



ANÁLISE NUMÉRICA DO ESCOAMENTO DE RESÍDUOS DE
BIOCOMBUSTÍVEIS EM BANCO DE TUBOS

Juliana Calazans de Cerqueira

Projeto de Graduação apresentado ao Curso de Engenharia Mecânica da Escola Politécnica, Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Engenheiro.

Orientadores: Gustavo Rabello dos Anjos
Rachel Manhães de Lucena

Rio de Janeiro
Setembro de 2022



UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

Departamento de Engenharia Mecânica

DEM/POLI/UFRJ



ANÁLISE NUMÉRICA DO ESCOAMENTO DE RESÍDUOS DE
BIOCOMBUSTÍVEIS EM BANCO DE TUBOS

Juliana Calazans de Cerqueira

PROJETO FINAL SUBMETIDO AO CORPO DOCENTE DO DEPARTAMENTO DE ENGENHARIA MECÂNICA DA ESCOLA POLITÉCNICA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE ENGENHEIRO MECÂNICO.

Aprovada por:

Prof. Gustavo Rabello dos Anjos, Ph.D.

Prof. Rachel Manhães de Lucena, D.Sc.

Prof. Marcelo José Colaço, D.Sc.

Prof. Carolina Palma Naveira Cotta, D.Sc.

RIO DE JANEIRO, RJ – BRASIL

SETEMBRO DE 2022

Calazans de Cerqueira, Juliana

Análise Numérica do Escoamento de Resíduos de Biocombustíveis em Banco de Tubos/ Juliana Calazans de Cerqueira. – Rio de Janeiro: UFRJ/Escola Politécnica, 2022.

XIX, 128 p.: il.; 29,7cm.

Orientadores: Gustavo Rabello dos Anjos

Rachel Manhães de Lucena

Projeto de Graduação – UFRJ/ Escola Politécnica/
Curso de Engenharia Mecânica, 2022.

Referências Bibliográficas: p. 94 – 97.

1. Análise Numérica. 2. Escoamento. 3.
Biocombustíveis. I. Rabello dos Anjos, Gustavo *et al.*
II. Universidade Federal do Rio de Janeiro, UFRJ, Curso
de Engenharia Mecânica. III. Análise Numérica do
Escoamento de Resíduos de Biocombustíveis em Banco de
Tubos.

*“I won’t go as a passenger
Waiting for the road to be laid”
(Sara Bairelles)*

*“It’s time to try defying gravity”
(Stephen Schwartz)*

*Don’t you know that
That patience is a virtue
And life is a waiting game;
Peace must be nurtured
And all the money in the world
can buy you nothing?
(Corine Bailey Rae)*

*“All I know, love will save the
day”
(Des’ree)*

Agradecimentos

É um imenso desafio agradecer a tantos que me ajudaram de alguma forma nesse trajeto em algumas poucas linhas, mas definitivamente não posso começar os agradecimentos sem mencionar meus pais, Max e Cris (mais conhecidos por mim como pai e mãe). Vocês foram implacáveis em me apoiar ao longo de toda a minha jornada. Não apenas acadêmica, mas de vida. Uma frase é muito pouco pra expressar a minha gratidão, mas um livro também não seria suficiente. Então vou me ater ao simples e singelo, mas completo, “muito obrigada por tudo”.

Agradeço também a minha família que esteve comigo ao longo dos anos, torcendo, apoiando e servindo de lembrança que a vida é muito mais feliz e completa com pessoas que amamos ao nosso redor. Obrigada Gustavo, por ser não apenas um irmão, mas servir de exemplo de profissional. Você e meu pai certamente me inspiram e me ajudam a ser do tipo de profissional que desejo ser. Competente, que compartilha conhecimento da maneira mais generosa e genuína e busca não somente crescer, mas também fazer o mesmo com quem os cercam.

Agradeço ao meu companheiro de vida, Adriano. Só nós sabemos tudo que passamos juntos. Imensamente grata pelo companheirismo incansável e imbatível na vida. Obrigada pelo amor, amizade, paciência, compreensão e carinho oferecidos desde que nos conhecemos. Obrigada por ser uma fortaleza na minha vida. Certamente minha caminhada foi mais suave, tranquila e feliz porque você está nela.

Quero registrar também o agradecimento aos amigos que tive o prazer de ter ao longo da vida. Na certeza de esquecer nomes e acabar sendo injusta, não farei menção direta a nenhum deles. Alguns já não tenho contato tão frequente dado as obrigações do dia a dia, mas isso não torna a participação de vocês na minha vida menos importante. Certamente cada um me marcou e me ensinou algo precioso. Mas existem aqueles que permanecem até hoje. Aqueles que apesar das obrigações

ou distâncias se fazem sempre presentes, independente da hora e do tema estiveram e estão sempre disponíveis e me ofertam as coisas mais caras e preciosas das suas vidas: tempo e atenção. Cada conversa me mudou, me fez ver e conhecer um pouco mais do mundo. E por tudo isso sou muito grata.

A graduação seria mais tortuosa sem o apoio entre nós discentes. Agradeço aos membros da equipe Minerva BAJA, principalmente aos membros de 2016 a 2018, com quem tive o prazer de compartilhar o laboratório, os aprendizados, as viagens, as horas não dormidas, as histórias e os encontros. Por causa de vocês o caminho foi menos turbulento e mais engraçado. Agradeço também aos amigos fora da equipe que fiz. Uns de corredor, outros de estudos, alguns pra vida. Obrigada a todos que se prontificaram em ajudar com alguma coisa em algum momento. Tudo isso facilitou meu caminharinho. Obrigada.

Agradeço aos funcionários e corpo docente da UFRJ, principalmente da Engenharia Mecânica, pelos auxílios prestados e conhecimentos transmitidos. Gostaria de agradecer especificamente ao Professor Gustavo Rabello dos Anjos, que aceitou me orientar ao longo dos últimos dois anos neste trabalho e o fez de forma tão generosa e paciente. Aproveito para agradecer também a Rachel Manhães de Lucena pela co-orientação e ao João Paulo Innocente de Souza, que também tiveram grande relevância no desenvolvimento e na qualidade do trabalho desenvolvido ao longo deste tempo e aqui apresentado.

Ainda, gostaria de agradecer a TechnipFMC, principalmente à equipe FTB (*Flexible Technology Brazil*), pela oportunidade de trabalhar com o que gosto, pelo acolhimentos, paciência e aprendizado. Já aprendi e amadureci muito desde que comecei a compartilhar meu cotidiano com vocês e tenho certeza que ainda há muito à aprender. Mas saibam que meu objetivo também é contribuir para que possamos evoluir cada vez mais.

Por fim, agradeço a Agência Nacional de Petróleo (ANP), principalmente ao Programa PRH-ANP 8: Engenharia Mecânica para o Uso Eficiente de Biocombustível, por acreditar na ciência e no seu poder modificador da realidade. Agradeço por acreditar no potencial de uma nova geração e pela bolsa de pesquisa que me foi concedida resultando neste trabalho.

Resumo do Projeto de Graduação apresentado à Escola Politécnica/UFRJ como parte dos requisitos necessários para a obtenção do grau de Engenheiro Mecânico

ANÁLISE NUMÉRICA DO ESCOAMENTO DE RESÍDUOS DE BIOCOMBUSTÍVEIS EM BANCO DE TUBOS

Juliana Calazans de Cerqueira

Setembro/2022

Orientadores: Gustavo Rabello dos Anjos

Rachel Manhães de Lucena

Departamento: Engenharia Mecânica

Nas últimas décadas, os níveis de gás carbônico (CO_2) emitidos na atmosfera têm aumentado, tendo como consequência aumento nas mudanças climáticas. A Bioenergia e Armazenamento de Carbono em formações geológicas dos resíduos da queima e produção de biocombustíveis podem auxiliar na redução dos níveis de CO_2 presentes na atmosfera. O presente trabalho tem como principal objetivo realizar uma análise computacional sobre a hidrodinâmica em meio poroso, utilizando o método de elementos finitos (MEF). Para atingir este objetivo foi desenvolvido um código em linguagem *Python* que viabilizasse a análise numérica da formulação da função corrente-vorticidade em um banco de tubos (meio poroso), tendo como parâmetros a serem analisados a função corrente, vorticidade, velocidade vertical e velocidade horizontal do escoamento. Observou-se que a quantidade, localização e geometria do obstáculo, além do espaço disponível para o desenvolvimento do escoamento, influenciam nos parâmetros analisados.

Palavras-Chave: Análise Numérica, Escoamento, Biocombustíveis.

Abstract of Undergraduate Project presented to POLI/UFRJ as a partial fulfillment of the requirements for the degree of Mechanical Engineer

NUMERICAL ANALYSIS OF BIOFUEL WASTE FLOWS IN TUBE BUNDLE

Juliana Calazans de Cerqueira

September/2022

Advisors: Gustavo Rabello dos Anjos

Rachel Manhães de Lucena

Department: Mechanical Engineering

In the lastest decades, the levels of carbon dioxide (CO_2) emitted into the atmosphere have increased, resulting in an increase in climate change. Bioenergy and Carbon Storage in geological formations of residues from the burning and production of biofuels can help to reduce the levels of CO_2 present in the atmosphere. The main objective of this work is to perform a computational analysis on hydrodynamics in porous media, using the finite element method (FEM). To achieve this goal, a code was developed in *Python* language to enable the numerical analysis of the formulation of the current-vorticity function in a bank of tubes (porous medium), having as parameters the stream function, vorticity, horizontal velocity and vertical velocity. It was observed that the quantity, location and geometry of the obstacle, in addition to the space available for the development of the flow, influence the parameters analyzed.

Keywords: Numerical Analisis, Flow, Biofuels.

Sumário

Lista de Figuras	xii
Lista de Tabelas	xv
Lista de Símbolos	xvi
Lista de Abreviaturas	xix
1 Introdução	1
1.1 Motivação	1
1.2 Objetivo	3
1.3 Organização do trabalho	4
2 Revisão de Conceitos Básicos	5
2.1 Biocombustíveis	5
2.2 Captura e Armazenamento de Carbono	6
2.2.1 Armazenamento em aquíferos salinos	7
2.2.2 Potencial do CCS no Brasil	8
2.3 Mecânica dos Fluidos	8
2.3.1 Leis Governantes	11
2.4 Método de Elementos Finitos - MEF	16
3 Metodologia	21
3.1 Modelo Teórico	21
3.1.1 Formulação Função Corrente-Vorticidade	21
3.2 Método Numérico	23
3.2.1 O MEF na Função Corrente Vorticidade	23

3.3	Gerador de Malhas	29
3.4	Implementação do código	29
3.5	Visualização dos resultados	29
4	Verificação do Código	31
4.1	Escoamento de <i>Lid Driven</i>	31
4.2	Escoamento entre placas planas	42
5	Análise do escoamento sobre os obstáculos	49
5.1	Escoamento ao redor de um obstáculo	51
5.1.1	Caso 1 - obstáculo circular em domínio retangular	51
5.1.2	Caso 2 - obstáculo circular em domínio quadrada	54
5.1.3	Caso 3 - Influência do passo de tempo	56
5.1.4	Caso 4 - obstáculo elíptico com maior eixo na horizontal	59
5.1.5	Caso 5 - obstáculo elíptico com maior eixo na vertical	60
5.2	Escoamento ao redor de dois alinhados horizontalmente	62
5.2.1	Caso 6 - Dois obstáculos circulares	63
5.2.2	Casos 7 e 8 - Influência da distância entre obstáculos	65
5.2.3	Caso 9 - Dois obstáculos elípticos com maior eixo na horizontal	69
5.2.4	Caso 10 - Dois obstáculos elípticos com maior eixo na vertical	70
5.3	Escoamento em meio poroso - 2x2	72
5.3.1	Caso 11 - obstáculos circulares	74
5.3.2	Caso 12 - obstáculos elípticos com maior eixo na horizontal	75
5.3.3	Caso 13 - obstáculos elípticos com maior eixo na vertical	76
5.4	Escoamento em meio poroso - 3x3	79
5.4.1	Caso 14 - obstáculos circulares	79
5.4.2	Caso 15 - obstáculos elípticos com maior eixo na horizontal	81
5.4.3	Caso 16 - obstáculos elípticos com maior eixo na vertical	82
5.5	Escoamento em meio poroso - 10x3	85
5.5.1	Caso 17 - obstáculos circulares	85
5.5.2	Caso 18 - obstáculos elípticos com maior eixo na horizontal	87
5.5.3	Caso 19 - obstáculos elípticos com maior eixo na vertical	89
6	Conclusões	92

Referências Bibliográficas	94
A Códigos Fonte	98
A.1 Script para geração das malhas no GMSH	98
A.2 Script dos casos de verificação	100
A.2.1 Escoamento de <i>Lid Driven</i>	100
A.2.2 Escoamento entre placas planas	109
A.3 Script dos casos de estudo	118

Lista de Figuras

2.1	Classificação do escoamento dos fluidos.	10
2.2	Tipos de movimento ou deformação de um fluido.	12
2.3	Exemplo de discretização de um domínio em sub-regiões por MEF.	16
2.4	Tipos de elementos.	18
2.5	Exemplo de uma malha estruturada e respectiva IEN.	19
2.6	Exemplo de malha com elementos triangulares de diferentes tamanhos.	20
4.1	Condições de contorno do escoamento na cavidade <i>Lid Driven</i>	31
4.2	Solução numérica do Teste 1.	33
4.3	Solução numérica do Teste 2.	35
4.4	Solução numérica do Teste 3.	36
4.5	Solução numérica do Teste 4.	37
4.6	Solução numérica do Teste 5.	38
4.7	Eixos selecionados para a avaliação dos valores na cavidade de <i>Lid Driven</i>	39
4.8	Comparação das soluções numéricas entres os Testes 1 a 5 e a literatura.	39
4.9	Solução numérica do Teste 6.	41
4.10	Comparação das soluções numéricas entres os Testes 6 e a literatura.	42
4.11	Condições de contorno do escoamento Entre Placas Planas.	43
4.12	Eixo selecionado para a avaliação dos valores do escoamento entre Placas Planas.	43
4.13	Solução numérica do Teste 8.	45
4.14	Solução numérica do Teste 10.	46
4.15	Comparação das soluções numéricas do perfil de Velocidade entres os Testes 7 a 10 e a literatura.	47

