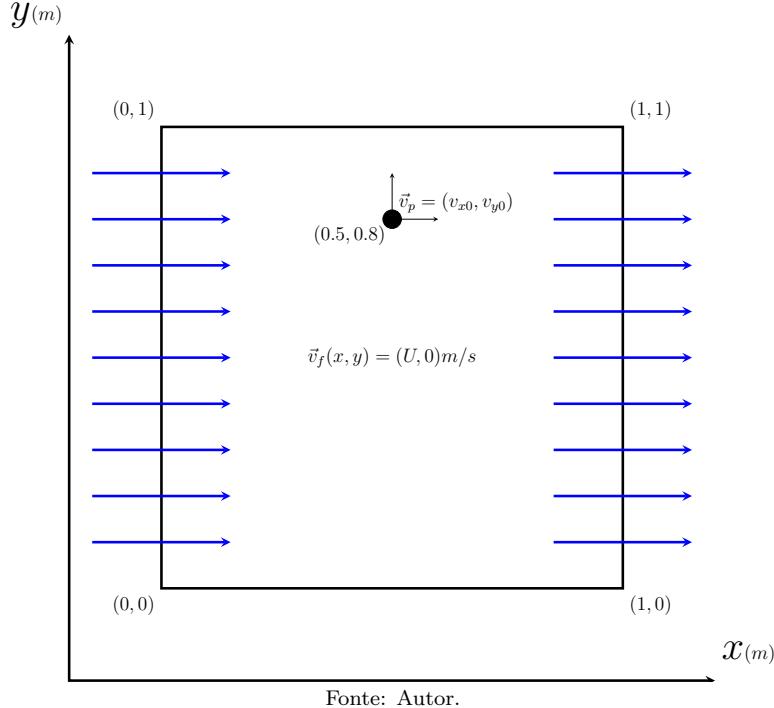
5.4 Validações de Problemas em Partículas

A simulação das forças é realizada sobre uma partícula isolada inserida em um malha com um escoamento com campo de velocidades constante em função da posição da partícula no eixo y , para facilitar os cálculos das soluções analíticas. O campo de velocidades é arbitrado e não calculado como anteriormente.

Conforme indicado, foi utilizado o Método de Diferenças Finitas para discretização e solução da dinâmica das partículas.

Na Figura 20, é apresentado o domínio utilizado para os casos de validação das forças que atuam sobre a partícula, como também o campo de velocidade do fluido e a velocidade da partícula, além da sua posição inicial. Para o caso da força de sustentação, foi necessário a utilização de um outro campo de velocidade conforme será apresentado mais a frente. Enquanto que para o caso da força virtual, foi necessário assumir que o campo de velocidade do fluido dependesse da

variável temporal. O domínio foi discretizado utilizando uma malha triangular linear não estruturada com 2304 elementos e 417 nós.



Fonte: Autor.

Figura 20: Condições de contorno de uma partícula isolada sob efeito de uma força.

A validação das forças é feita individualmente, para que se possa comparar os valores do movimento simulado da partícula com a curva da solução analítica esperada. Os parâmetros definidos para as simulações foram: uma partícula com diâmetro $d_p = 0.001m$, densidade $\rho_p = 30000Kg/m^3$, variação de tempo $dt = 1.5625e^{-6}s$, tempo total $t_{max} = 0.4s$, em um fluido com densidade $\rho_f = 1000Kg/m^3$ e viscosidade dinâmica $0.89e^{-3}Pa.s$.

5.4.1 Força Gravitacional

A força gravitacional é a primeira e mais simples implementação de uma força. A partícula é acelerada pela constante de aceleração gravitacional. A valida-

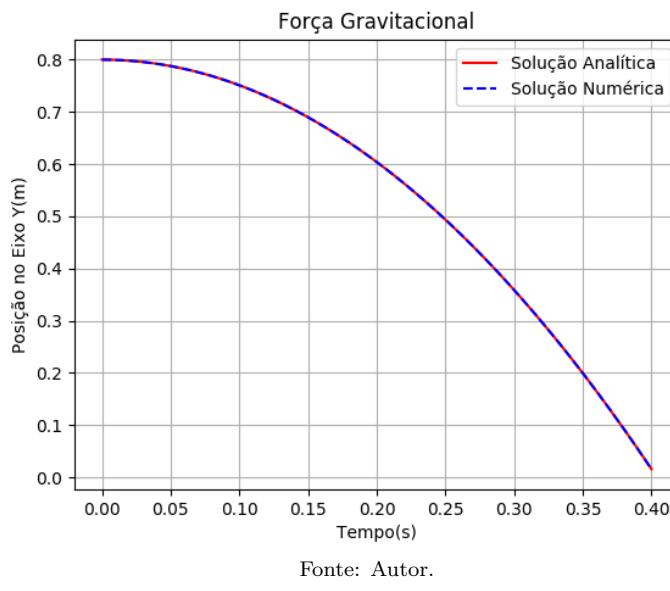
ção deste caso permite verificar a base da estrutura de movimentação de partículas do código.

Para o caso de uma partícula isolada sob efeito da força gravitacional, a solução analítica para a posição da partícula no eixo y foi encontrada pelo método de coeficientes a determinar, e está apresentada a seguir:

$$y(t) = -\frac{g}{2}t^2 + v_{y0}t + p_{y0} \quad (5.12)$$

onde g é a aceleração da gravidade, tomada como $g = 9.80665m/s^2$, v_{y0} é a velocidade e p_{y0} a posição inicial da partícula no eixo y .

Para a força gravitacional, utilizou-se uma malha com as condições de contorno e a posição inicial da partícula apresentadas na Figura 20. A velocidade inicial tomada foi $v_{y0} = 0$. Na Figura 21 pode-se observar a evolução da posição da partícula no eixo y em função do tempo. Foi utilizado um campo de velocidades com valor constante de $U = 2m/s$. O erro relativo médio calculado para este caso foi de $7.706e^{-4}\%$ e com desvio padrão de $1.753e^{-3}\%$.



Fonte: Autor.

Figura 21: Comparação de resultado da solução numérica e analítica 5.12 do percurso de uma partícula em queda livre.

5.4.2 Força de Arrasto

A força de arrasto é a principal fonte de movimentação de uma partícula quando submersa em um fluido em movimento. A partícula é movimentada pela força de cisalhamento do fluido em transito que exerce atrito sobre sua superfície. Isto causa um efeito de arrasto, ou carregamento, devido a condição de não escorregamento da superfície da partícula.

Como pode-se analisar pela equação da força (Eq. 5.13), quanto maior for a diferença entre os vetores de velocidade da partícula e do fluido maior será a força exercida sobre ela. E ao se aproximarem da mesma velocidade, a força é reduzida até o ponto de se tornar nula quando a partícula se encontrar com a mesma velocidade do escoamento no mesmo nó.

Neste caso, a validação é feita para analisar o comportamento da força em relação as propriedades da partícula.

A solução analítica para a posição da partícula sob efeito da força de arrasto no eixo x foi obtida através do método da variação de parâmetros, com auxílio da ferramenta *WolframAlpha*, mostrada a seguir:

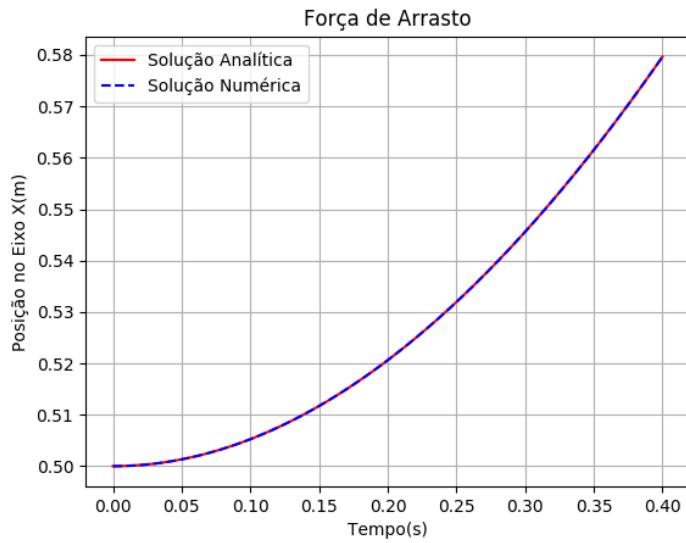
$$x(t) = \frac{m_p}{c_d} (U - v_{x0}) \left(1 - e^{-\frac{c_d}{m_p} t} \right) + v_{x0}t + p_{x0} \quad (5.13)$$

define-se c_d como:

$$c_d = 3\pi\mu_f d_p \quad (5.14)$$

onde m_p é a massa da partícula, μ_f é a viscosidade dinâmica do fluido, d_p é o diâmetro da partícula, v_{x0} é a velocidade e p_{x0} a posição inicial da partícula no eixo x .