4.16	Comparação das soluções numéricas da Função Corrente entres os Testes 7 a 10 e a literatura.	47
4.17	Comparação das soluções numéricas da Vorticidade entres os Testes 7 a 10 e a literatura.	48
5.1	Desenho esquemático com condições de contorno.	51
5.2	Malha do domínio retangular.	52
5.3	Solução numérica do Caso 1.	53
5.4	Malha do domínio quadrado.	54
5.5	Solução numérica do Caso 2.	55
5.6	Comparativo dos resultados do domínio retangular e quadrado.	55
5.7	Solução numérica do Caso 3.	56
5.8	Comparação dos resultados das variáveis a jusante do obstáculo, de acordo com o passo de tempo	57
5.9	Comparação dos resultados das variáveis a montante do obstáculo, de acordo com o passo de tempo	58
5.10	Solução numérica do Caso 4.	60
5.11	Solução numérica do Caso 5.	61
5.12	Comparação das soluções numéricas entre os Casos 2, 4 e 5.	62
5.13	Desenho esquemático com condições de contorno com dois obstáculos alinhados horizontalmente.	63
5.14	Solução numérica do Caso 6.	64
5.15	Solução numérica da Função Corrente dos Caso 7 e 8.	65
5.16	Solução numérica da Vorticidade dos Casos 7 e 8.	66
5.17	Solução numérica da Velocidade Horizontal dos Casos 7 e 8.	67
5.18	Solução numérica da Velocidade Vertical dos Casos 7 e 8.	68
5.19	Solução numérica do Caso 9.	69
5.20	Solução numérica do Caso 10	70
5.21	Comparação das soluções numéricas entre os Casos 6, 9 e 10.	71
5.22	Desenho esquemático com condições de contorno com quatro obstáculos, organizados dois a dois.	73
5.23	Solução numérica do Caso 11.	75
5.24	Solução numérica do Caso 12.	76

5.25	Solução numérica do Caso 13.	77
5.26	Comparação das soluções numéricas entre os Casos 11, 12 e 13, em x constante.	78
5.27	Comparação da solução numérica entre os Caso 11, 12 e 13, em y constante.	79
5.28	Solução numérica do Caso 14.	80
5.29	Solução numérica do Caso 15.	81
5.30	Solução numérica do Caso 16.	82
5.31	Comparação da solução numérica entre os Caso 14, 15 e 16, em x constante.	83
5.32	Comparação da solução numérica entre os Casos 14, 15 e 16, em y constante.	84
5.33	Desenho esquemático com condições de contorno com trinta obstáculos obstáculos, organizados em 10×3	85
5.34	Malha de domínio retangular com trinta obstáculos.	86
5.35	Solução numérica do Caso 17.	87
5.36	Solução numérica do Caso 18.	88
5.37	Solução numérica do Caso 19.	90

Lista de Tabelas

4.1	Parâmetros dos testes com diferentes malhas, passo de tempo e intervalos de tempo, para o caso da cavidade de <i>Lid Driven</i> e com $Re = 10$.	32
4.2	Parâmetros dos testes com diferentes malhas, passo de tempo e intervalos de tempo, para o caso de escoamento entre placas planas e com $Re = 10$.	44
5.1	Tabela resumo dos casos estudados	50
5.2	Parâmetros dos casos com diferentes malhas, passo de tempo, posições dos centros e permeabilidades horizontais para o caso de dois obstáculos circulares em sequência.	64
5.3	Parâmetros dos três casos com quatro obstáculos testados em uma domínio quadrado de 1×1 ($x = 0.1$ e $y = 0.1$).	73

Lista de Símbolos

D	derivada material, p. 12
L	comprimento característico, p. 9
M	número de Mach, p. 11
N	função de forma, p. 24
Re	número de Reynolds, p. 9
V	volume, p. 9
Δt	intervalo de tempo, p. 20
Γ	região do contorno, p. 24
Ω	domínio, p. 23
γ	peso específico, p. 9
\int	integral, p. 23
\mathbf{G}	matriz do gradiente, p. 25
\mathbf{K}	matriz de rigidez, p. 25
\mathbf{M}	matriz de massa, p. 25
\mathbf{a}	vetor aceleração, p. 12
\mathbf{v}	vetor velocidade, p. 12
μ	viscosidade dinâmica, p. 9
$\nabla \times$	operador rotacional, p. 22

∇	operador nabla, p. 13
ν	viscosidade cinemática, p. 9
ω	vorticidade, p. 22
ω_a	velocidade angular, p. 13
ω_z	vorticidade em z, p. 22
∂	derivada parcial, p. 12
ψ	função corrente, p. 15
ρ	massa específica, p. 9
σ	tensão normal, p. 15
\sum_e	somatório de elementos, p. 24
τ	tensão cisalhante, p. 15
τ_{yx}	tensão de cisalhamento, p. 9
ξ	vetor vorticidade, p. 13
$\{F\}$	matriz coluna de cargas nodais, p. 19
$\{d\}$	matriz coluna de deslocamentos nodais, p. 19
a_x	aceleração em x, p. 12
a_y	aceleração em y, p. 12
a_z	aceleração em z, p. 12
c	velocidade do som, p. 11
g	gravidade, p. 9
g_x	gravidade em x, p. 15
g_y	gravidade em y, p. 15
g_z	gravidade em z, p. 15

k	módulo de elasticidade, p. 9
m	massa, p. 9
n	número de nós, p. 18
n_{gl}	número de graus de liberdade, p. 18
p	pressão escalar, p. 21
t	tempo, p. 12
v	velocidade do escoamento, p. 11
v_x	velocidade em x, p. 12
v_y	velocidade em y, p. 12
v_z	velocidade em z, p. 12
w	função peso, p. 23
K	matriz de rigidez, p. 19

Lista de Abreviaturas

BECCS	<i>Biomass Energy with Carbon Capture and Storage</i> , p. 2
Bio-CCS	<i>Bioenergia e Armazenamento de Carbono</i> , p. 2
CCS	<i>Carbon Capture and Storage</i> , p. 2
CDR	<i>Carbon Dioxide Removal</i> , p. 2
CH ₄	Metano, p. 6
CO ₂	Dióxido de carbono, p. 1
CO	Monóxido de carbono, p. 1
GEE	Gases de Efeito Estufa, p. 5
GtCO ₂	Giga toneladas de dióxido de carbono, p. 2
HC	Hidrocarbonetos voláteis, p. 1
IEN	Matriz de Conectividade, p. 18
MEF	Método de elementos finitos, p. 4
NO _x	Óxidos de nitrogênio, p. 1
N ₂	Nitrogênio, p. 6
O ₂	Oxigênio, p. 6
PNMC	Política Nacional sobre Mudança do Clima, p. 8
SO _x	Óxidos de enxofre, p. 1

Capítulo 1

Introdução

1.1 Motivação

A temperatura média da Terra aumentou em cerca de 0,76 °C nos últimos 150 anos. Apesar de um efeito causal entre o aumento da temperatura global e o aumento da concentração de gases de efeito estufa na atmosfera ainda não ter sido comprovado, estudos indicam alta correlação entre estes dois fatores [1, 2].

Os combustíveis, amplamente utilizados pelo mundo para geração de energia, tanto de origem fóssil como biocombustíveis são um dos principais responsáveis pela emissão desses gases provocadores do efeito estufa. Ao longo do ciclo de vida de combustíveis e biocombustíveis ocorre a emissão de diversos gases de efeito estufa, como por exemplo, monóxido de carbono (CO), dióxido de carbono (CO₂), metano (CH₄), óxidos de enxofre (SO_x), óxidos de nitrogênio (NO_x) e hidrocarbonetos voláteis (HC). Destes, o CO₂ é o gás que apresenta maior volume de emissões ao longo de todo o ciclo de vida de um combustível, seja combustível de origem fóssil, seja biocombustível [3].

Desde a era pré industrial até 2019, a emissão cumulativa de CO₂ no planeta foi mais de 2000 GtCO₂ (2400 ± 240 GtCO₂); das quais 62% foram acumulados desde 1970. Na última década (de 2010 a 2019) as emissões de carbono na atmosfera foram quase na mesma quantidade das reservas de carbono remanescentes disponíveis para limitar aumento da temperatura global em 1,5 °C e cerca de 1/3 para limitar o aumento em 2°C da temperatura global. O aumento da temperatura global de 1,5 °C já pode impactar sistemas de biodiversidade únicos devido aos extremos climáticos.

Os riscos aumentam conforme elevação da temperatura global aumenta, e quando esse aumento ultrapassa 3 °C, efeitos mais extremos em termos de disrupção poderão ser observados em grande escala [4, 1, 2, 5].

Na última década, dez países juntos foram responsáveis por 75% dos 6,5 GtCO₂/ano emitidos, dentre eles, encontra-se o Brasil. Tendo em vista este cenário, o aquecimento global tem sido um dos maiores desafios e alvo de amplos estudos nas últimas décadas, e diversas nações têm reunido esforços para mitigar as mudanças climáticas. No final do ano de 2016 entrou em vigor Acordo de Paris [6], o qual tem como principal objetivo a limitação do aumento da temperatura global para menos de 2 °C acima dos níveis pré-industriais [4, 6].

Esta pode ser considerada uma meta ousada e requer não apenas uma rápida transição para geração de energia a baixo custo de carbono e consequentemente a redução da emissão de CO₂, como a redução da concentração do gás. Sendo assim, é necessário que haja uma emissão negativa de CO₂ para que tal meta seja alcançada. Isto porque estudos estimam que, mesmo assumindo as políticas climáticas mais ambiciosas, cerca de 600-700 GtCO₂ de combustível fóssil emitidos foram acumulados no planeta no século XXI [4].

Fontes alternativas de energia tais quais eólica, solar e de biocombustíveis são consideradas fontes “neutras” pois resultam em emissões muito baixas de gases de efeito estufa. Vários projetos de produção de biocombustíveis em escala comercial estão avançados em muitas partes do mundo, abrangendo uma ampla seleção de tecnologias e opções de matérias-primas, incluindo captura e sequestro de carbono (CCS, do inglês: *Carbon Capture and Storage*) que apoia a remoção de dióxido de carbono (CDR, do inglês: *Carbon Dioxide Removal*)[4, 7].

Uma tecnologia promissora para alcançar a redução de CO₂ na atmosfera é a Bioenergia e Armazenamento de Carbono (Bio-CCS ou BECCS, do inglês: *Biomass Energy/Bioenergy with Carbon Capture and Storage*). Projetos de BECCS são baseados no conceito de capturar e posteriormente armazenar o CO₂ proveniente da queima ou processamento dos biocombustíveis, resultando em emissão negativa de CO₂. Desta forma, esta tecnologia pode ser potencialmente aplicada a uma ampla gama de usinas de energia, incluindo usinas de 100% biomassa, instalações de produção de biocombustíveis ou biorrefinarias. [1, 2, 7].

De acordo com o relatório de Alamo et. al. [8], se o CO₂ liberado por processos de base biológica for capturado e armazenado, por exemplo, em formações geológicas (ou seja, se a CCS for aplicada em combinação com processos de bioenergia), as emissões negativas de CO₂ podem ser potencialmente alcançadas. Além disso, a produção atual de biocombustíveis sofre desafios. Uma das principais preocupações relacionadas à atual produção de biocombustível está atrelada ao fornecimento de biomassa, incluindo a competição entre alimentos e energia e uso da terra [8].

O acoplamento da produção de biocombustíveis à infraestrutura de CCS tem uma vantagem relacionada ao potencial de obtenção de emissões negativas de CO₂, melhorando assim a sustentabilidade do uso de biocombustíveis avançados. Além disso, a Bio-CCS aplicada ao biocombustível oferece uma vantagem potencial adicional em relação a Bio-CCS aplicada a usinas (convencionais): a implementação da CCS é relativamente mais fácil devido ao fato de que a unidade de captura CO₂ pode já estar incluída no processo de produção de biocombustível como parte do processo de *upgrade* do gás para garantir que o gás de alimentação para a unidade de síntese (e, conseqüentemente, o produto final do biocombustível) atenda aos requisitos de qualidade (razão H₂/CO). Portanto, a integração da planta de biocombustível à infraestrutura CCS se limita à adição de condicionamento de CO₂ [8].

De acordo com Machado [7], no Brasil a BECCS faz-se relevante devido principalmente à dois fatores. O primeiro é devido a significativa produção de etanol no país a partir de biomassa e o segundo dá-se devido a extensa ocorrência de bacias sedimentares com reservatórios potenciais para armazenamento [7].

1.2 Objetivo

Considerando a relevância dos estudos teóricos e numéricos acerca da Bio-CCS, o presente trabalho tem como principal objetivo realizar uma análise computacional sobre a hidrodinâmica em meio poroso, utilizando o método de elementos finitos (MEF). A fim de alcançar o objetivo principal, este trabalho tem como objetivos específicos: a) estudar a influência de um obstáculo no escoamento; b) verificar a influência do espaço para desenvolvimento do escoamento antes do mesmo encontrar um obstáculo; c) estudar a influência da geometria dos obstáculos no escoamento;

d) estudar a influência do número e disposição dos obstáculos presentes em um meio poroso no desenvolvimento do escoamento.

1.3 Organização do trabalho

Este trabalho encontra-se dividido em capítulos, de forma que o presente capítulo (capítulo 1) faz uma breve introdução do trabalho, com apresentação da motivação e objetivo deste trabalho. Em sequência, no capítulo 2, é apresentada uma breve revisão dos conceitos básicos a respeito de biocombustíveis, Captura e Armazenamento de Carbono e Mecânica dos Fluidos. O capítulo 3 versa sobre a metodologia adotada, expondo o modelo teórico e o modelo numérico, bem como sobre os softwares utilizados e a implementação do código. No capítulo 4 são expostas a verificação do código desenvolvido. Os resultados e discussões acerca do trabalho desenvolvido podem ser visualizados no capítulo 5. E o capítulo 6 apresenta as conclusões. Além disso, o código desenvolvido em *Python* para a realização deste trabalho pode ser visualizado no apêndice A.

Capítulo 2

Revisão de Conceitos Básicos

2.1 Biocombustíveis

Biocombustíveis são combustíveis produzidos a partir de recursos orgânicos (biomassa) que podem ser cultivados inúmeras vezes, diferentemente dos combustíveis fósseis que são formados a partir de processos geológicos. Os combustíveis fósseis têm sido utilizados em grande escala na história recente e seu uso tem sido fortemente atrelado ao aumento da temperatura global em razão de emissão de gases poluentes para a atmosfera, conhecido como Gases de Efeito Estufa (GEE), dentre os quais o de maior relevância é o CO_2 [9, 10, 4].

Em razão do crescimento da demanda de combustíveis nos últimos tempos e os combustíveis fósseis serem considerados não renováveis e altamente poluentes, o estudo e uso de biocombustíveis vem crescendo ao longo das últimas décadas. Isto porque os biocombustíveis são compreendidos como uma fonte de energia renovável e de caminho neutro de carbono (uma vez que sua matéria-prima, a biomassa, absorve a maior parte de CO_2 liberado para o meio ambiente) e biodegradável [10, 11, 12, 9].

Com o avançar dos anos, a tecnologia de geração de biocombustíveis também avançou e atualmente os biocombustíveis são classificados em 4 gerações de tecnologia. A primeira geração inclui biodiesel e bioetanol produzidos a partir de plantações de alimentos de alto teor de açúcar ou plantas com grande quantidade de óleos, como por exemplo, cana-de-açúcar, milho, semente de girassol e soja. A primeira geração de biocombustíveis trouxe consigo um debate sobre concorrência entre alimentação e energia. A segunda geração refere-se aos biocombustíveis produzidos principal-

mente a partir de materiais lignocelulósicos e resíduos agrícolas e florestais (como por exemplo, madeira, palha e pinhão). Esta geração minimizou a “insegurança alimentar”, uma vez que sua matéria-prima não concorre com a alimentação da população. A terceira tem como matéria-prima algas produzidas em larga escala e de refino relativamente fácil. E a quarta geração se refere aos biocombustíveis produzidos a partir de organismos modificados, com maior poder de absorção de CO_2 (cianobactérias microalgas, fungos e leveduras) [10, 9, 13].

2.2 Captura e Armazenamento de Carbono

No início dos anos de 1880 surgiram os primeiros indicativos que dos componentes atmosféricos, o nitrogênio (N_2) e o oxigênio (O_2) (que correspondem as componentes que tem maior percentual na composição do ar atmosférico) não tem influência no efeito estufa. Contudo, componentes como dióxido de carbono (CO_2) e metano (CH_4), que estão em menores proporções, têm grande relação com o efeito estufa. Apesar de não ter uma relação de causalidade entre CO_2 e o efeito estufa comprovada, é de grande consenso que não deve-se esperar esta comprovação para começar a atuar na redução dos níveis de CO_2 , ou seja, na redução desse gás da atmosfera [14, 1, 2].

O CCS é um processo que consiste em separar e capturar CO_2 de fontes estacionárias (por exemplo, refinarias) e transportá-lo para reservatórios de modo a isolá-lo da atmosfera por longos períodos (na ordem de centenas, milhares ou milhões de anos) [1]. Com o aumento de CO_2 antropogênico na atmosfera, o CCS passa a ser entendido como uma tecnologia promissora na redução de GEE. Se o CCS aplicado já tem grande impacto na redução das concentrações de GEE, essa tecnologia atrelada à produção de biocombustíveis (BECCS) possui ainda maior relevância ao possibilitar uma produção negativa de CO_2 . No caso da produção de etanol baseado em cana-de-açúcar, por exemplo, no processo de fermentação ocorre uma liberação de CO_2 puro permitindo que, caso o mesmo seja capturado, as emissões de CO_2 sejam negativas; ou seja, a emissão de CO_2 pode ficar abaixo do nível “zero” uma vez que haverá mais consumo do que produção deste gás [1, 15, 4, 16].

As três formas que são comumente consideradas para o armazenamento são: o

geológico, nos oceanos e na superfície de carbonatação mineral [1].

2.2.1 Armazenamento em aquíferos salinos

No caso do armazenamento nos oceanos o processo consiste em injetar o CO_2 em grandes profundidades, onde será possível que ele se dissolva e forme hidratos ou compostos mais pesados que a água do mar, se depositando no fundo dos oceanos [1]. Antigos reservatórios de óleo e aquíferos salinos são reservatórios naturais de CO_2 em potencial. No caso de aquíferos salinos, em profundidades abaixo de 800m o CO_2 já encontra-se em estado líquido, e o seu escoamento depende da densidade e viscosidade do CO_2 líquido e da substância presente no aquífero, ou seja, dependem da temperatura e pressão a qual estão submetidos e a salinidade da água. Além disso, sob determinadas condições o CO_2 pode ser parcialmente dissolvido na salmoura [14].

Devido a diferença de densidade, o CO_2 pode ter uma tendência a subir nas camadas rochosas após ser injetado. Por isso, é importante que essa injeção ocorra abaixo de rochas de baixa permeabilidade pois estas funcionam como uma espécie de tampa do reservatório. Entretanto, o acúmulo de CO_2 nos limites de rochas porosas e rochas de baixa permeabilidade pode ocasionar uma fratura nesta rocha que está funcionando como tampa do reservatório, e conseqüentemente fuga de CO_2 do reservatório [14]. A adição de CO_2 nas águas do mar pode alterar a composição química da mesma e afetar diretamente a vida marinha [1].

Huppert e Neufeld [14] descrevem o mecanismo de aprisionamento de CO_2 nos aquíferos salinos. Segundo eles, a diferença de concentração de CO_2 e a salmoura podem resultar em uma corrente convectiva, o que em alguns aquíferos pode levar a um aumento de fluxo de CO_2 dissolvido. A interação de forças entre o CO_2 e a salmoura pode resultar na formação de pequenos conglomerados (“gânglios”) dentro da rocha resultando em um processo de aprisionamento residual ou capilar. Além disso, após muito tempo, o CO_2 pode se transformar em carbonato, precipitando e resultando em um aprisionamento estável.

Já Bachu [1] fala de dois mecanismos principais de aprisionamento, o físico e o químico. O aprisionamento físico pode ocorrer devido a uma barreira de baixa permeabilidade que impede o escoamento do CO_2 ou devido a uma tensão interfacial entre o CO_2 e água, que impossibilita a movimentação do CO_2 , mesmo que haja um

caminho livre para escoamento. Já no aprisionamento químico o CO_2 é adsorvido em material orgânico (por exemplo, carvão) ou é dissolvido em outros fluidos, gerando uma substância que, em última instância pode gerar uma reação química com a rocha [1].

2.2.2 Potencial do CCS no Brasil

O Brasil atualmente é um beneficiário do mercado de crédito de carbono, 80% da sua produção de energia vem de fontes renováveis e em 2009 foi estabelecida a Lei que instituiu a Política Nacional sobre Mudança do Clima - PNMC (Lei 12.187) que tinha como meta a redução de CO_2 a quase a metade dos níveis de emissão registrada pelo país até 2020; o que corresponde a cerca de 1 Gt CO_2 . Apesar das emissões de GEE no Brasil serem relativamente baixas, o CCS pode se tornar cada vez mais importante para o país alcançar sua meta. Isto porque o setor petrolífero brasileiro foi fortemente impactado após a descoberta de petróleo no pré-sal, um petróleo que tem como característica alta concentração de CO_2 e conseqüentemente aumenta significativamente a emissão do mesmo na atmosfera [17, 18, 15, 19].

Tendo em vista este fato, desde meados dos anos 2000 o país tem desenvolvido pesquisas focadas na tecnologia de armazenamento de carbono em solo brasileiro e potenciais reservatórios de CO_2 . Estimativa preliminar aponta da capacidade teórica de armazenamento de CO_2 em mais de 2000 Gt CO_2 em reservatórios geológicos brasileiros [17, 15]. MOREIRA *et. al.* [15] aponta o potencial de utilização de aquíferos salinos brasileiros como o Tubarão (que encontra-se abaixo do aquífero Guarani) para estocagem de CO_2 , mas ressalta que ainda são poucos os estudos relacionados de rochas porosas sendo utilizadas para este fim.

2.3 Mecânica dos Fluidos

De acordo com Fox e McDonald [20], “a mecânica dos fluidos é o estudo de fluidos em repouso ou em movimento”. Apesar de pequena, esta definição abarca um amplo, e por muitas vezes complexo, campo de estudos. O estudo da mecânica dos fluidos envolve questões desde movimento de rios e mares, passando por aerodinâmica de carros e aviões, fenômenos atmosféricos até a circulação do sangue nas artérias de

um ser humano [20, 21].

Ainda de acordo com Fox e McDonald [20], os fluidos são substâncias que a menor aplicação de uma tensão tangencial a uma substância a deforma continuamente. Sendo assim, líquidos e gases são formas nas quais os fluidos se apresentam. Estas substâncias são entendidas como suaves, contínuas e esse conceito de continuidade é base da mecânica dos fluidos clássica. Desta forma, as propriedades do fluido são consideradas contínuas [20].

A massa específica é definida como a quantidade de matéria por unidade de volume, usualmente expressa por:

$$\rho = \frac{\delta m}{\delta V} \quad [kg/m^3], \quad (2.1)$$

onde ρ é a massa específica; δm é a variação de massa e δV a variação de volume. A partir da relação com a aceleração da gravidade (g), pode-se definir esta propriedade de a partir da força peso exercida por unidade de volume, sendo esta forma conhecida como peso específico (γ).

$$\gamma = \frac{\delta m g}{\delta V} = \rho g \quad [N/m^3]. \quad (2.2)$$

A viscosidade dinâmica (μ) é outra importante característica de um fluido. Ela é equivalente ao módulo de elasticidade (k) de substâncias sólidas, relacionando a tensão de cisalhamento (τ_{yx}) de um fluido a sua taxa de deformação (du/dy), ou seja:

$$\tau_{yx} = \mu \frac{du}{dy} \quad [Pa.s]. \quad (2.3)$$

É comum relacionar a massa específica com a viscosidade dinâmica. Esta relação é conhecida como viscosidade cinemática (ν) e é dada por:

$$\nu = \frac{\mu}{\rho} \quad [m^2/s]. \quad (2.4)$$

O escoamento de um fluido é comumente caracterizado a partir do número de Reynolds, sendo este um número adimensional que relaciona as forças inerciais e viscosas a partir da equação

$$Re = \rho \frac{\nu L}{\mu}, \quad (2.5)$$

na qual L é o comprimento característico. Logo, a partir da equação 2.5 pode-se notar que a influência das forças viscosas é mais relevante para escoamentos com um

baixo número de Reynolds, e que com o aumento deste número, as forças viscosas vão ficando menos relevantes.

Os escoamentos dos fluidos podem ser classificados de acordo com alguns critérios [20], conforme pode ser visualizado na Figura 2.1.

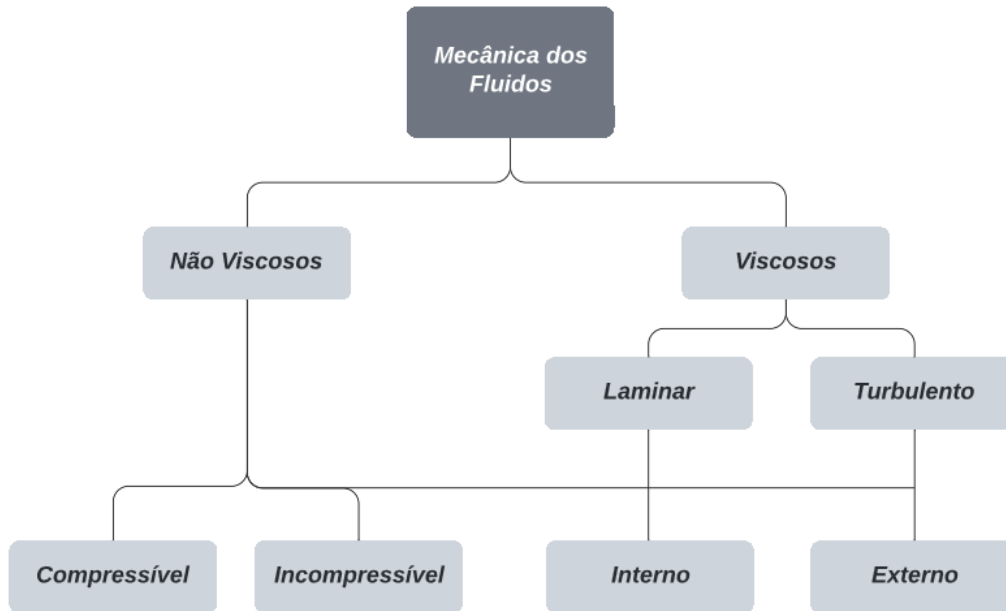


Figura 2.1: Classificação do escoamento dos fluidos. Fonte: Própria.

O escoamentos não viscosos (ou invíscidos) são aqueles os quais os efeitos da viscosidade são desconsiderados ($\mu = 0$), já nos escoamentos viscosos a influência da viscosidade é considerada. Neste caso, ele pode ainda ser dividido entre fluidos newtonianos (os quais a tensão cisalhante é diretamente proporcional ao gradiente das velocidades (equação 2.3) em todas as direções) e fluidos não newtonianos (os quais a taxa de deformação não é diretamente proporcional à tensão de cisalhamento). Neste caso, outras equações são necessárias para descrever o comportamento deste tipo de fluido. Um exemplo de fluido newtoniano é a água, e um exemplo de fluido não newtoniano é a pasta de dente [20, 21].

A compressibilidade ou incompressibilidade versa sobre a densidade do fluido durante o escoamento. Densidades contantes ou com baixa variação ao longo do escoamentos são considerados incompressíveis, caso contrário, como compressíveis. A maneira mais usual de determinar se um fluido é compressível ou incompressível é de acordo com o número de Mach. O número de Mach relaciona a velocidade do

fluido com a velocidade do som, de modo que

$$M = \frac{v}{c}, \quad (2.6)$$

onde v é a velocidade do escoamento e c é a velocidade do som. Os fluidos são considerados incompressíveis quando o número de Mach é menor que 0,3 [20, 21].

No escoamento laminar, as camadas do fluido escoam adjacentes umas das outras, sem se misturarem, já no escoamento turbulento, essas camadas interagem umas com as outras e geram movimentos aleatórios. A classificação entre laminar e turbulento pode ser determinada a partir do número de Reynolds. O ponto exato que um escoamento deixa de ser laminar e torna-se turbulento não é bem definido, mas baixos números de Reynolds estão relacionados a escoamentos laminares, e altos valores de Reynolds indicam escoamentos turbulentos [20, 21].

O escoamento pode ainda ser classificado com interno ou externo. Os internos são aqueles que ocorrem dentro de dutos e outras superfícies (dentro de uma mangueira, por exemplo) e quando o fluido escoar livremente sem confinamento este é classificado como escoamento externo (como ocorre com a água que sai de uma torneira, por exemplo) [20, 21].

Além dessas classificações, é usual identificar um escoamento de acordo com o número de coordenadas necessárias para descrevê-lo. Os escoamentos reais são tridimensionais (ou seja, necessitam de 3 posições de referência dois eixos ortogonais para que sejam descritos completamente); contudo em algumas situações o escoamento pode ser simplificado de modo a ser representado por coordenada unidimensional ou bidimensional, sem que haja perda de informação (como por exemplo em escoamentos uniformes dentro de um duto) [20, 21].

2.3.1 Leis Governantes

O comportamento de um fluido pode ser analisado e descrito matematicamente através de algumas leis governantes. Ao longo de um escoamento, um elemento de fluido está sujeito a diferentes tipos de movimentações e deformações, como translação, rotação, deformação linear e angular (Figura 2.2).