Foi utilizada a malha padrão com as condições de contorno e a posição inicial da partícula apresentadas na Figura 20, com velocidade inicial $v_{x0} = 0$. Demonstra-se na Figura 22 a evolução da posição da partícula no eixo x em função do tempo. Foi utilizado um campo de velocidades com valor constante de $U = 2m/s$. Pode-se observar que a partícula é acelerada até atingir a velocidade do fluido, como se era esperado. O erro relativo médio calculado para este caso foi de $3.083e^{-5}\%$ e com desvio padrão de $1.6723e^{-5}\%$.



Fonte: Autor.

Figura 22: Comparação de resultado da solução numérica e analítica 5.13 do percurso de uma partícula em movimento de arrasto em um escoamento.

5.4.3 Força de Sustentação

A força de sustentação é a força que permite que a partícula se mantenha elevada quando está sob efeito de um fluido em movimento. Ela é causada por um efeito de queda de pressão em uma região com maior velocidade, o que gera um gradiente de pressão e enfim a força.

Portanto, esta força afeta a partícula somente quando há um gradiente de velocidade ao longo dos contornos da partícula, ou seja, ela só ocorre quando o campo de velocidades não for constante em volta da partícula. Esta é a mesma força que possibilita o voo dos aviões, pois os perfis de suas asas geram o gradiente de velocidades necessário.

A força de sustentação possui a seguinte solução analítica para a posição da partícula no eixo y semelhante à força de arrasto (Eq. 5.15), contudo, com diferentes constantes. Esta solução foi obtida através do método de variação de

parâmetros, realizado com a ferramenta *WolframAlpha*, resultando em:

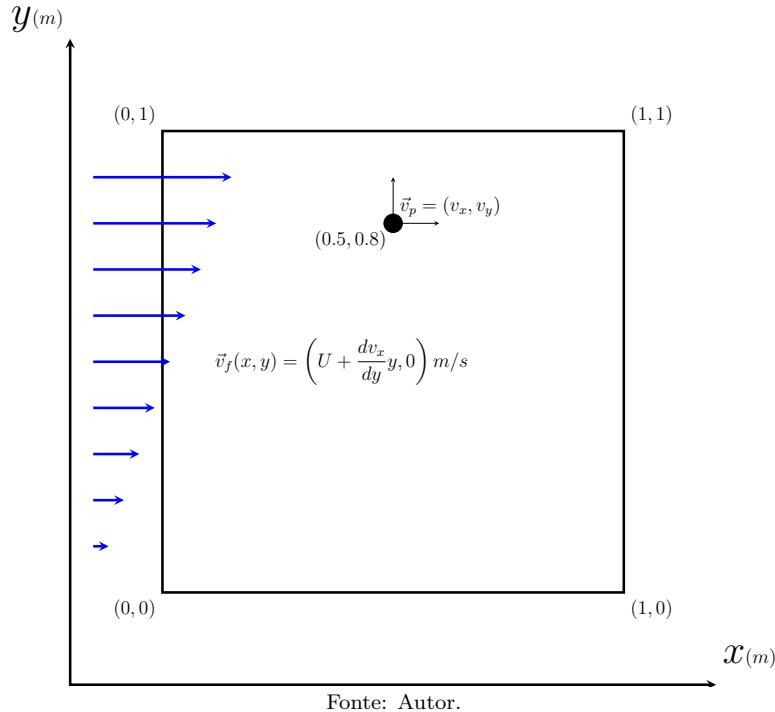
$$y(t) = \frac{m_p}{c_l} (0 - v_{y0}) \left(1 - e^{-\frac{c_l}{m_p} t} \right) + v_{y0}t + p_{y0} \quad (5.15)$$

define-se c_l como:

$$c_l = 1.61 \mu_f d_p \sqrt{Re_G} \quad (5.16)$$

onde Re_G é o número de Reynolds de cisalhamento(Eq. 2.52) ρ_f é a densidade do fluido e $\frac{dv_x}{dy}$ é a variação da velocidade na partícula sobre o eixo perpendicular ao movimento.

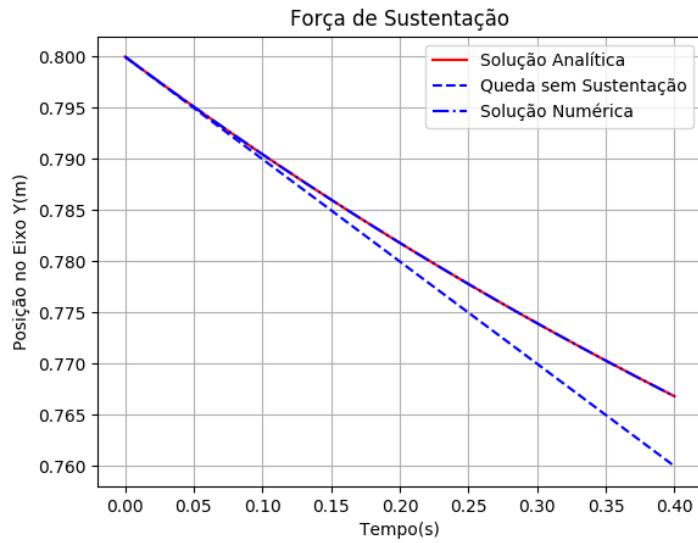
Como é apresentado na Figura 23, foi utilizado a mesma malha que nos casos anteriores, porém, o campo de velocidade do fluido foi alterado, para que fosse possível observar os efeitos da força de sustentação com mais clareza. As condições utilizadas são:



Fonte: Autor.

Figura 23: Condições de contorno de uma partícula isolada sob efeito da força de sustentação.

Demonstra-se na Figura 24 a evolução da posição da partícula no eixo x em função do tempo. Foi utilizado um valor base para o campo de velocidades de $U = 2\text{m/s}$, uma velocidade inicial $v_{y0} = -0.1\text{m/s}$, e um gradiente de velocidades no eixo y de $\frac{dv_x}{dy} = 100\text{m/m.s}$ para auxiliar a comparação. Também foi incluída uma curva que demonstra a trajetória da partícula sem o efeito da força de sustentação, para que se possa notar que a partícula é desacelerada e não acompanha esta curva. O erro relativo médio calculado para este caso foi de $1.9241e^{-6}\%$ e com desvio padrão de $1.1167e^{-6}\%$.



Fonte: Autor.

Figura 24: Comparação de resultado da solução numérica e analítica 5.15 do percurso de uma partícula em movimento de sustentação em um escoamento.

5.4.4 Força de Massa Virtual (*Added Mass*)

A força de massa virtual é uma força de reação entre o fluido e a movimentação da partícula durante o escoamento. O valor dessa força está relacionado a massa de fluido que estaria se deslocando na posição da partícula.