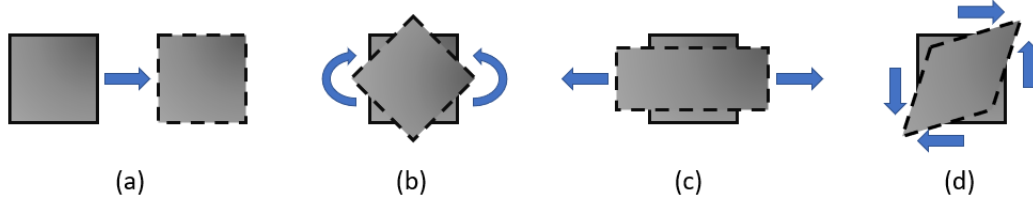


Figura 2.2: Tipos de movimento ou deformação de um fluido: (a) translação, (b) rotação, (c) deformação linear e (d) deformação por cisalhamento. Fonte: Própria.

Uma partícula fluida translada ao longo de um escoamento, para que seja possível caracterizar este comportamento, faz-se necessário a descrição matemática do campo de velocidades e aceleração do elemento. O vetor velocidade, pode ser escrito como:

$$\mathbf{v} = v_x \mathbf{i} + v_y \mathbf{j} + v_z \mathbf{k}. \quad (2.7)$$

De modo que \mathbf{v} represente o vetor de velocidades e v_x , v_y e v_z representem as componentes escalares deste vetor nos eixos cartesianos x , y , e z , respectivamente. Além disto, esta é uma função dependente do tempo, de modo que ao analisar a taxa de variação da velocidade no tempo, pode-se descrever a aceleração desta partícula através da equação:

$$\mathbf{a} = \frac{d\mathbf{v}}{dt} = \left(v_x \frac{\partial \mathbf{v}}{\partial x} + v_y \frac{\partial \mathbf{v}}{\partial y} + v_z \frac{\partial \mathbf{v}}{\partial z} \right) + \frac{\partial \mathbf{v}}{\partial t}. \quad (2.8)$$

O termo entre parênteses é a aceleração convectiva e a parcela $\frac{\partial \mathbf{v}}{\partial t}$ é a aceleração local. Como a aceleração também é uma grandeza vetorial, a equação 2.8 pode ser expressa a partir de suas componentes escalares, dadas pelas equações 2.9 a 2.11.

$$a_x = \left(v_x \frac{\partial v_x}{\partial x} + v_y \frac{\partial v_x}{\partial y} + v_z \frac{\partial v_x}{\partial z} \right) + \frac{\partial v_x}{\partial t}, \quad (2.9)$$

$$a_y = \left(v_x \frac{\partial v_y}{\partial x} + v_y \frac{\partial v_y}{\partial y} + v_z \frac{\partial v_y}{\partial z} \right) + \frac{\partial v_y}{\partial t}, \quad (2.10)$$

$$a_z = \left(v_x \frac{\partial v_z}{\partial x} + v_y \frac{\partial v_z}{\partial y} + v_z \frac{\partial v_z}{\partial z} \right) + \frac{\partial v_z}{\partial t}. \quad (2.11)$$

É comum as equações acima também serem encontradas escritas a partir da utilização de um operador chamado derivada material (D/Dt , equação 2.12).

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + (\mathbf{v} \cdot \nabla). \quad (2.12)$$

O primeiro termo à direita da igualdade é composta por uma parte local e o segundo termo é uma parte convectiva ou advectiva. Desta forma, a aceleração também é comumente expressa por:

$$\mathbf{a} = \frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v}. \quad (2.13)$$

Além de transladar, as partículas fluidas podem girar em torno do seu próprio eixo. A velocidade angular ($\omega_{\mathbf{a}}$) indica a taxa de rotação e é expressa em coordenadas cartesianas como:

$$\omega_{\mathbf{a}} = \omega_{a_x} \mathbf{i} + \omega_{a_y} \mathbf{j} + \omega_{a_z} \mathbf{k}. \quad (2.14)$$

Conforme Çengel e Cimbala [21], a taxa de rotação para um eixo é a “taxa de rotação média de duas retas inicialmente perpendiculares que se cruzam nesse ponto”. Deste modo a velocidade angular no eixo z (ω_{a_z}) pode ser escrita como:

$$\omega_{a_z} = \frac{1}{2} \left(\frac{\partial v_y}{\partial x} - \frac{\partial v_x}{\partial y} \right). \quad (2.15)$$

Considerando que a velocidade angular nos outros eixos pode ser escrita de modo análogo, a velocidade angular pode ser reescrita como:

$$\omega_{\mathbf{a}} = \frac{1}{2} \left[\left(\frac{\partial v_z}{\partial y} - \frac{\partial v_y}{\partial z} \right) i + \left(\frac{\partial v_x}{\partial z} - \frac{\partial v_z}{\partial x} \right) j + \left(\frac{\partial v_y}{\partial x} - \frac{\partial v_x}{\partial y} \right) k \right]. \quad (2.16)$$

Note que $\omega_{\mathbf{a}}$ é a metade do produto vetorial $\nabla \times \mathbf{v}$ (conhecido como rotacional de \mathbf{v}). Então:

$$2\omega_{\mathbf{a}} = \nabla \times \mathbf{v}. \quad (2.17)$$

Sendo o rotacional de $\omega_{\mathbf{a}}$ conhecido como vetor **vorticidade**, o mesmo pode ser definido, também como:

$$\xi = 2\omega_{\mathbf{a}} = \nabla \times \mathbf{v}. \quad (2.18)$$

Esta equação é utilizada para classificar escoamentos entre rotacionais e irrotacionais. Caso um escoamento seja irrotacional, as partículas de um fluido movimentam-se mas não giram entre si, ou seja, elas não possuem velocidade angular e $\xi = 0$, caso o escoamento seja rotacional, ocorre o oposto, pois as partículas rodam ao redor do seu próprio eixo e a velocidade angular é diferente de zero.

Outra análise a respeito dos fluidos refere-se a quantidade de matéria que flui em um determinado escoamento. Quando consideramos um volume diferencial, a quantidade de matéria que nela entra deve ser equivalente a quantidade de matéria

que dela sai. Esta ideia pode ser expressa a partir da **equação da continuidade**, descrita matematicamente como:

$$\frac{\partial \rho v_x}{\partial x} + \frac{\partial \rho v_y}{\partial y} + \frac{\partial \rho v_z}{\partial z} + \frac{\partial \rho}{\partial t} = 0. \quad (2.19)$$

O termo ρ é a massa específica; v_x , v_y e v_z são as componentes da velocidade nos eixos x , y e z respectivamente e; t é o tempo. A mesma equação pode ser escrita de forma mais compacta ao reconhecermos e substituírmos o termo divergente nela presente, de modo que a mesma fique:

$$\nabla \cdot \rho \mathbf{v} + \frac{\partial \rho}{\partial t} = 0, \quad (2.20)$$

na qual \mathbf{v} é o vetor velocidade. De acordo com Fox e McDonald [20] a equação da continuidade pode ser simplificada para a situação na qual a massa específica é constante (fluido incompressível), tornando-se:

$$\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z} = \nabla \cdot \mathbf{v} = 0. \quad (2.21)$$

Outra simplificação usual é quando o escoamento independe do tempo. Neste caso, a equação da continuidade pode ser reescrita como:

$$\frac{\partial \rho v_x}{\partial x} + \frac{\partial \rho v_y}{\partial y} + \frac{\partial \rho v_z}{\partial z} = \nabla \cdot \rho \mathbf{v} = 0. \quad (2.22)$$

Além da equação da continuidade, outra equação amplamente utilizada nos estudos de mecânica dos fluidos é a equação da **quantidade de movimento**, que é resultado da segunda Lei de Newton aplicada a um elemento diferencial de fluido, balanceando as forças de superfície e de campo que atuam neste elemento. As equações 2.23, 2.24 e 2.25 descrevem a quantidade de movimento, que descrevem o comportamento de qualquer fluido, nãis quais g_x , g_y e g_z são as componentes do vetor aceleração da gravidade nos eixos x , y e z , respectivamente e; σ e τ são as tensões normais e cisalhantes, respectivamente.

$$\rho \left(\frac{\partial v_x}{\partial t} + v_x \frac{\partial v_x}{\partial x} + v_y \frac{\partial v_x}{\partial y} + v_z \frac{\partial v_x}{\partial z} \right) = \rho g_x + \frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z}, \quad (2.23)$$

$$\rho \left(\frac{\partial v_y}{\partial t} + v_x \frac{\partial v_y}{\partial x} + v_y \frac{\partial v_y}{\partial y} + v_z \frac{\partial v_y}{\partial z} \right) = \rho g_y + \frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z}, \quad (2.24)$$

$$\rho \left(\frac{\partial v_z}{\partial t} + v_x \frac{\partial v_z}{\partial x} + v_y \frac{\partial v_z}{\partial y} + v_z \frac{\partial v_z}{\partial z} \right) = \rho g_z + \frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z}. \quad (2.25)$$

Note que o lado direito das equações 2.23, 2.24 e 2.25 possui termos relativos à taxa de variação das tensões. Caso esteja-se considerando um fluido Newtoniano de escoamento incompressível, a viscosidade é constante, uma vez que a tensão de cisalhamento é proporcional à taxa de deformação angular. Desta forma, as equações da quantidade de movimento (2.23, 2.24 e 2.25) podem ser simplificadas de modo que se tornem as conhecidas **equações de Navier-Stokes**, escritas nas suas componentes cartesianas como:

$$\rho \left(\frac{\partial v_x}{\partial t} + v_x \frac{\partial v_x}{\partial x} + v_y \frac{\partial v_x}{\partial y} + v_z \frac{\partial v_x}{\partial z} \right) = \rho g_x - \frac{\partial p}{\partial x} + \mu \left(\frac{\partial^2 v_x}{\partial x^2} + \frac{\partial^2 v_x}{\partial y^2} + \frac{\partial^2 v_x}{\partial z^2} \right), \quad (2.26)$$

$$\rho \left(\frac{\partial v_y}{\partial t} + v_x \frac{\partial v_y}{\partial x} + v_y \frac{\partial v_y}{\partial y} + v_z \frac{\partial v_y}{\partial z} \right) = \rho g_y - \frac{\partial p}{\partial y} + \mu \left(\frac{\partial^2 v_y}{\partial x^2} + \frac{\partial^2 v_y}{\partial y^2} + \frac{\partial^2 v_y}{\partial z^2} \right), \quad (2.27)$$

$$\rho \left(\frac{\partial v_z}{\partial t} + v_x \frac{\partial v_z}{\partial x} + v_y \frac{\partial v_z}{\partial y} + v_z \frac{\partial v_z}{\partial z} \right) = \rho g_z - \frac{\partial p}{\partial z} + \mu \left(\frac{\partial^2 v_z}{\partial x^2} + \frac{\partial^2 v_z}{\partial y^2} + \frac{\partial^2 v_z}{\partial z^2} \right). \quad (2.28)$$

As equações de Navier-Stokes também pode ser apresentadas na sua forma vetorial, tal como é possível observar na equação 2.29,

$$\rho \left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot (\nabla) \mathbf{v} \right) = -\nabla p + \rho \mathbf{g} + \mu \nabla^2 \mathbf{v}, \quad (2.29)$$

na qual o termo $\frac{\partial \mathbf{v}}{\partial t}$ refere-se a variação da velocidade no tempo; o termo $(\mathbf{v} \cdot \nabla) \mathbf{v}$ é conhecido como termo convectivo; o gradiente de pressão (∇p) indica que o fluido escoar na direção com maior variação de pressão; o termo $\rho \mathbf{g}$ representa as forças de corpo atuantes em um fluido, \mathbf{g} , a gravidade; o termo $\mu \nabla^2 \mathbf{v}$ refere-se a difusividade. Considerando a equação 2.29 e a relação entre viscosidade dinâmica, viscosidade cinemática e massa específica, expressa na equação 2.4, a equação de Navier-Stokes pode também ser escrita vetorialmente em função da viscosidade cinemática (ν):

$$\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} = -\frac{1}{\rho} \nabla p + \mathbf{g} + \nu \nabla^2 \mathbf{v}. \quad (2.30)$$

Por fim, outra equação importante para a análise do escoamento de um fluido é a **função corrente** ($\psi(x, y, t)$). Note que esta é uma função válida apenas para escoamentos bidimensionais, conseqüentemente apresentando apenas os componentes u e v da velocidade, de tal forma que:

$$v_x = \frac{\partial \psi}{\partial y} \quad \text{e} \quad v_y = -\frac{\partial \psi}{\partial x}. \quad (2.31)$$

2.4 Método de Elementos Finitos - MEF

O Método de Elementos Finitos (MEF) é utilizado para analisar um meio contínuo a partir da discretização deste meio e posterior análise do comportamento de sub-regiões de geometria simples (triângulos, quadriláteros, tetraedros, hexaedros, etc.), conhecida como elementos finitos (Figura 2.3). O MEF permite uma análise das leis matemáticas que regem o comportamento de pequenas partições contribuindo para o conhecimento do comportamento do meio contínuo como um todo. Desta forma, este método permite a solução de um problema complexo, eventualmente sem solução analítica, a partir da divisão deste problema em problemas menores com soluções matemáticas mais simples. Trata-se então de um método que fornece soluções aproximadas [22, 23].

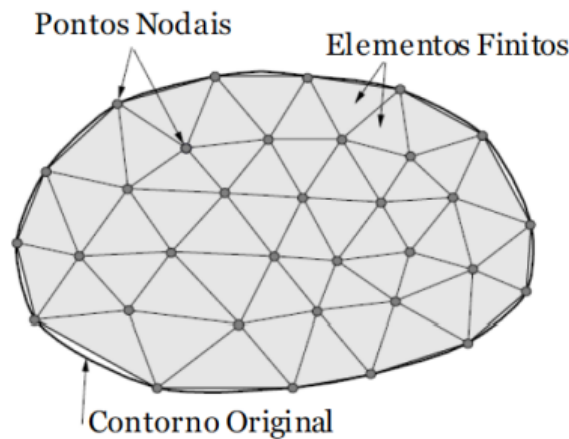


Figura 2.3: Exemplo de discretização de um domínio em sub-regiões por MEF.
Fonte: Melaconian, 2014 [23].

A primeira referência explícita ao método dos elementos finitos, inclusive quanto à sua nomenclatura, foi feita na década de 60, visando solucionar um problema de mecânica dos sólidos; embora algumas décadas antes já houvesse metodologias próximas ao método sendo propostas. A partir dos anos 70, este método começou a ganhar força para a solução de problemas envolvendo fluidos a partir do método de Galerkin [22, 24].

Por se tratar de um método numérico, de solução aproximada, é importante que se esteja atento às possíveis fontes de erros de aproximação e também deve-se estar atento à magnitude desses erros. Por isso, conhecer o procedimento de modelagem

é de extrema importância. O MEF pode ser visto de duas perspectivas distintas; a primeira do ponto de vista do usuário de um software comercial pronto para a solução de problemas baseado no método; o segundo do ponto de vista do desenvolvedor do software idealiza e realiza a programação do seu próprio software. Independente do ponto de vista, o conhecimento do método é muito relevante. Se o desenvolvedor precisa conhecer o método para poder programar o software, o usuário também precisa conhecer bem o método para que, com base em uma análise crítica, não tome como verdade absoluta os resultados obtidos em uma simulação. É necessário, então, que o usuário tenha conhecimento não somente das características físicas e geometrias que está buscando resolver, como das limitações do MEF e os potenciais erros que podem ser inseridos, a fim de evitar resultados incompatíveis com possíveis erros e análises não-realísticas da solução de um problema [22].

Portanto, o primeiro ponto a ser considerado é que o domínio deve ser dividido em um número finito de segmentos, áreas ou volumes (para casos de problemas em 1D, 2D e 3D, respectivamente), designados elementos finitos (Figura 2.4). Estes elementos podem ter diversas formas geométricas e dentro de cada elemento existem pontos representativos, chamados de nós, de modo que um nó na fronteira de um elemento finito pode pertencer a mais de um elemento. Ao conjunto de organização dos subdomínios em elementos com seus nós denominamos malha (Figuras 2.3, 2.5 e 2.6). A quantidade de nós, o tamanho e o tipo de elementos são fatores que influenciam na precisão do método [23, 22].

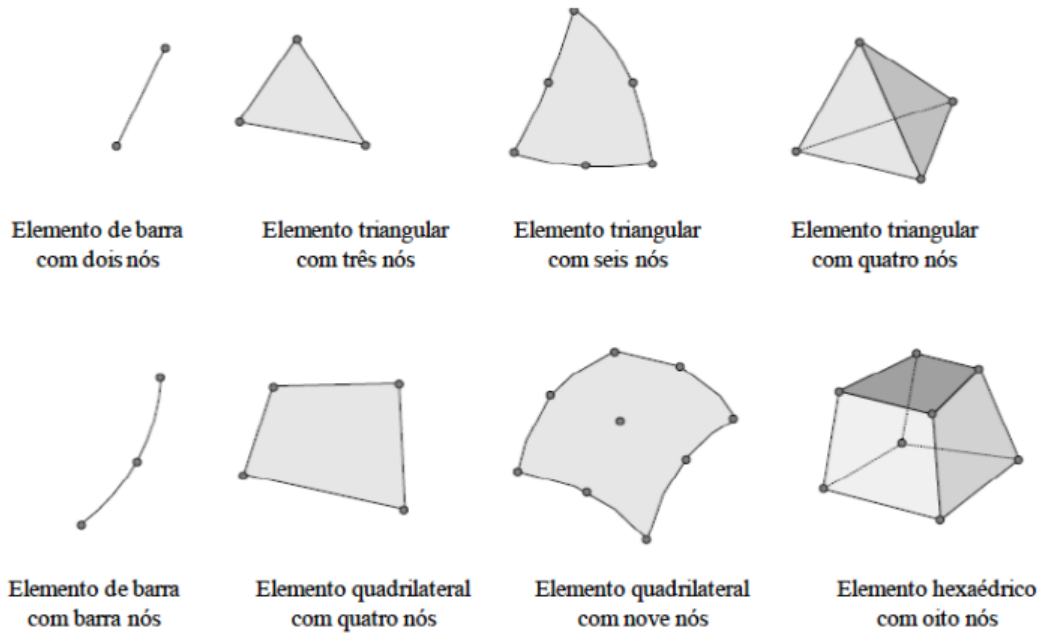


Figura 2.4: Tipos de elementos. Fonte: Melaconian, 2014 [23].

As malhas podem ser geradas com auxílio de *softwares* geradores de malhas, que relacionam o valor de um conjunto de variáveis definidas em cada nó (graus de liberdade), ou seja, no caso de um problema unidimensional, um nó tem 1 grau de liberdade (na direção do eixo x), enquanto em um problema bidimensional e tridimensional os nós possuem 2 e 3 graus de liberdade, respectivamente (eixo x e y no caso do bidimensional; e eixos x , y e z , no caso do tridimensional). A informação quanto à interconectividade dos pontos em malhas é guardada em uma matriz de conectividade (IEN) (Figura 2.5). E o valor de determinada variável entre os nós é descrito na forma de sistemas lineares (matriz de rigidez), sendo o número de equações do sistema linear em questão proporcional ao número de nós. Se um domínio for discretizado a ter n nós, então o número de graus de liberdade é dado por $N_{gl} = n \times n_{gl}$, onde n_{gl} é o número de graus de liberdade por nó [22, 23, 24].

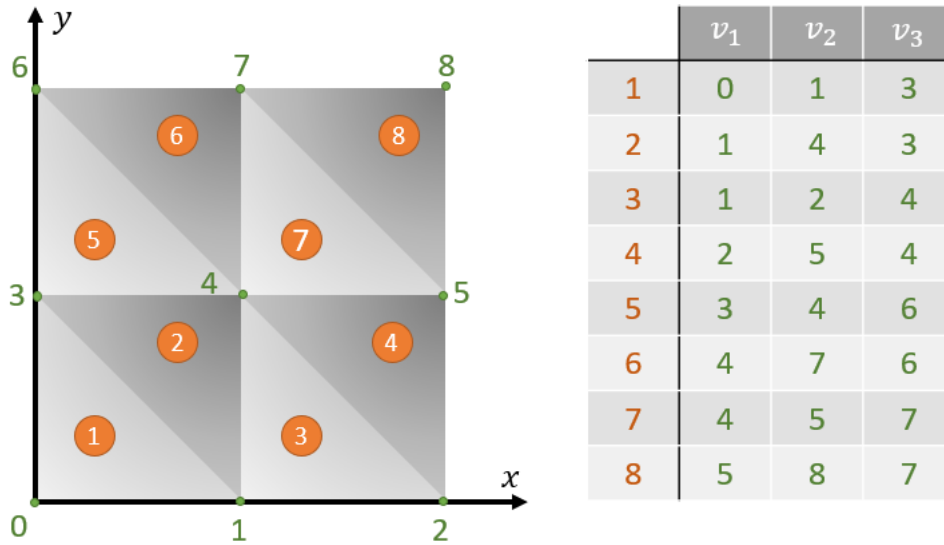


Figura 2.5: Exemplo de uma malha estruturada com 9 nós, 8 elementos triangulares e a respectiva IEN. Fonte: Própria.

A análise do MEF tem como ponto de partida a relação entre forças nodais de cada elemento individual. O conceito de rigidez vem da ideia aplicada a elementos elásticos, onde há uma relação entre a força aplicada e o deslocamento. Pela Lei de Hooke a constante elástica da mola (k) é coeficiente de rigidez responsável por fazer esta relação [23, 25]. Então, de modo análogo:

$$\begin{Bmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \\ f_n \end{Bmatrix} = \begin{bmatrix} k_{11} & k_{12} & k_{13} & \dots & k_{1n} \\ k_{21} & k_{22} & k_{23} & \dots & k_{2n} \\ k_{31} & k_{32} & k_{33} & \dots & k_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ k_{n1} & k_{n2} & k_{n3} & \dots & k_{nn} \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_n \end{Bmatrix} \quad (2.32)$$

Ou de modo mais compacto:

$$\{F\} = \{K\} \{d\} \quad (2.33)$$

Onde:

$\{F\}$: Matriz coluna que guarda todas as cargas nodais;

$\{d\}$: Matriz coluna com todos os deslocamentos nodais

Após discretizado o modelo a ser estudado, definido as propriedades dos elementos e montada a matriz de rigidez dos elementos, é necessário a aplicação das cargas e definição das condições de contorno. Todas estas etapas descritas até aqui fazem

parte do pré-processamento do problema. E ainda no pré-processamento que as propriedades físicas e/ou mecânicas do(s) material(is) a ser(em) estudado(s) é(são) definida(s). Somente após estas etapas serem realizadas é que pode-se partir para a etapa de processamento (solução do sistema de equações) [22, 23].

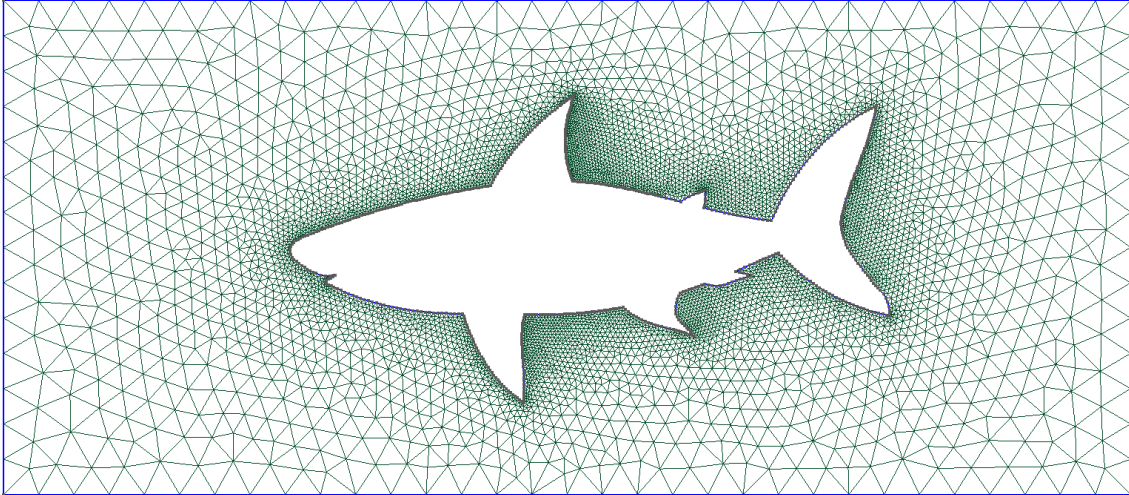


Figura 2.6: Exemplo de malha com elementos triangulares de diferentes tamanhos.
Fonte: Própria.

Na etapa de processamento são geradas as equações necessárias para a solução do problema. A quantidade de equações está diretamente relacionada a quantidade de elementos da malha gerada, impactando diretamente na qualidade do resultado e no tempo de resolução necessário. Cabe ressaltar que soluções explícitas requerem maior quantidade de processamento (memória do *hardware*), enquanto o método implícito exige menos. No método implícito, o cálculo do deslocamento $\{d\}$ é realizado a partir da inversão da matriz $[K]$:

$$\{d_{t+\Delta t}\} = [K]^{-1}\{F_{t+\Delta t}\} \quad (2.34)$$

Contudo, esta abordagem tem como um dos principais problemas a convergência do processo iterativo, em alguns casos [23].

Capítulo 3

Metodologia

3.1 Modelo Teórico

3.1.1 Formulação Função Corrente-Vorticidade

Para solucionar o problema da função corrente-vorticidade, utiliza-se as equações de quantidade de movimento e conservação da massa. Como visto no capítulo 2, para um fluido newtoniano (ν constante) e incompressível (ρ contante), a equação da quantidade de movimento torna-se a conhecida Equação de Navier-Stokes [20], que na forma vetorial é dada por:

$$\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{v} + \mathbf{g}, \quad (3.1)$$

na qual ρ é a massa específica do fluido, ν a viscosidade cinemática, \mathbf{v} o campo de velocidade e, p a pressão escalar. Como tanto \mathbf{v} como p são incógnitas, a solução da equação exige que ambas sejam satisfeitas simultaneamente, ou seja, as deixa acopladas, dificultando ainda mais a solução da equação e exigindo também uma outra equação para que o número de incógnitas e equações seja igual, permitindo a solução do sistema. Neste caso, a equação adicional a ser considerada é a de conservação da massa:

$$\nabla \cdot \mathbf{v} = 0. \quad (3.2)$$

Aplicando o operador rotacional ($\nabla \times$) na equação de Navier-Stokes (equação 2.29), obtemos a equação da vorticidade:

$$\nabla \times \left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) = \nabla \times \left(-\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{v} + \mathbf{g} \right). \quad (3.3)$$

Mas, na equação 3.3, o termo $\mathbf{v} \cdot \nabla \mathbf{v}$ pode ser reescrito como $\nabla \frac{v^2}{2} - \mathbf{v} \times \nabla \times \mathbf{v}$ de modo que ela se torne:

$$\nabla \times \left(\frac{\partial \mathbf{v}}{\partial t} + \nabla \frac{v^2}{2} - \mathbf{v} \times (\nabla \times \mathbf{v}) \right) = \nabla \times \left(-\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{v} + \mathbf{g} \right). \quad (3.4)$$

Aplicando a multiplicação distributiva do operador rotacional ($\nabla \times$) tanto do lado esquerdo, como do lado direito da equação 3.4, teremos:

$$\frac{\partial(\nabla \times \mathbf{v})}{\partial t} + \nabla \times \nabla \frac{v^2}{2} - \nabla \times (\mathbf{v} \times (\nabla \times \mathbf{v})) = -\frac{1}{\rho} \nabla \times \nabla p + \nu \nabla^2 (\nabla \times \mathbf{v}) + \nabla \times \mathbf{g}. \quad (3.5)$$

Como $\nabla \times \nabla = 0$ e sendo $v^2 = \mathbf{v} \cdot \mathbf{v}$ [26], os termos $\nabla \times \nabla \frac{v^2}{2}$ e $-\frac{1}{\rho} \nabla \times \nabla p$ na equação 3.5 são zerados e a equação passa a ser dada por:

$$\frac{\partial(\nabla \times \mathbf{v})}{\partial t} - \nabla \times (\mathbf{v} \times (\nabla \times \mathbf{v})) = \nu \nabla^2 (\nabla \times \mathbf{v}) + \nabla \times \mathbf{g}. \quad (3.6)$$

Sabendo que $\nabla \times \mathbf{v} = \omega$, a equação 3.6 reduz-se a:

$$\frac{\partial \omega}{\partial t} + \mathbf{v} \cdot \nabla \omega - \omega \cdot \nabla \mathbf{v} = \nu \nabla^2 \omega + \nabla \times \mathbf{g}, \quad (3.7)$$

para casos de problemas bidimensionais, o termo $\omega \cdot \nabla \mathbf{v}$ do lado esquerdo da equação 3.7 zera, uma vez que a vorticidade (ω_z) é perpendicular a $\nabla \mathbf{v}$, deixando a equação de Transporte da Vorticidade em 2D como:

$$\frac{\partial \omega_z}{\partial t} + \mathbf{v} \cdot \nabla \omega_z = \nu \nabla^2 \omega_z. \quad (3.8)$$

Considerando a definição de potencial de velocidade pela função escalar, teremos que os campos de velocidade em um escoamento bidimensional são:

$$v_x = \frac{\partial \psi}{\partial y} \quad \text{e} \quad v_y = -\frac{\partial \psi}{\partial x}. \quad (3.9)$$

Considerando a definição de vorticidade em um escoamento 2D:

$$\omega_z = \omega \mathbf{k} = \frac{\partial v_y}{\partial x} - \frac{\partial v_x}{\partial y}, \quad (3.10)$$

E substituindo as velocidades definidas acima, obtemos a Formulação corrente-vorticidade:

$$\nabla^2 \psi = -\omega_z. \quad (3.11)$$

As equações 3.8, 3.9, 3.10 e 3.11 são as equações chaves que serão utilizadas no desenvolvimento da solução do problema deste trabalho através do Método de Elementos finitos (MEF), que será melhor descrito na seção 3.2.