Para encontrar-se a solução analítica da força de massa virtual que pudesse ser observada e comparada com os resultados das simulações, sendo não nula, foi preciso assumir um campo de velocidades que variasse no tempo. Para isso, define-se que a aceleração do campo de velocidades do escoamento é constante $\frac{d\vec{v}}{dt} = \vec{a} = const.$. Portanto, obtém-se a solução analítica da força de massa virtual para a posição da partícula no eixo x , com o método de variação de parâmetros, feito com a ferramenta *WolframAlpha*, com o seguinte resultado:

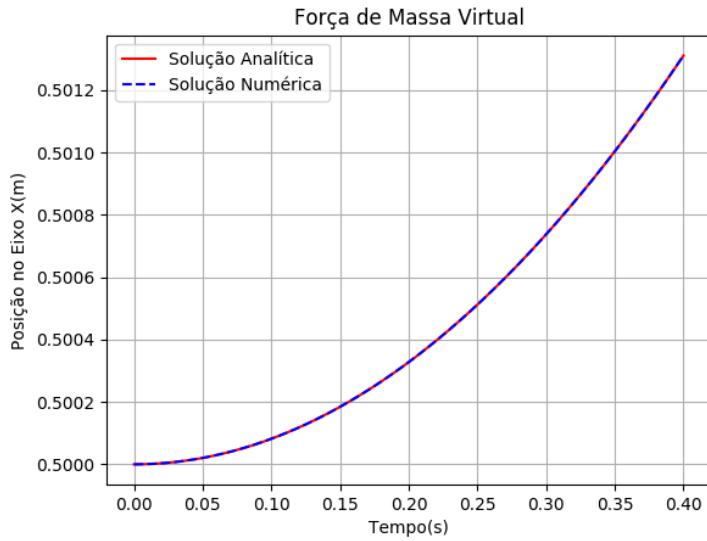
$$x(t) = \frac{a_x c_m}{2(c_m + m)} t^2 + v_{x0} t + p_{x0} \quad (5.17)$$

define-se c_m como:

$$c_m = \frac{1}{2} \rho_f V_p \quad (5.18)$$

onde a_x é a aceleração, ou variação da velocidade no tempo, do campo de velocidades do escoamento no eixo x e V_p é o volume da partícula.

Novamente, foi utilizado a mesma malha e condições dos casos anteriores conforme apresentado na Figura 20, além disso, a condição inicial da componente tangencial da velocidade foi $v_{x0} = 0$. Revela-se na Figura 25 a evolução da posição da partícula no eixo x em função do tempo. Foi utilizado um campo de velocidades que varia no tempo com valor constante, como explicado anteriormente, na forma de $U = 2m/s + a_x t$, onde $a_x = 1m/s^2$. Pode-se observar que a partícula é acelerada até atingir a velocidade do fluido, como se era esperado. O erro relativo médio calculado para este caso foi de $6.6574e^{-3}\%$ e com desvio padrão de $1.1362e^{-3}\%$.



Fonte: Autor.

Figura 25: Comparaçāo de resultado da solução numérica e analítica 5.17 do percurso de uma partícula em movimento de aceleração em um escoamento.

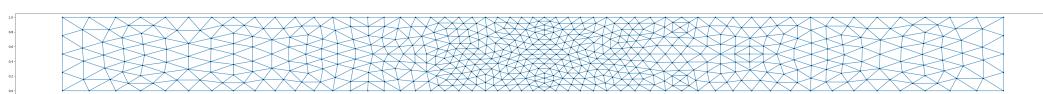
6 RESULTADOS DE SIMULAÇÕES

6.1 Introdução

Neste capítulo serão apresentadas as simulações realizadas para analisar o comportamento de partículas presentes em turbomáquinas em ação. Foram escolhidos cinco tipos de geometria para serem simulados:

- **Canal reto.**

Uma geometria simples, que busca simular o comportamento básico das partículas sobre efeito de um escoamento. A malha utilizada tem 1072 elementos definidos sobre 593 nós, exposta na Figura 26.

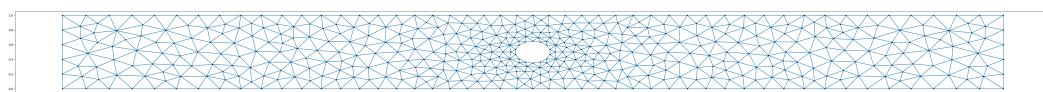


Fonte: Autor.

Figura 26: Malha da simulação em um canal reto.

- **Canal com um obstáculo circular no centro.**

Esta geometria visa simular o efeito de obstáculos, ou até mesmo as próprias partículas no escoamento. Porém, não é utilizada neste trabalho a implementação *two-way*, que considera os efeitos das partículas no escoamento. Neste caso a simulação foi aplicada a uma malha de 868 elementos e 494 nós.

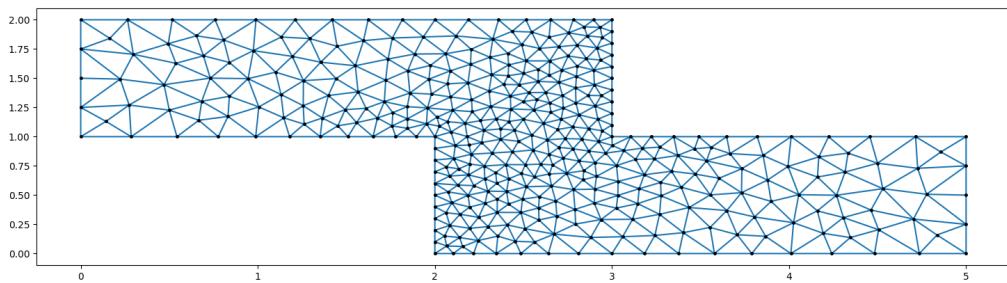


Fonte: Autor.

Figura 27: Malha da simulação em um canal com um obstáculo.

- **Canal com degrau.**

O escoamento em um canal com um degrau, ou uma diferença brusca de percurso, possui diversos efeitos interessantes para serem observados. Como, por exemplo, nas regiões de curva, pode fazer-se observações sobre o comportamento em conexões de tubulações reais. Com uma malha de 676 elementos definidos sobre 380 nós.

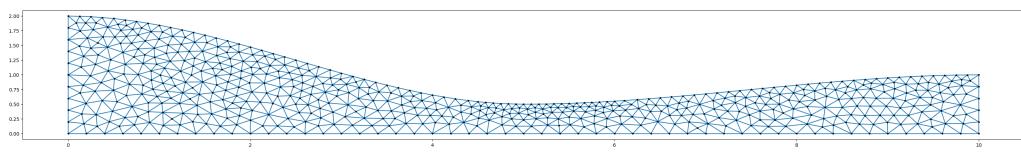


Fonte: Autor.

Figura 28: Malha da simulação em um canal com degrau.

- **Canal com restrição.**

O escoamento em um canal com restrição demonstra os efeitos da redução de seção transversal. Um exemplo deste tipo de escoamento seria em medidores de vazão, como o Tubo de Venturi. Sua malha foi criada com 1047 elementos distribuidos sobre 600 nós:



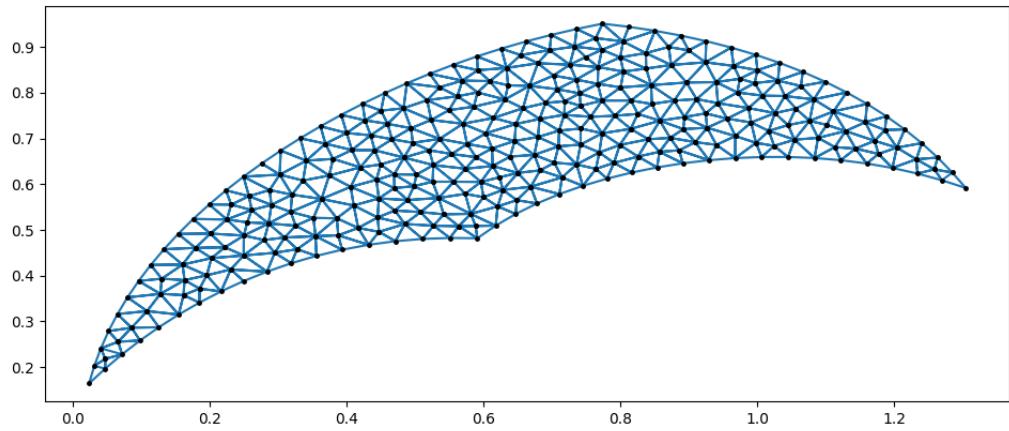
Fonte: Autor.

Figura 29: Malha da simulação em um canal com restrição.

- **Palheta de um Impelidor.**

Finalmente, será analisado o comportamento de partículas em um escoamento presente em uma seção de um impelidor de uma turbomáquina. Neste

trabalho, foi tomado um referencial estacionário na palheta, sem efeito da rotação do impelidor no escoamento. Foi utilizada uma malha de 1058 elementos definidos sobre 531 nós.



Fonte: Autor.

Figura 30: Malha da simulação em uma palheta de um Impelidor.

Foi utilizado o *software ParaView* para exibição dos gráficos de resultados. A escala das figuras mostra o valor da magnitude da velocidade em cada nó.