3.2 Método Numérico

3.2.1 O MEF na Função Corrente Vorticidade

Conforme mencionado anteriormente, no Método de Elementos Finitos (MEF) o domínio de interesse é dividido em sub-domínios, chamados de elementos. Uma função simples é utilizada para caracterizar a variação de cada variável dentro de cada elemento (função de forma). A justaposição da variação das variáveis em cada elemento é usada para descrever todo o campo da variável de interesse. A partir disto, o primeiro passo do MEF é transformar a equação original (forma forte) em uma forma fraca, a partir da ponderação da mesma por um função peso e a integração sobre um domínio. Após, reduz-se a ordem e faz-se aproximações a partir da substituição de termos, de modo que resulte em um sistema linear que poderá ser resolvido por meio de métodos de álgebra linear, permitindo que a solução aproximada seja encontrada.

No caso da abordagem de função corrente-vorticidade em escoamentos bidimensionais, sabe-se que a forma forte da equação de transporte da vorticidade pode ser expressa por:

$$\frac{\partial \omega_z}{\partial t} + \mathbf{v} \cdot \nabla \omega_z = \nu \nabla^2 \omega_z, \quad (3.12)$$

na qual ω_z é a vorticidade no eixo z , \mathbf{v} é o campo vetorial de velocidade e ν é a velocidade cinemática do fluido, conforme visto anteriormente (equação 3.8).

No passo seguinte do MEF inicia-se o processo de transformação da forma forte em forma fraca. Para tal, pondera-se a equação em sua forma forte (equação 3.12) com a função peso ($w(x, y)$) e em seguida integra-se a função ponderada no domínio (Ω), resultando em:

$$\int_{\Omega} w \left(\frac{\partial \omega_z}{\partial t} + \mathbf{v} \cdot \nabla \omega_z - \nu \nabla^2 \omega_z \right) d\Omega = 0. \quad (3.13)$$

Aplicando-se a multiplicação distributiva da função peso (w) na equação 3.13 e separando as integrais do resultado desta distribuição tem-se:

$$\int_{\Omega} w \frac{\partial \omega_z}{\partial t} d\Omega + \int_{\Omega} w \mathbf{v} \cdot \nabla \omega_z d\Omega - \int_{\Omega} w \nu \nabla^2 \omega_z d\Omega = 0. \quad (3.14)$$

Visando a redução da ordem do Laplaciano (termo de segunda ordem na terceira integral da equação 3.14 - ∇^2), com base na identidade de Green, integra-se por

partes o termo em questão de modo que a equação 3.14 torna-se:

$$\int_{\Omega} w \frac{\partial \omega_z}{\partial t} d\Omega + \int_{\Omega} w \mathbf{v} \cdot \nabla \omega_z d\Omega - \int_{\Gamma} w \nu \nabla \omega_z d\Omega + \int_{\Omega} \nabla w \cdot \nu \nabla \omega_z d\Omega = 0, \quad (3.15)$$

onde Γ refere-se a região do contorno e Ω ao domínio. Em seguida, defini-se da função vorticidade ($\omega_z(x, y)$) e a função peso ($w(x, y)$) como uma combinação linear tal que:

$$\omega_z(x, y) = \sum_{i=0}^{\infty} N_i(x, y) a_i, \quad (3.16)$$

$$w(x, y) = \sum_{j=0}^{\infty} N_j(x, y) b_j, \quad (3.17)$$

onde N_i e N_j são de forma (ou interpolação) e a_i e b_j são constantes a serem determinadas. Note, que até este ponto, nenhuma aproximação foi realizada. Para realizar a aproximação, trunca-se os somatórios infinitos das equações 3.16 e 3.17 no número total de pontos da malha (n), de modo que as funções vorticidade ($\omega_z(x, y)$) e peso ($w(x, y)$) sejam aproximadas como:

$$\omega_z(x, y) \approx \sum_{i=0}^n N_i(x, y) a_i, \quad (3.18)$$

$$w(x, y) \approx \sum_{j=0}^n N_j(x, y) b_j. \quad (3.19)$$

Aplicando as aproximações das equações 3.18 e 3.19 na equação 3.15, esta será reescrita como:

$$\begin{aligned} & \int_{\Omega} \sum_j^n \sum_j^n N_j w_j \frac{\partial (N_i \omega_{z_i})}{\partial t} d\Omega + \int_{\Omega} \sum_j^n \sum_j^n N_j w_j \mathbf{v} \cdot \nabla (N_i \omega_{z_i}) d\Omega \\ & - \int_{\Gamma} \sum_j^{n\Gamma} \sum_j^{n\Gamma} N_j w_j \nu \nabla (N_i \omega_{z_i}) d\Gamma + \int_{\Omega} \sum_j^n \sum_j^n N_j w_j \nabla (N_i \omega_{z_i}) \cdot \nu \nabla (N_i \omega_{z_i}) d\Omega = 0. \end{aligned} \quad (3.20)$$

Mas, o método de Galerkin as funções de interpolação para o campo de interesse e a função peso são iguais:

$$N_i(x, y) = N_j(x, y). \quad (3.21)$$

Considerando a equação 3.21, substitui-se os \sum_j e \sum_j por \sum_e na equação 3.20, onde o subíndice e do somatório \sum_e refere-se a “elemento”. Além disto, retira-se as constantes espaciais w_z das integrais e divide-se o resultado por w_j , resultando em:

$$\begin{aligned} & \sum_e \frac{\partial \omega_{z_i}}{\partial t} \int_{\Omega} N_i N_j d\Omega + \sum_e \omega_{z_i} \int_{\Omega} \mathbf{v} \cdot (N_j \nabla N_i) d\Omega \\ & - \sum_e \omega_{z_i} \int_{\Gamma} N_j (\nu \nabla N_i) d\Gamma + \sum_e \omega_{z_i} \int_{\Omega} \nabla N_j \cdot (\nu \nabla N_i) d\Omega = 0. \end{aligned} \quad (3.22)$$

Então, neste momento pode-se utilizar o conceito de diferenças finitas progressivas (aquela no qual considera-se o tempo posterior para a análise) na discretização do tempo, de modo que equação fique:

$$\begin{aligned} & \sum_e \left(\frac{\omega_{z_i}^{n+1} - \omega_{z_i}^n}{\Delta t} \right) \int_{\Omega} N_i N_j d\Omega + \sum_e \omega_{z_i} \int_{\Omega} \mathbf{v} \cdot (N_j \nabla N_i) d\Omega \\ & - \sum_e \omega_{z_i} \int_{\Gamma} N_j (\nu \nabla N_i) d\Gamma + \sum_e \omega_{z_i} \int_{\Omega} \nabla N_j \cdot (\nu \nabla N_i) d\Omega = 0. \end{aligned} \quad (3.23)$$

Esta equação terá alguns termos com correspondente matricial, tal que:

$$\int_{\Omega} N_i N_j d\Omega = \mathbf{M}, \quad (3.24)$$

$$\int_{\Omega} N_j \nabla N_i d\Omega = \mathbf{G}, \quad (3.25)$$

$$\int_{\Omega} \nabla N_j \cdot \nabla N_i d\Omega = \mathbf{K}, \quad (3.26)$$

onde \mathbf{M} é uma integral sem derivadas e conhecida como matriz de massa; \mathbf{G} é uma integral um termo sem derivada e um termo com gradiente e conhecida como matriz do gradiente e; \mathbf{K} é uma integral com multiplicação escalar entre gradientes, conhecida como matriz de rigidez (ou viscosa). Considerando que $w_j = 0$ para condição de contorno de Dirichlet e que tanto a velocidade (\mathbf{v}) como a viscosidade cinemática (ν) são constantes, a equação do elemento e pode ser descrita como:

$$m_{ij}^e \left(\frac{\omega_{z_i}^{n+1} - \omega_{z_i}^n}{\Delta t} \right) + \mathbf{v} \cdot g_{ij}^e \omega_{z_i} + \nu k_{ij}^e \omega_{z_i} = 0. \quad (3.27)$$

Note que está equação refere-se as componentes Locais, de modo que o sobre índice e refere-se a elemento, e que os subíndices i e j versam sobre os triângulos do elemento, variando de 1 a 3. Após o *Assembling* (junção das matrizes dos elementos

em uma única matriz geral), a equação Global do transporte da vorticidade será dada pela equação:

$$M_{ij} \left(\frac{\omega_{z_i}^{n+1} - \omega_{z_i}^n}{\Delta t} \right) + \mathbf{v} \cdot G_{ij} \omega_{z_i} + \nu K_{ij} \omega_{z_i} = 0, \quad (3.28)$$

e neste caso, os os subíndices i e j referem-se sobre a malha.

Para a formulação da forma fraca da função corrente (equação 3.29), repete-se o mesmo procedimento inicial feito no caso da formulação corrente-vorticidade:

$$\nabla^2 \psi = -\omega_z. \quad (3.29)$$

Primeiramente multiplica-se a função corrente (equação 3.29) pela função peso ($w(x, y)$) e realiza-se a integração no domínio (Ω) de modo que:

$$\int_{\Omega} w(\nabla^2 \psi + \omega_z) d\Omega = 0. \quad (3.30)$$

Após a multiplicação distributiva da função peso, tem-se:

$$\int_{\Omega} w(\nabla^2 \psi) d\Omega + \int_{\Omega} w \omega_z d\Omega = 0. \quad (3.31)$$

Após, realiza-se a integração por partes do Laplaciano (∇^2) e a separação do contorno (Γ) e do domínio (Ω). Em seguida aplica-se a identidade de Green, resultando na equação:

$$\int_{\Gamma} w(\nabla \psi) d\Gamma - \int_{\Omega} \nabla w \cdot \nabla \psi d\Omega + \int_{\Omega} w \omega_z d\Omega = 0. \quad (3.32)$$

Note que no caso da Função Corrente não há derivada no tempo. Sendo:

$$\psi(x, y) \approx \sum_{i=0}^n N_i(x, y) a_i, \quad (3.33)$$

$$\omega_z(x, y) \approx \sum_{i=0}^n N_i(x, y) b_i, \quad (3.34)$$

$$w(x, y) \approx \sum_{j=0}^n N_j(x, y) c_j. \quad (3.35)$$

Após as devidas substituições, considerando as equações 3.33, 3.34 e 3.35, a equação 3.32 pode ser reescrita como:

$$\int_{\Gamma} \sum_e^{n\Gamma} N_j w_j \left(\nabla \sum_i^{n\Gamma} N_i \psi_i \right) d\Gamma - \int_{\Omega} \nabla \sum_j^n N_j w_j \cdot \nabla \sum_i^n N_i \psi_i d\Omega + \int_{\Omega} \sum_j^n N_j w_j \sum_i^n N_i \omega_{z_i} d\Omega = 0 \quad (3.36)$$

Considerando o método de Galerkin, substitui-se os \sum_j e \sum_j por \sum_e e coloca-se em evidência as constantes ω_z e ψ , de modo que a equação fique:

$$\sum_{e\Gamma} \psi_i w_j \int_{\Gamma} N_j (\nabla N_i) d\Gamma - \sum_e \psi_i w_j \int_{\Omega} \nabla N_j \cdot \nabla N_i d\Omega + \sum_e w_z w_j \int_{\Omega} N_j N_i d\Omega = 0. \quad (3.37)$$

Dividindo a equação 3.37 por w_j e considerando que para a condição de contorno de Dirichlet $w_j = 0$, ficamos com a seguinte equação para os elementos:

$$-k_{ij}^e \psi_i + m_{ij}^e \omega_{z_i} = 0. \quad (3.38)$$

O após o *Assembling* (junção das matrizes dos elementos em uma única matriz geral) das matrizes globais K_{ij} e M_{ij} , a equação global da função corrente será dada por:

$$K_{ij} \psi_i = M_{ij} \omega_{z_i}. \quad (3.39)$$

Para a formulações do campos de velocidade de componentes do eixo x e y , o processo será, novamente análogo ao processo realizado no caso da função de transporte da vorticidade e da função corrente que foi descrita anteriormente. A partir das equações da forma forte da velocidade em x e y (equação 3.40):

$$v_x = \frac{\partial \psi}{\partial y} \quad \text{e} \quad v_y = -\frac{\partial \psi}{\partial x}. \quad (3.40)$$

Após a ponderação pela função peso ($w(x, y)$) e integração no domínio (Ω), a forma fraca das equações que representam o campos de velocidade da componente do eixo x e y (equação 3.40) (v_x e v_y , respectivamente) será:

$$\int_{\Omega} w \left(v_x - \frac{\partial \psi}{\partial y} \right) d\Omega \quad \text{e} \quad \int_{\Omega} w \left(v_y + \frac{\partial \psi}{\partial x} \right) d\Omega, \quad (3.41)$$

com as equações globais sendo:

$$M_{ij} v_{x_i} = G_{y_{ij}} \psi_i \quad \text{e} \quad M_{ij} v_{y_i} = -G_{x_{ij}} \psi_i. \quad (3.42)$$

Por fim, iremos transformar para a forma fraca a equação na forma forte da vorticidade em um escoamento bidimensional (equação 3.43).

$$\omega_z = \frac{\partial v_y}{\partial x} - \frac{\partial v_x}{\partial y}. \quad (3.43)$$

Mais uma vez, segue-se a mesma lógica aplicada às demais equações até aqui. Deste forma, após a multiplicação da equação 3.43 pela função peso ($w(x, y)$), tem-se:

$$\int_{\Omega} w \left(\omega_z - \frac{\partial v_y}{\partial x} + \frac{\partial v_x}{\partial y} \right) d\Omega = 0, \quad (3.44)$$

que, ao fim do processo resultará na seguinte equação global:

$$M_{ij}\omega_{z_i} = G_{x_{ij}}v_{y_i} - G_{y_{ij}}v_{x_i} \quad (3.45)$$

Desta forma, para a forma implícita, teremos as seguintes equações a serem implementadas:

- Transporte de vorticidade:

$$\left(\frac{M}{\Delta t} + \nu K + \mathbf{v} \cdot \mathbf{G} \right) \omega_z^{n+1} = \frac{M}{\Delta t} \omega_z^n + c.c. \quad (3.46)$$

- Formulação corrente-vorticidade:

$$K\psi = M\omega_z + c.c. \quad (3.47)$$

Equações auxiliares:

- Campos de velocidades (v_x e v_y) - a partir da função de corrente:

$$Mv_x = G_y\psi \quad (3.48)$$

$$Mv_y = -G_x\psi \quad (3.49)$$

- Vorticidade (ω_z) - a partir do gradiente de velocidades:

$$M\omega_z = G_xv_x - G_yv_y, \quad (3.50)$$

estando ψ fixa no tempo e ω_z variando com o tempo.

3.3 Gerador de Malhas

A fim de garantir a qualidade de uma boa malha, para geração de malha utilizou-se um *software* gratuito, robusto e consolidado, o **Gmsh** (versão 4.10), que permitiu a criação da malha desejada, com o controle dos elementos gerados conforme desejado [27].

O arquivo da malha computacional gerado pelo **Gmsh**, em extensão *.msh*, foi importado para o *Python*, utilizando-se a biblioteca *meshio*, e as informações relevantes para a solução do problema via o MEF, como a matriz de conectividade (IEN), foram obtidas em sequência, conforme pode ser visualizado no código fonte, apresentado no Apêndice A deste trabalho, das linhas 41 a 57.

3.4 Implementação do código

Após determinar as equações necessárias e definir a malha, o código numérico foi implementado na linguagem de programação *Python*. A escolha da linguagem se deu devido ao fato de ser uma linguagem relativamente simples, de fácil acesso e com vasto repertório de bibliotecas disponíveis. Para este trabalho foram utilizadas as bibliotecas *numpy*, *matplotlib*, *time* e *meshio* (versão 5.3.4). Além disso, cabe ressaltar que o código apresenta medidas adimensionais. O código, na íntegra, pode ser visualizado no Apêndice A deste trabalho.

Após implementação do código, foi realizada a validação do mesmo através de simulações de casos de escoamentos conhecidos, vastamente documentados na literatura, como o caso de um escoamento no interior de uma cavidade com uma tampa móvel (*Lid Driven*) e o escoamento entre placas planas.

3.5 Visualização dos resultados

Para visualizar os resultados obtidos com a implementação dos códigos optou-se por utilizar o *software* gratuito **Paraview** [28]. Isto pois, além de ser gratuito, este *software* é amplamente utilizado para visualização e pós processamento de dados, devido a sua confiabilidade e facilidade. Para tal, os resultados foram exportados via *Python* no formato *.vtk*, conforme pode ser visualizado no código fonte, apresentado

no Apêndice A deste trabalho, das linhas 267 a 274.

O **Paraview** foi utilizado para visualização dos resultados em 2D e obtenção dos valores das soluções obtidas a cada iteração. Para a confecção dos gráficos de dispersão com linhas suáveis, apresentados neste trabalho nas seções a seguir, utilizou-se o *software* **Excel**.

Capítulo 4

Verificação do Código

A implementação do código foi realizada em um computador com o processador Intel Core i7-10700K, 8-Core, 16-Threads, 3,8GHz e; com 32 GB de memória RAM. Os resultados são apresentados a seguir.

4.1 Escoamento de *Lid Driven*

Um problema comumente utilizado para verificação de códigos no campo de mecânica dos fluidos é o escoamento na cavidade de *Lid Driven*, por ser um problema amplamente testado e validado. Este problema consiste em uma cavidade na qual as velocidades verticais e horizontais, exceto no limite superior, onde uma superfície se desloca horizontalmente com uma velocidade constante. As velocidades verticais do fluido (v_y) são zero e as velocidades horizontais do fluido (v_x) são zero nas laterais e no limite inferior e igual a um no limite superior do domínio.

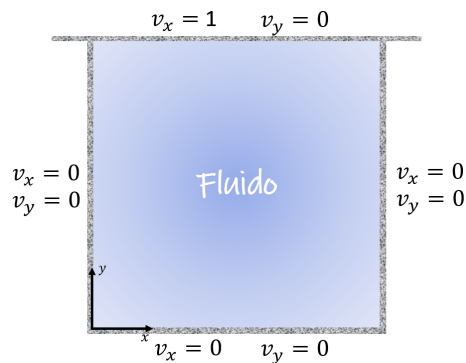


Figura 4.1: Desenho esquemático das condições de contorno do escoamento na cavidade *Lid Driven*, em uma cavidade de dimensões 1×1 . Fonte: Própria.

O desenho esquemático das condições de contorno pode ser visualizado na figura 4.1. Cabe ressaltar que a condição de contorno da vorticidade é calculada dentro do código, a partir da equação 3.50.

Para esta verificação foram testados 2 casos distintos. Um com o número de Reynolds igual a 10, e outro com o número de Reynolds igual a 100. No caso de $Re = 10$, foram feitos testes com diferentes malhas, diferenciando-se entre si apenas pelo seu refino quanto ao número de nós e elementos, e o passo de tempo e intervalos de tempos processados. Cabe ressaltar que os números de nós e números de elementos são consequência de parâmetro de refino ao redor do domínio e dos obstáculos impostos no código desenvolvido para a geração de malha no *software* GMSH (descrito no apêndice A deste trabalho). A malha gerada é uma malha não estruturada, de elementos triangulares. Para o cálculo da malha temporal, foi considerado a forma implícita *up-wind*. A tabela 4.1 apresenta os parâmetros utilizados em cada um dos testes.

Tabela 4.1: Parâmetros dos testes com diferentes malhas, passo de tempo e intervalos de tempo, para o caso da cavidade de *Lid Driven* e com $Re = 10$.

Teste	Número de nós	Número de elementos	Passo de tempo	dt	Tempo de execução [s]
1	511	1024	450	0.01	20
2	511	1024	4500	0.01	24
3	1935	3868	450	0.001	503
4	1935	3868	4500	0.001	4808
5	11833	23764	450	0.001	63467

As Figuras 4.2 a 4.6 apresentam os resultados encontrados para cada um dos testes feitos, para as quatro variáveis analisadas (Função Corrente, Vorticidade, Velocidade Horizontal e Velocidade Vertical).

A partir dos testes realizados, podemos identificar, imediatamente, que o tempo de processamento está relacionado ao refino da malha. Ou seja, quanto maior o número de nós e elementos, mais refinada a malha, mas também mais tempo é necessário para que o código seja executado, indo de segundos (no caso da pior malha) a horas (quase 1 dia de processamento). Cabe ressaltar que os intervalos de tempo

foram alterados do primeiro e segundo teste para os demais, devido a capacidade de processamento do computador utilizado, a fim de evitar erros de sobrecarregamento (*overflow*). Além disso, ressalta-se também que devido ao incremento substancial de tempo de processamento entre os tempos dos Testes 4 e 5, optou-se por analisar os resultados até então encontrados, antes de proceder um novo teste, semelhante ao Teste 5, se diferenciando apenas pelo passo de tempo.

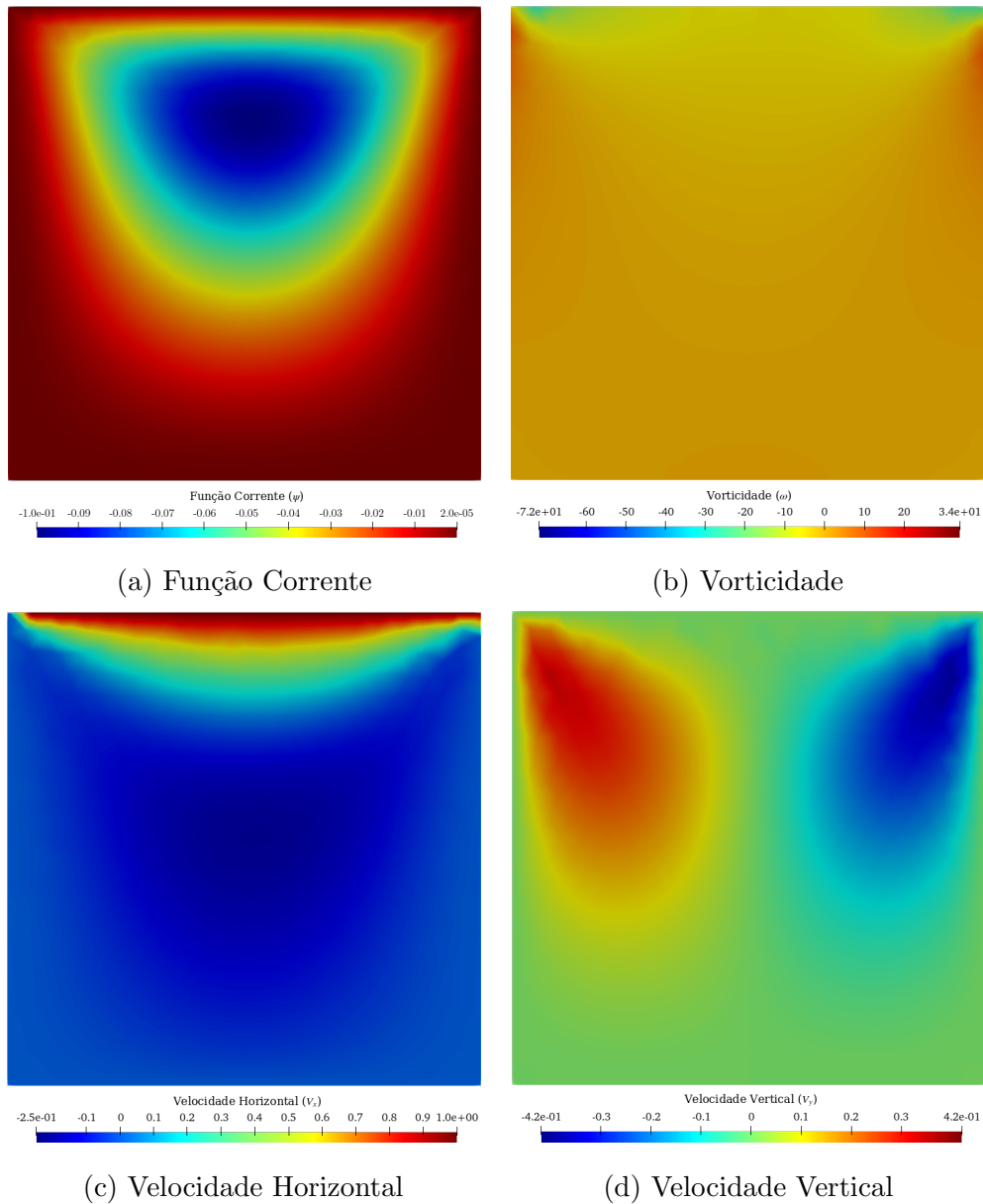


Figura 4.2: Teste 1 - Solução numérica do caso da cavidade de *Lid Driven* para $Re = 10$, malha com 511 nós e 1024 elementos, com 450 passos e $dt = 0.01$.

Comparando os Teste 1 e 2 (Figuras 4.2 e 4.3, respectivamente), observa-se que visualmente os testes apresentaram resultados semelhantes. Além disso, o tempo

de processamento não teve aumento significativo, levando o teste com mais passos cerca de 1,2 vezes mais do que o tempo de processamento com menos passos, mas em ambos casos um total de menos de meio minuto para obtenção do resultado final. Desta forma, optou-se por melhorar o refinamento da malha (aumentando o número de nós e elementos no domínio) e foram realizados os Testes 3 (Figura 4.4) e 4 (Figura 4.5) a fim de compará-los entre si e posteriormente com os Testes 1 e 2.

Novamente, visualmente os Testes 3 e 4 (Figuras 4.4 e 4.5, respectivamente) apresentam resultados semelhantes, contudo o tempo de processamento aumentou consideravelmente neste caso, quando o passo de tempo foi aumentado de 450 para 4500, resultando em um tempo quase 10 vezes maior para o caso da malha mais refinada (Teste 4), necessitando em um pouco mais de 8 minutos para o Teste 3 e mais de uma hora de processamento no Teste 4. Percebe-se então que, para malhas mais refinadas, o passo de tempo passa a ser um critério bastante relevante e que deve ser observado nas simulações de interesse deste trabalho.

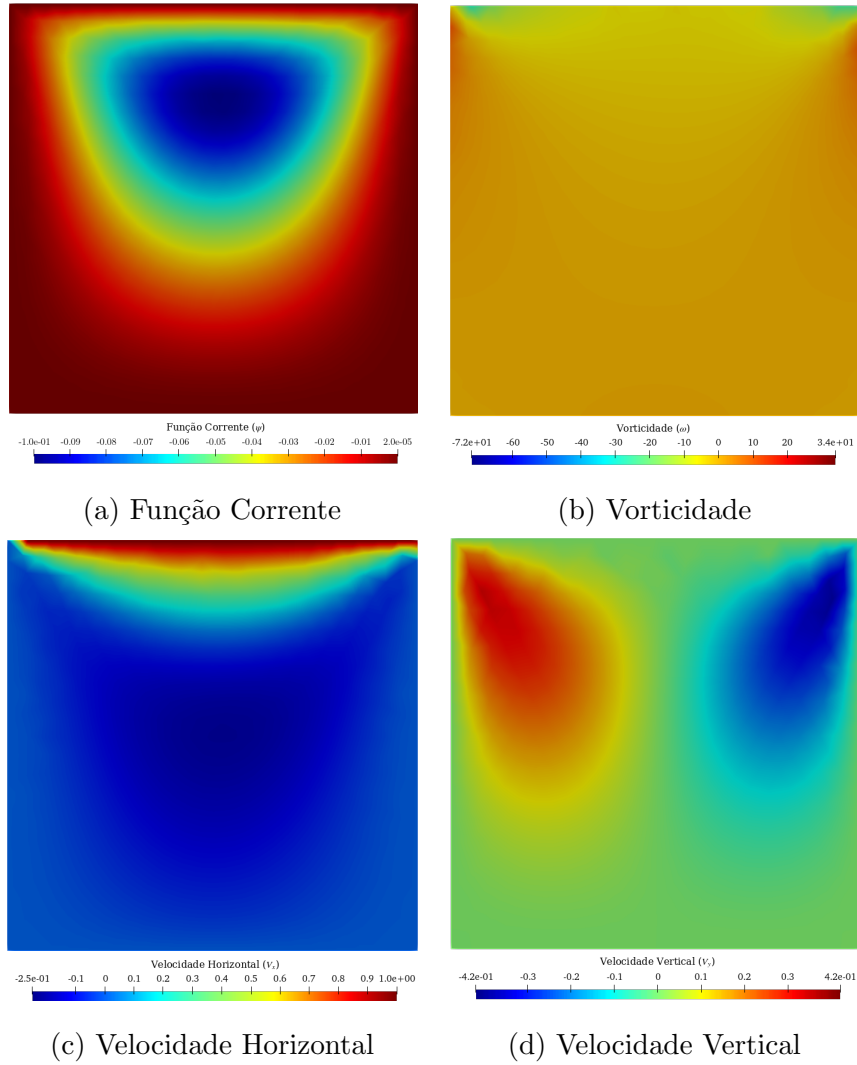


Figura 4.3: Teste 2 - Solução numérica do caso da cavidade de *Lid Driven* para $Re = 10$, malha com 511 nós e 1024 elementos, com 4500 passos e $dt = 0.01$.