Em cada simulação, com exceção das simulações na palheta de um impelidor, são inseridas cinco partículas de ouro perfeitamente rígidas, igualmente espalhadas, com $d_p = 1mm = 1.10^{-3}m$ de diâmetro e densidade de $\rho_p = 20000kg/m^3$. Estas partículas estão submersas em um fluido ideal com características: $\mu_f = 50kg/m.s$ de viscosidade dinâmica e densidade de $\rho_f = 50kg/m^3$, para que trabalhe em uma situação com número de Reynolds reduzido, e menor que a restrição da equação BBO(Eq. 2.6).

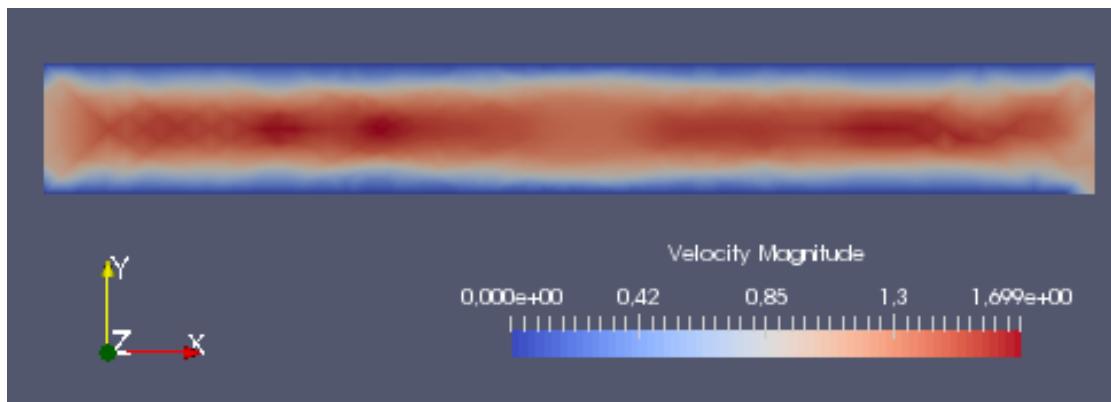
As partículas são iniciadas com velocidade nula e em posições próximas à entrada do escoamento, fora de sua região de transição. As simulações foram executadas para um tempo total de 5s com um tempo entre cálculos de $dt =$

$3.33e^{-5}s$. O trajeto de cada partícula é exibido com uma cor diferente para auxiliar a interpretação.

6.2 Simulação Em Um Canal Reto

O escoamento em um canal reto permite visualizar o movimento livre das partículas sob efeito do fluido. A sua configuração é similar ao exemplo de Poiseuille 5.3.1. As condições do escoamento são de um comprimento $L = 8m$ com altura constante de $D = 1m$, o fluido entra com velocidade constante pela região da esquerda com velocidade $\vec{v}_{ent} = (1,0)m/s$. Nas paredes há a condição de não escorregamento, portanto a velocidade nas paredes inferior e superior é nula, $\vec{v}_{sup} = \vec{v}_{inf} = 0$. A corrente possui uma condição de contorno variando de 0 a 1 nas bordas de acordo com sua posição, equivalente a $\psi_{cc} = y$.

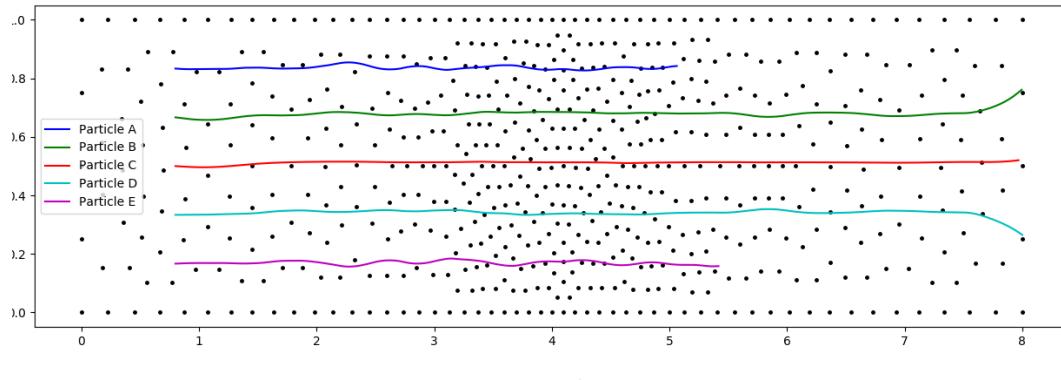
Seu campo de velocidades é apresentado na Figura 31:



Fonte: Autor.

Figura 31: Campo de velocidades de um escoamento em um canal reto.

A trajetória das partículas pode ser analisada na Figura 32:



Fonte: Autor.

Figura 32: Trajeto de partículas inseridas em um escoamento em um canal reto.

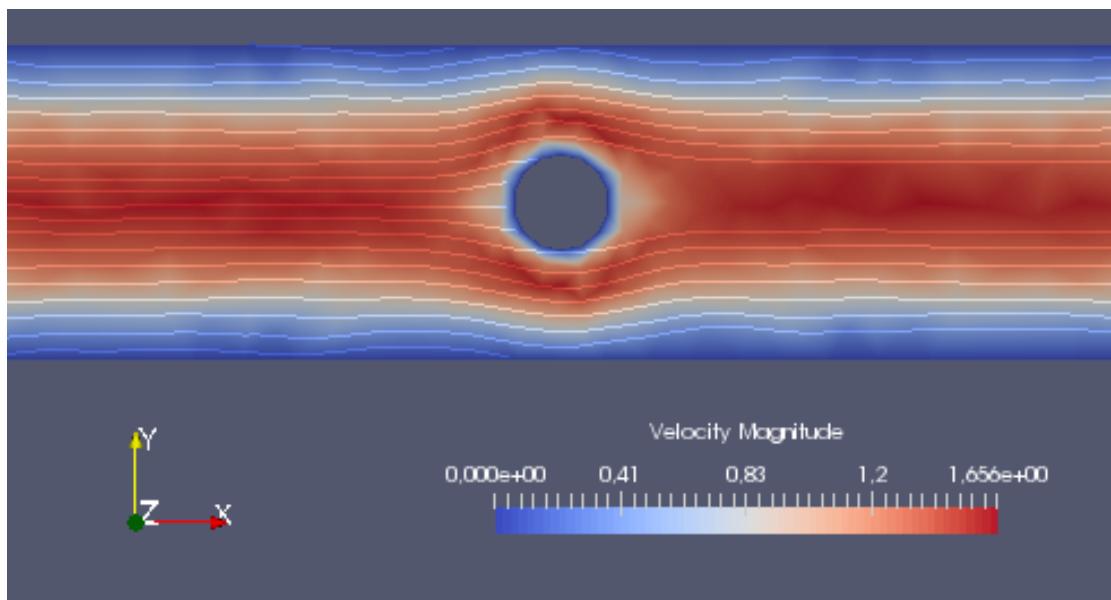
Pode-se observar que as partículas seguem o escoamento como esperado, com uma diferença de distância no percurso causado pelo perfil de velocidades. Na região de saída, as partículas tendem a se espalhar devido às condições de contorno na região.

6.3 Simulação Em Um Canal Com Obstáculo

Em um escoamento com obstáculo, pode-se observar o efeito do obstáculo no campo de velocidades do escoamento, assim como o comportamento das partículas ao interagir com este obstáculo.

Novamente, as condições do escoamento são iguais à simulação anterior 6.2, porém, com as condições de contorno aplicadas também a região do obstáculo. Onde $\vec{v}_{obstaculo} = 0$ e $\psi_{obstaculo} = y$.

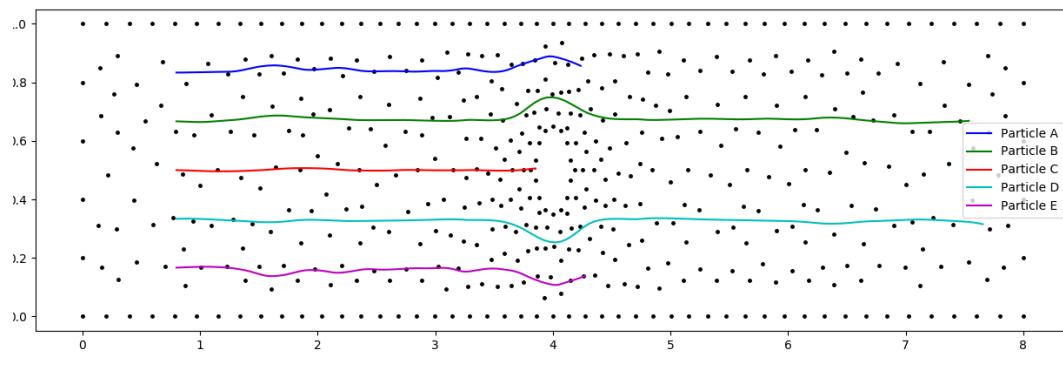
Para auxiliar na visualização do resultado, é exibido em conjunto da velocidade as linhas de corrente do campo, desenhadas com o auxílio de uma ferramenta de desenho das curvas de nível. Observa-se a distorção no campo de velocidades causada pelo obstáculo na Figura 33:



Fonte: Autor.

Figura 33: Campo de velocidades de um escoamento com obstáculo.

E as partículas realizam percursos apresentados na Figura 34:



Fonte: Autor.

Figura 34: Trajeto de partículas inseridas em um escoamento com obstáculo.

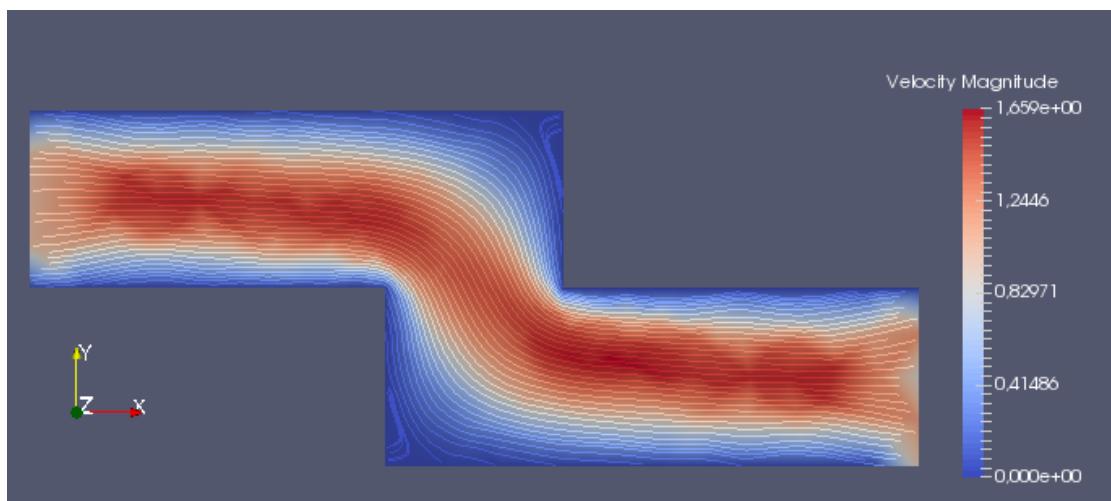
Nesta simulação pode-se observar o efeito do obstáculo que desvia o trajeto das partículas e a colisão de uma delas, que fica retida devido às correntes de velocidade nulas na condição de contorno das paredes.

6.4 Simulação Em Um Canal Com Degrau

No escoamento com degrau há um desvio no trajeto que tende a causar colisões das partículas devido a inércia de seu movimento. Dependendo de suas características, as partículas estão mais propensas a colidirem com a parede direita superior ou serão mais influenciadas pelo fluido e terão sua trajetória alterada.

Para este escoamento são adaptadas as condições anteriores para a geometria de um comprimento $L = 5m$ com altura de entrada e saída de $D = 1m$, o fluido entra com velocidade constante pela região da esquerda com velocidade $\vec{v}_{ent} = (1, 0)m/s$. Nas paredes há a condição de não escorregamento, portanto, a velocidade nas paredes é nula, $\vec{v}_{parede} = 0$. A corrente possui uma condição de contorno variando de 0 a 1 onde ela é 0 para as paredes inferiores e 1 para as superiores. E na região vertical esquerda de entrada toma-se a corrente como $\psi_{cc} = y - 1$, e $\psi_{cc} = y$ na região vertical direita de saída.

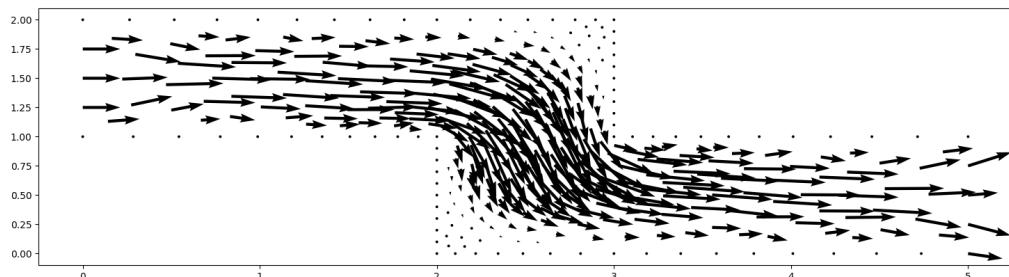
Novamente, são incluídas as curvas de nível da velocidade na exibição do campo de velocidades na Figura 35:



Fonte: Autor.

Figura 35: Campo de velocidades de um escoamento com degrau.

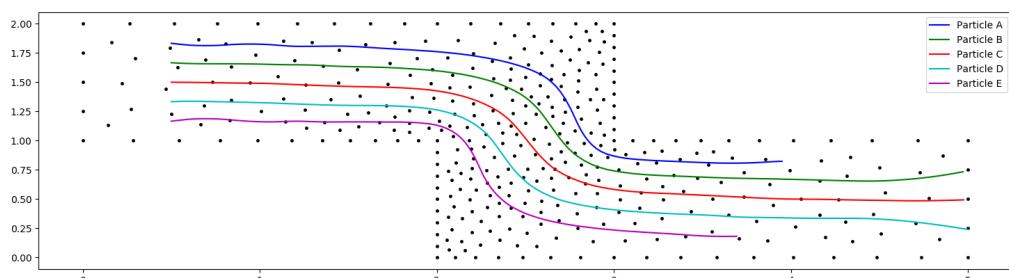
Como outra forma de visualizar o campo de velocidades, são exibidas as velocidades nos nós como vetores em um diagrama de flechas ou *quiver plot*:



Fonte: Autor.

Figura 36: Vetores de velocidades de um escoamento com degrau.

As curvas dos trajetos percorridos pelas partículas nesta simulação são apresentados na Figura 37:



Fonte: Autor.

Figura 37: Trajeto de partículas inseridas em um escoamento com degrau.

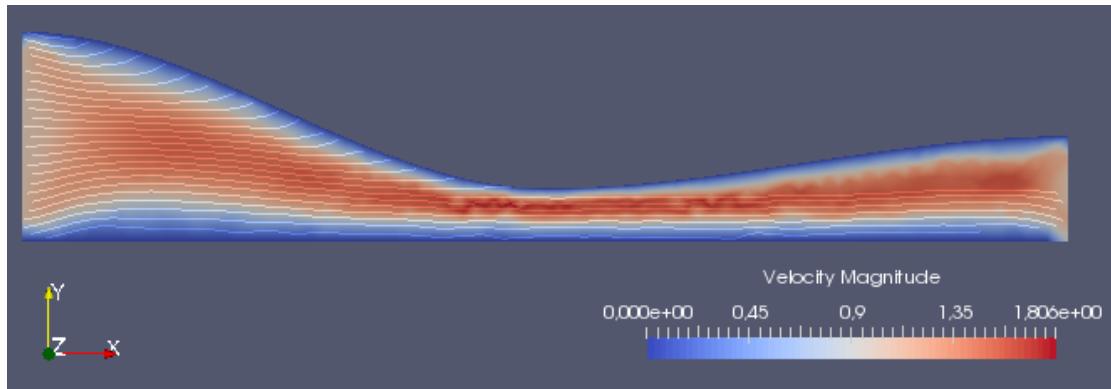
Neste caso, pode-se notar que as partículas seguem a curva realizada pelo escoamento fielmente. Caso as propriedades das partículas fossem alteradas poderia-se encontrar uma situação em que sua inércia fosse grande demais para fazer a curva a tempo e algumas colidiriam na parede.

6.5 Simulação Em Um Canal Com Restrição

No escoamento com restrição há um aumento de velocidade na região restringida. Isto ocorre devido à necessidade de aumentar o fluxo de fluido na região, de maneira a respeitar a Lei da Continuidade (Eq. 2.10).

As condições do escoamento são de um comprimento $L = 10m$ com altura de entrada de $D_{entrada} = 2m$ e uma entrada de saída de $D_{saída} = 1m$. O fluido entra com velocidade constante pela região da esquerda com velocidade $\vec{v}_{ent} = (1, 0)m/s$. Nas paredes há a condição de não escorregamento, portanto, a velocidade nas paredes inferior e superior é nula, $\vec{v}_{sup} = \vec{v}_{inf} = 0$. Novamente a corrente possui uma condição de contorno variando de 0 a 1 nas bordas de acordo com sua posição, equivalente a $\psi_{cc} = y$.

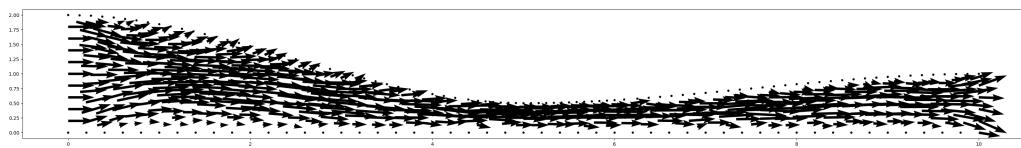
Novamente, são incluídas as curvas de nível da velocidade na exibição do campo de velocidades na Figura 38:



Fonte: Autor.

Figura 38: Campo de velocidades de um escoamento com restrição.

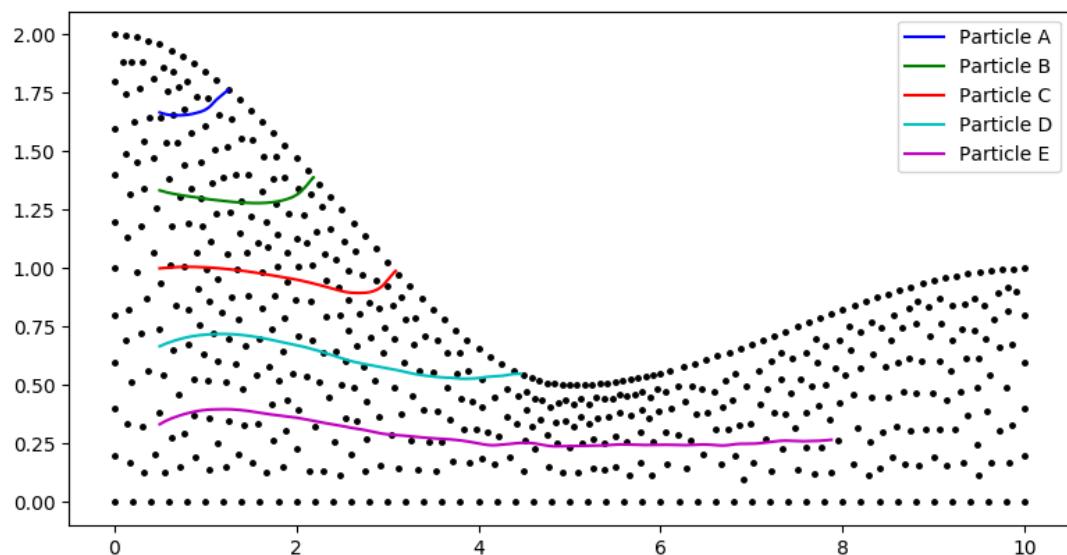
Visualizando-se os vetores do campo de velocidades, pode-se observar o aumento da velocidade na região restrita e um componente vertical da velocidade elevado, na região inicial do escoamento:



Fonte: Autor.

Figura 39: Vetores de velocidades de um escoamento com restrição.

O trajeto das partículas realizado neste caso é demonstrado na Figura 40:



Fonte: Autor.

Figura 40: Trajeto de partículas inseridas em um escoamento com restrição.

Neste caso pode-se observar a colisão das partículas na extremidade superior logo após a entrada devido a uma corrente criada na região de entrada do escoamento. Ao longo do canal, pode se observar que a partícula sofre uma aceleração ao atravessar a região restringida.

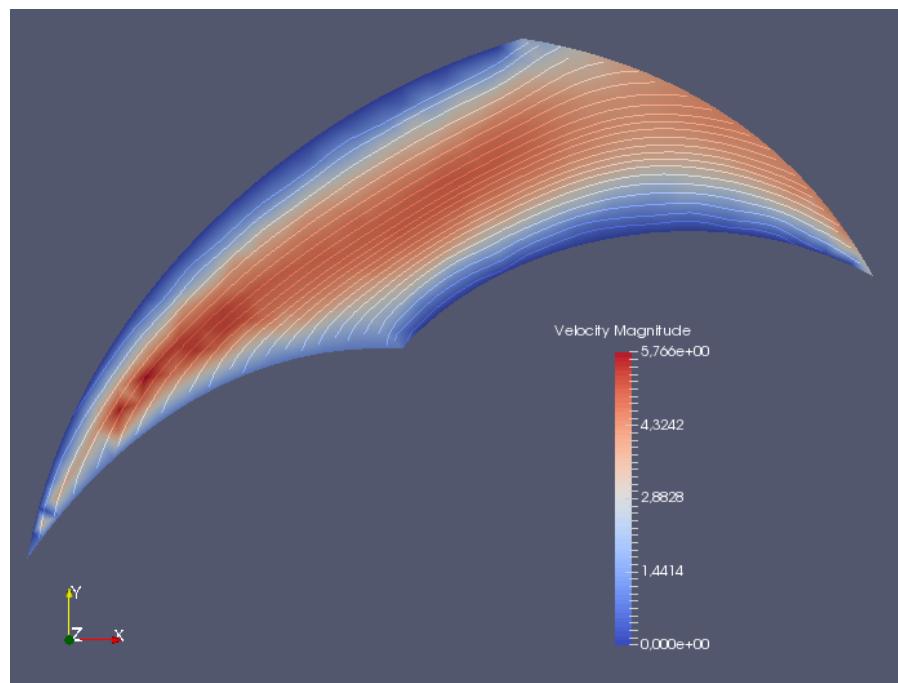
6.6 Simulação Em Uma Palheta de Impelidor

Finalmente, a simulação em uma palheta de impelidor visa representar o comportamento das partículas em uma situação real de trabalho de uma turbomáquina. Foram feitas simulações para três tipos de partículas com diferentes materiais e diâmetros. Foram utilizadas partículas de ouro com $\rho_{p,Au} = 20000\text{kg/m}^3$ de densidade e $d_{p,Au} = 1\text{mm} = 1.10^{-3}\text{m}$ de diâmetro, de ferro, com $\rho_{p,Fe} = 7300\text{kg/m}^3$ de densidade e $d_{p,Fe} = 1\text{mm} = 1.10^{-3}\text{m}$ de diâmetro, e de areia com $\rho_{p,Areia} = 1600\text{kg/m}^3$ de densidade e $d_{p,Areia} = 0.5\text{mm} = 5.10^{-4}\text{m}$ de diâmetro. Com respeito ao tempo entre instantes, foram utilizados os intervalos de $dt = 3.33e^{-5}\text{s}$ para a simulação com partículas de ouro, $dt = 8.11e^{-6}\text{s}$, para a de ferro e $dt = 4.44e^{-7}\text{s}$ para a de areia. Já que quanto menor forem o tamanho e densidade das partículas torna-se necessário um intervalo para garantir a convergência.

A geometria palheta foi extraída de um contorno de uma palheta real de uma bomba exemplo, porém, feita em escala. Foi tomado um arco na seção de entrada gerado com raio de $R_{ent} = 0.5\text{m}$, e a saída com um arco de $R_{sai} = 1.0\text{m}$. A velocidade de entrada do fluido foi atribuída como $\vec{v}_{ent} = (0, 1)\text{m/s}$, no arco de entrada na região inferior. Novamente, utiliza-se a condição de não escorregamento das paredes laterais, portanto, na parede esquerda e direita tem-se $\vec{v}_{parede} = 0$. A função de corrente na parede esquerda é aplicada como 1 e na direita como 0. Para a região de entrada, a corrente é tomada com valor crescente começando por 0 no nó coincidente com a parede direita até 1 com o nó presente na parede esquerda, similarmente é realizado o recíproco para a região de saída.

Neste trabalho, não serão levadas em conta as forças provenientes da rotação da máquina. Portanto, é assumido que o domínio presente da palheta é estacionário e é tomado como referência.

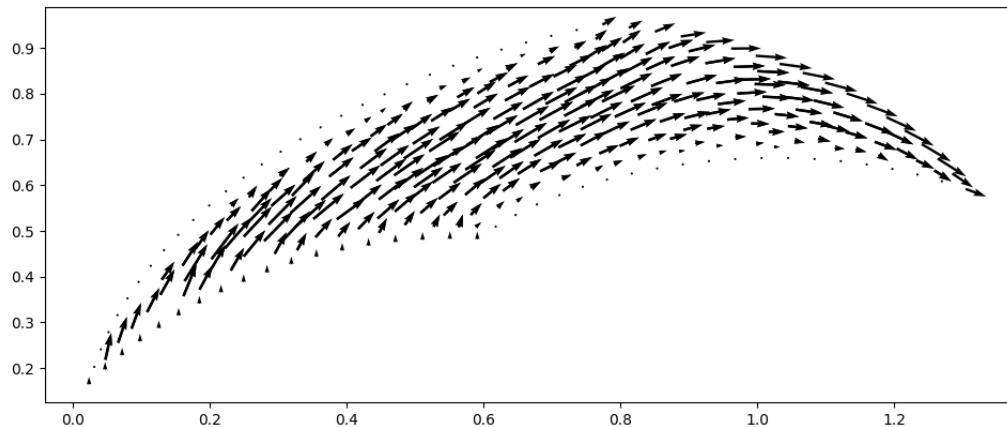
Novamente, são incluídas as curvas de nível da velocidade na exibição do campo de velocidades na Figura 41:



Fonte: Autor.

Figura 41: Campo de velocidades de um escoamento em uma palheta de um impelidor.

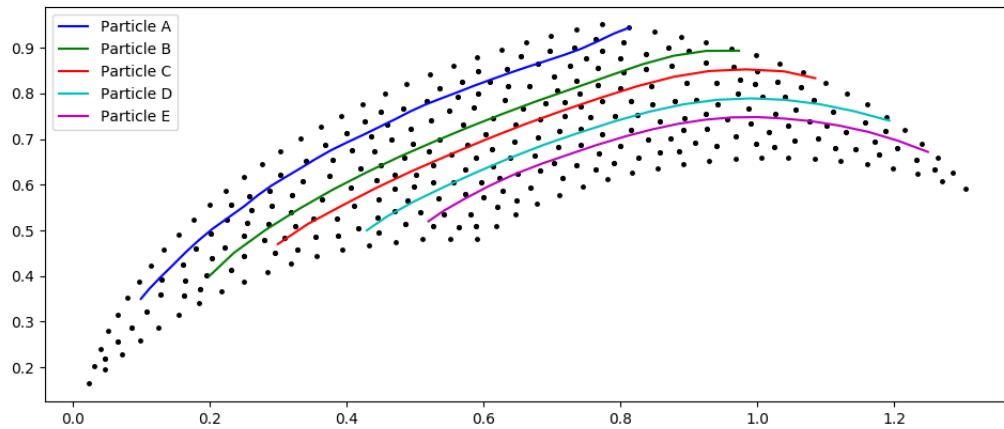
E os vetores das velocidades nos nós:



Fonte: Autor.

Figura 42: Vetores de velocidades de um escoamento em uma palheta de um impelidor.

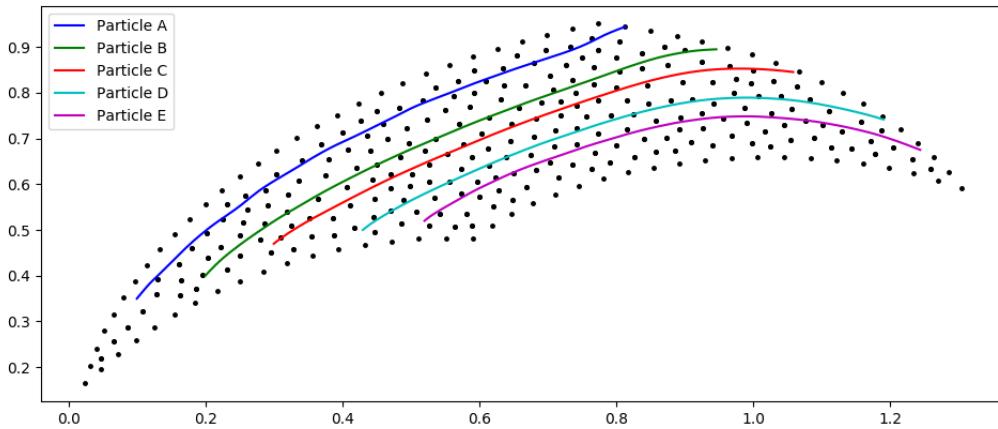
O resultado da simulação com as partículas de ouro na palheta do impelidor resultou no trajeto exibido na Figura 43:



Fonte: Autor.

Figura 43: Trajeto de partículas de ouro inseridas em um escoamento em uma palheta de um impelidor.

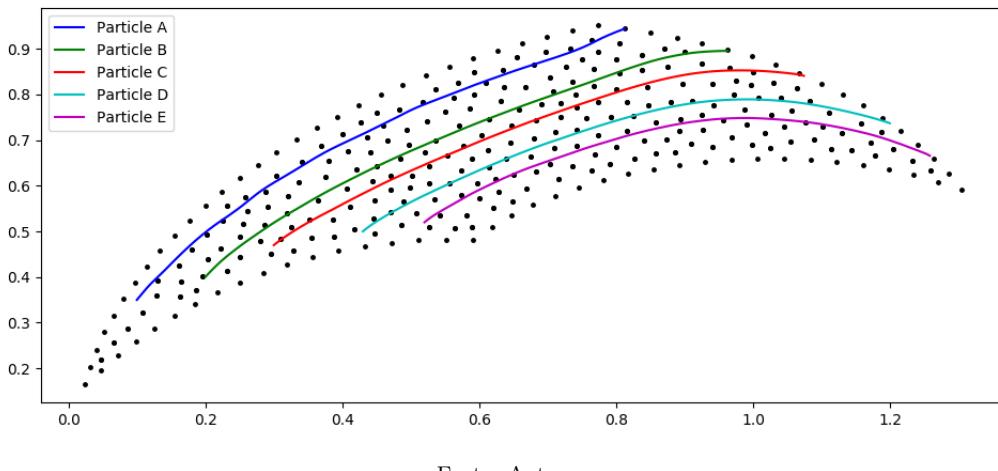
Na simulação de partículas de ferro foi verificado o comportamento demonstrado na Figura 44:



Fonte: Autor.

Figura 44: Trajeto de partículas de ferro inseridas em um escoamento em uma palheta de um impelidor.

E, finalmente, as partículas de areia realizaram a trajetória apresentada na Figura 45:



Fonte: Autor.

Figura 45: Trajeto de partículas de areia inseridas em um escoamento em uma palheta de um impelidor.

Pode-se observar que as partículas seguem ao longo da geometria da palheta,

até mesmo a partícula mais distante acompanha as demais devido a uma região de velocidade elevada presente no canto esquerdo inferior.

As simulações com partículas mais leves levaram um tempo elevado para serem concluídas, devido a sua restrição no intervalo de cálculos. Isto causou uma variação grande no tempo de cálculo das simulações, onde a simulação com partículas de ouro demorou em torno de 2h para ser completada, e as de ferro e areia demoraram em torno de 6h e 12h horas, respectivamente.

CONCLUSÃO

Neste trabalho foram apresentadas as equações de Navier-Stokes, através da fórmula de corrente-vorticidade, e Basset–Boussinesq–Oseen para a simulação de escoamentos multifásicos particulados utilizando o Método de Elementos Finitos para a solução do campo de velocidades e o Método de Diferenças Finitas para a movimentação das partículas. Devido ao uso da formulação de corrente-vorticidade foi possível simplificar os elementos utilizados na malha da simulação para elementos triangulares lineares na forma Euleriana, com a posição dos nós da malha estacionária. Isto facilitou o desenvolvimento e implementação do código numérico. A interação entre as partículas e o escoamento foi implementada do tipo *one-way*, sem influência da presença das partículas no escoamento.

O código foi criado pelo autor utilizando a linguagem de programação *Python*. Foram importadas ferramentas de análise de dados, visualização gráfica e solução algébrica para auxiliar na otimização do código. O código foi construído com o paradigma de orientação a objetos em mente, e permitiu a estruturação do código em uma biblioteca de solução de escoamentos particulados de uso livre. A biblioteca possui uma licença de uso sem restrições e permite importações de estruturas de malha no formato *.msh* ou criação de uma malha através das coordenadas dos pontos.

As simulações de validação para os problemas de troca térmica em 2D das equações de Laplace e Poisson, encontrados na 5.2, e para os problemas de fluidos de Poiseuille e Couette, apresentados na 5.3, demonstraram resultados próximos aos esperados em comparação com as soluções analíticas de cada caso. Estes casos serviram como comprovação da acurácia do Método de Elementos Finitos. Para o Método de Diferenças Finitas, foi validada cada força aplicada a partícula separadamente, e estas também obtiveram bons resultados.

O objetivo principal deste trabalho foi estudar o comportamento das partículas inseridas em um escoamento permanente de uma turbomáquina. Para isto foram então simulados escoamentos com malhas de diferentes geometrias bidimensionais: um canal reto, um canal com um obstáculo, um canal com degrau, um canal com restição, e uma palheta de impelidor. Para este caso final, sendo o principal, foram realizadas outras simulações com diferentes características para as partículas. Verificou-se que o comportamento das partículas imersas nestas situações segue o esperado, com a dominação da força de arrasto no comportamento das partículas em relação as demais forças.

Para trabalhos futuros, apresentam-se em seguida algumas possibilidades para linhas de desenvolvimento:

- Adicionar mais forças exercidas às partículas ao modelo, como a força de Magnus.
- Adicionar efeitos das partículas na solução do escoamento (esquema *two-way*).
- Adicionar efeito da cada partícula sobre as demais (esquema *four-way*).
- Reformulação das equações para utilizar outra formulação, como a **ALE**.
- Adaptação do código para permitir solução de problemas tridimensionais (3D).

REFERÊNCIAS

- [1] LEWY, R. C. K. F. H. Über die partiellen differenzengleichungen der mathematischen physik. *Mathematische Annalen*, Springer, v. 100, p. 32–74, 1928. ISSN 0025-5831,1432-1807.
- [2] BOOLE, G. *A Treatise on the Calculus of Finite Differences*. 1. ed. [S.l.]: Cambridge University Press, 1859. (Cambridge Library Collection - Mathematics). ISBN 9781108000925.
- [3] STIFFNESS and Deflection Analysis of Complex Structures. *Journal of the Aeronautical Sciences (Institute of the Aeronautical Sciences)*, v. 23, p. 805–823, 1956. ISSN 1936-9956.
- [4] LEWIS, R. W.; NITHIARASU, P.; SEETHARAMU, K. *Fundamentals Of The Finite Element Method For Heat And Fluid Flow*. [S.l.]: Wiley, 2004. ISBN 9780470847886,0-470-84788-3,0-470-84789-1.
- [5] ANJOS, G. R.; MANGIAVACCHI, N.; PONTES, J. Arbitrary lagrangian–eulerian method for two-phase flows. In: _____. *Encyclopedia of Two-Phase Heat Transfer and Flow II*. [S.l.: s.n.], 2015. cap. Chapter 3, p. 75–110.
- [6] BAKER, J. Flow-regime transitions at elevated pressures in vertical two-phase flow. Argonne National Lab., Ill. Associated Midwest Universities, Lemont, Ill.
- [7] ELGHOBASHI, S. Particle-laden turbulent flows: direct simulation and closure models. *Flow, Turbulence and Combustion*, Springer, v. 48, p. 301–314, 1991. ISSN 1386-6184,1573-1987.

- [8] BALACHANDAR S.; EATON, J. K. Turbulent dispersed multiphase flow. *Annual Review of Fluid Mechanics*, Annual Reviews, v. 42, p. 111–133, 2010. ISSN 0066-4189,1545-4479.
- [9] ELGHOBASHI, S. On predicting particle-laden turbulent flows. *Flow, Turbulence and Combustion*, Springer, v. 52, p. 309–329, 1994. ISSN 1386-6184,1573-1987.
- [10] SO, S. L. *Instrumentation for Fluid-Particle Flow*. [S.l.]: Noyes Publications, 1999. ISBN 0-8155-1433-6.
- [11] HUSSEIN M. FATHY; TABAKOFF, W. Dynamic behavior of solid particles suspended by polluted flow in a turbine stage. *Journal of Aircraft*, American Institute of Aeronautics and Astronautics, v. 10, p. 434–440, 1973. ISSN 0021-8669,1533-3868.
- [12] TABAKOFF W.; HAMED, A. The dynamics of suspended solid particles in a two-stage gas turbine. *Journal of Turbomachinery*, The American Society of Mechanical Engineers, v. 108, p. 298, 1986. ISSN 0889-504X.
- [13] UZOL, O. et al. Experimental investigation of unsteady flow field within a two stage axial turbomachine using particle image velocimetry. *ASME Turbo Expo 2002: Power for Land, Sea, and Air*.
- [14] GHENAIET, A.; TAN, S. C.; ELDER, R. L. Prediction of an axial turbomachine performance degradation due to sand ingestion. *Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy*, v. 219, n. 4, p. 273–287, 6 2005.
- [15] CENTRIFUGAL Pump Help-John Anspach Consulting. <http://jacpump.files.wordpress.com/2011/04/>

thermalchgo-017w2.jpg. Acessado: 2019-05-15. Disponível em: <<http://jacpump.files.wordpress.com/2011/04/thermalchgo-017w2.jpg>>.

- [16] BLAKE, J. R.; GIBSON, D. C. Cavitation bubbles near boundaries. *Annual Review of Fluid Mechanics*, Annual Reviews, v. 19, p. 99–123, 1987. ISSN 0066-4189,1545-4479.
- [17] SNYDER, A. Encapsulation and inheritance in object-oriented programming languages. *SIGPLAN Not.*, ACM, New York, NY, USA, v. 21, n. 11, p. 38–45, jun. 1986. ISSN 0362-1340.
- [18] ROSSUM, G. van. *Python Language Reference*. version 3.5. [S.l.], 1995–. Disponível em: <<https://www.python.org>>.
- [19] PONTES, J. da R. M.; MANGIAVACCHI, N. *Fenômenos de Transferência*. [S.l.]: SBM, 2010.
- [20] CROWE, C. T. *Multiphase Flows with Droplets and Particles*. [S.l.]: CRC Press, 2011. ISBN 978-1-4398-4050-4,978-1-4398-4051-1.
- [21] ANJOS, G. R. dos. *Hydrodynamics Field Solution of electrochemical Cells Through Finite Element Method*. Tese (Doutorado) — Federal University of Rio de Janeiro, Brazil.
- [22] GEUZAIN, C.; REMACLE, J.-F. Gmsh: A 3-d finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, John Wiley and Sons, v. 79, 2009.
- [23] WALT, S. van der; COLBERT, S. C.; VAROQUAUX, G. The numpy array: A structure for efficient numerical computation. *Computing in Science & Engineering*, AIP, 2011. Disponível em: <<https://www.numpy.org/>>.

- [24] HUNTER, J. D. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, AIP, 2007. Disponível em: <<https://matplotlib.org/>>.
- [25] AHRENS, J.; GEVECI, B.; LAW, C. *ParaView: An End-User Tool for Large Data Visualization*. [S.l.]: Elsevier, 2005. ISBN 978-0123875822.
- [26] JONES, E. et al. *SciPy: Open source scientific tools for Python*. 2001–. Disponível em: <<https://www.scipy.org/>>.