Ao compararmos o Teste 3 (figura 4.4) com o Teste 1 (figura 4.2), e o Teste 4 (figura 4.5) com o Teste 2 (figura 4.3) também não observamos visualmente maiores diferenças. A principal diferença nesses casos é a definição da malha, que deixa de ser mais “grosseira”, com geometrias mais poligonais, e passa a ter uma geometria mais suave, de modo que nos Testes 1 e 2 a transição parece ser mais abrupta em de um ponto para outro no casos dos Testes 1 e 2.

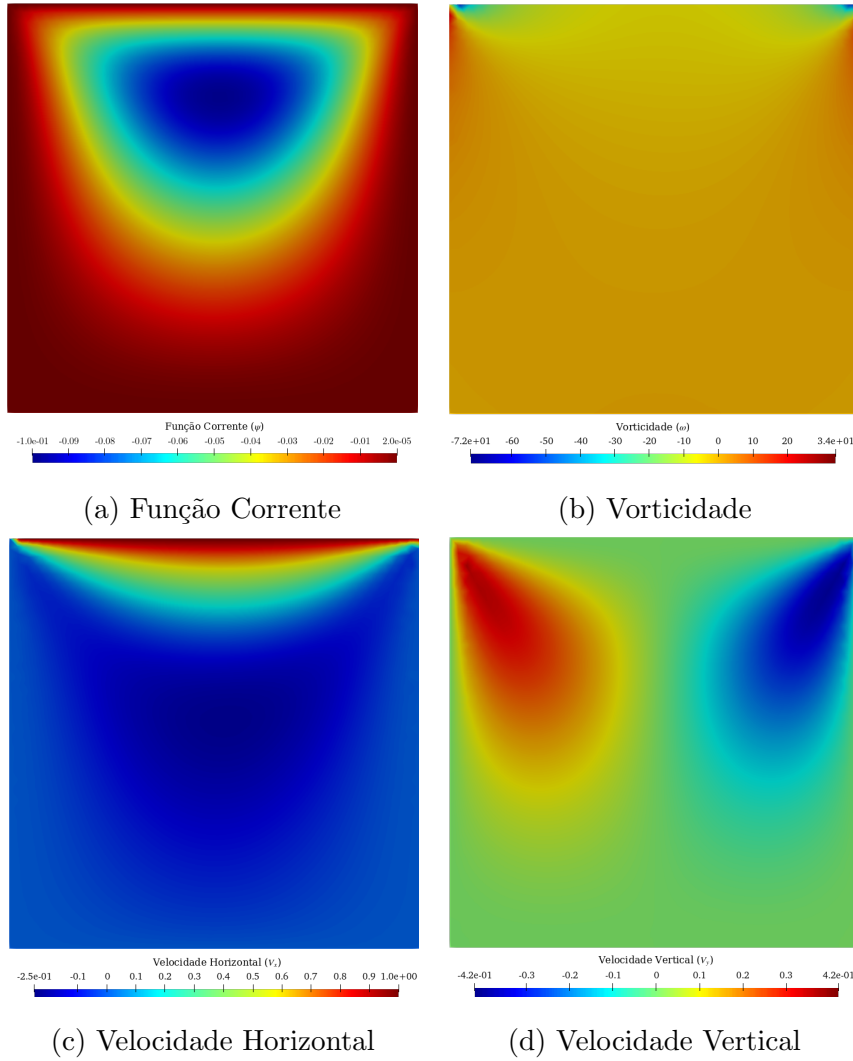


Figura 4.4: Teste 3 - Solução numérica do caso da cavidade de *Lid Driven* para $Re = 10$, malha com 1935 nós e 3868 elementos, com 450 passos e $dt = 0.01$.

Apesar de visualmente a qualidade da malha não aparentar fazer diferença substancial nos resultados ao observar os resultados dos Testes 1 a 4, optou-se por fazer novo teste, com uma malha ainda mais refinada, aproximando cada vez mais os elementos da malha para elementos quase infinitesimais de modo a tender a um domínio computacional mais próximo de um domínio contínuo. Contudo, dado a variação do tempo de processamento observado entre os Testes 3 e 4, foi realizado apenas um teste com uma malha melhor, mas com apenas 450 passos. O tempo de processamento do Teste 5 (4.6) foi ainda mais considerável, chegando a mais de 17 horas para obtenção do resultado. Desta forma, foi decidido que novos testes para este caso não seriam realizados, e os 5 testes feitos seriam comparados quantitativa-

vamente, após a análise qualitativa realizada até este ponto.

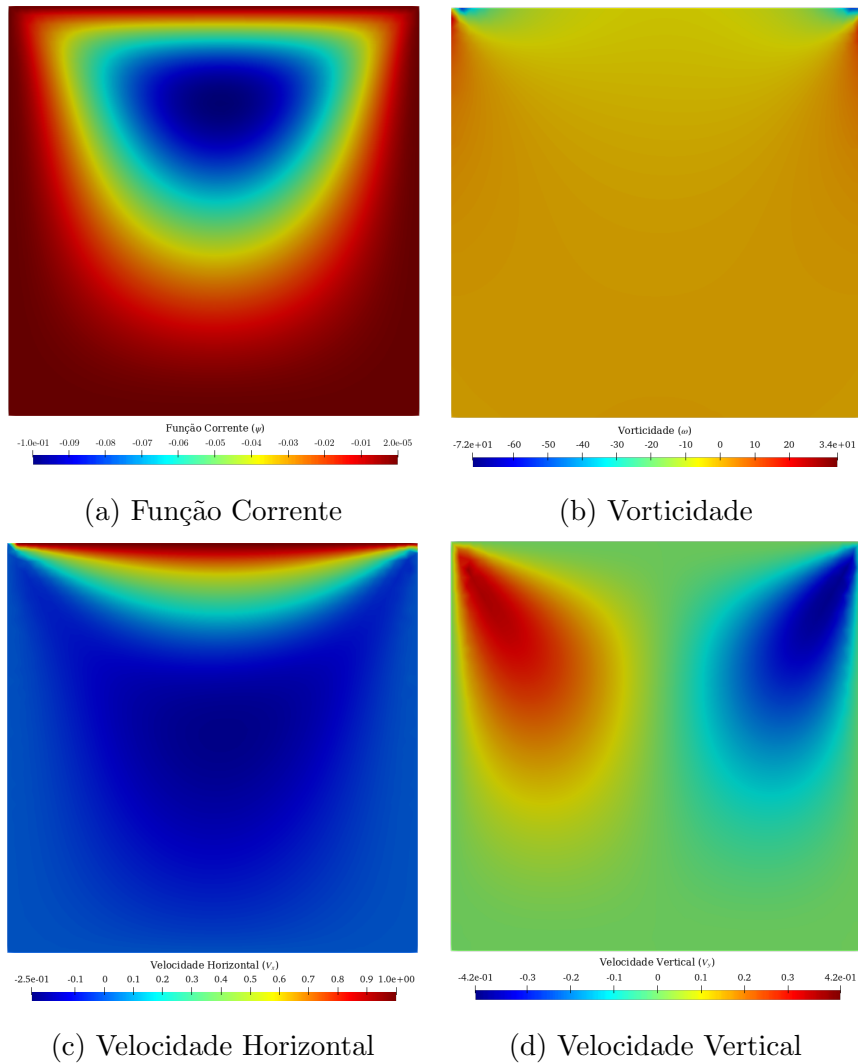


Figura 4.5: Teste 4 - Solução numérica do caso da cavidade de *Lid Driven* para $Re = 10$, malha com 1935 nós e 3868 elementos, com 4500 passos e $dt = 0.01$.

Dando continuidade a avaliação dos testes, partiu-se para uma análise mais acurada. Buscando validar os resultados encontrados e saber qual dos testes melhor otimiza as respostas encontradas, comparou-se as Velocidades Horizontais e Verticais, em cada um dos testes com valores já testados e consolidados na literatura. Para tal, foram plotados os valores das Velocidades Horizontal e Vertical das soluções encontradas (bem como a referência estabelecida pela literatura [29]) sobre os eixos $x = 0.5$ e $y = 0.5$ (figura 4.7). Os gráficos de ambos casos são expostos na figura 4.8.

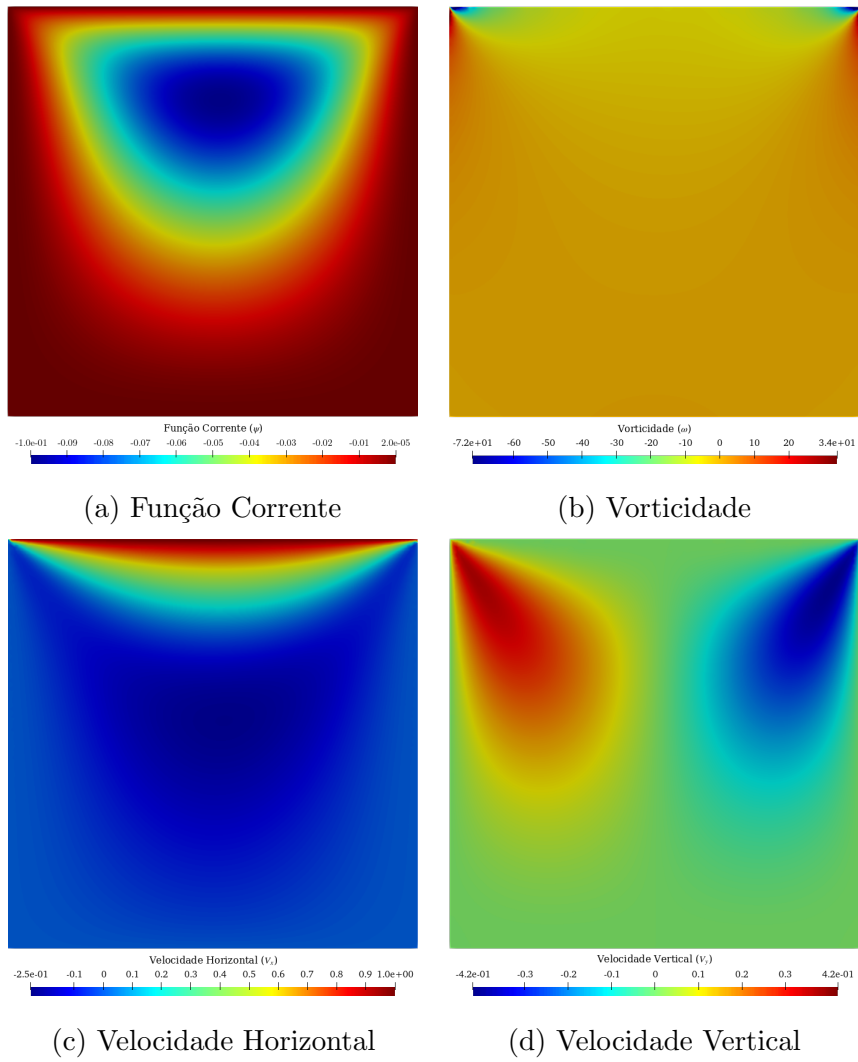
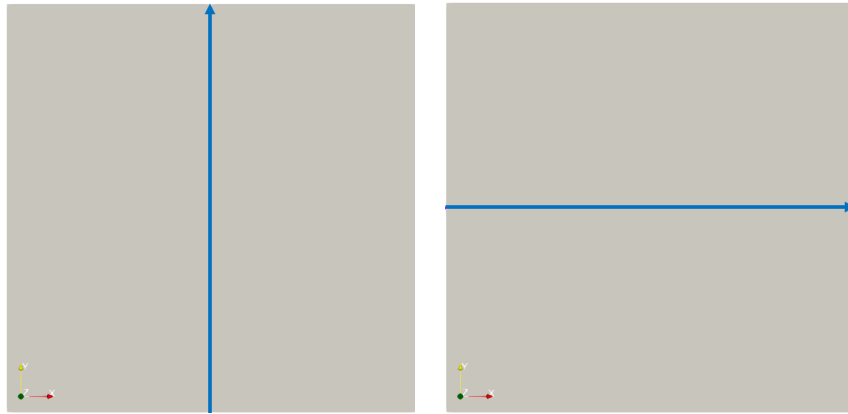


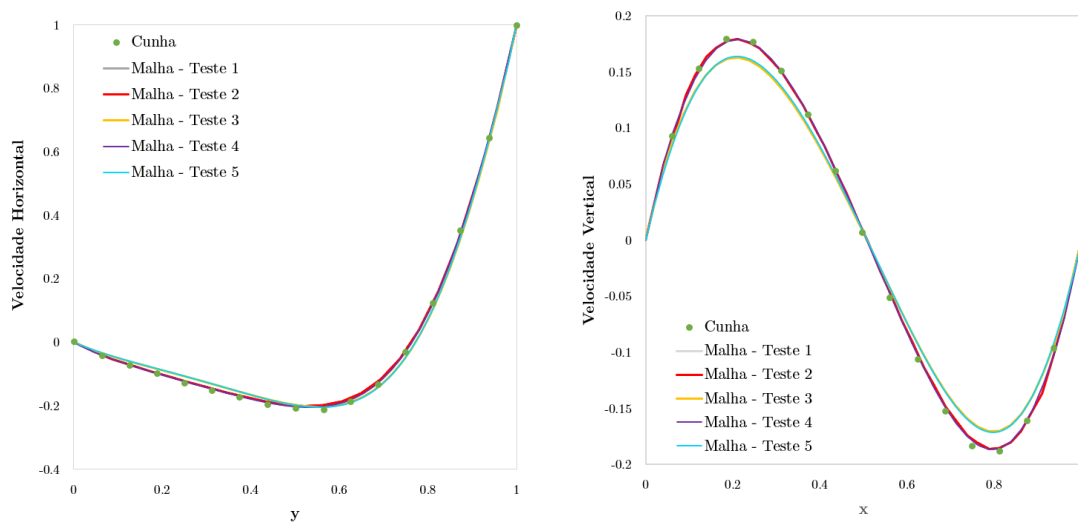
Figura 4.6: Teste 5 - Solução numérica do caso da cavidade de *Lid Driven* para $Re = 10$, malha com 11833 nós e 23764 elementos, com 450 passos e $dt = 0.01$.



(a) Eixo vertical em $x = 0.5$ (b) Eixo horizontal em $y = 0.5$

Figura 4.7: Eixos $x = 0.5$ e $y = 0.5$, selecionados para a avaliação dos valores das Velocidades Horizontais e Verticais na cavidade de *Lid Driven*.

Como pode ser visualizado na Figura 4.8, na análise da velocidade horizontal todos os testes convergem para os resultados descritos por Cunha [29], embora os casos de 450 passos (exceto o caso de menos número de nós por malha) desviem levemente do resultado descrito na literatura. No caso da Velocidade Vertical, o mesmo ocorre, ficando mais evidente essa discrepância dos casos com menor passo de tempo (novamente exceto o caso da malha com menor número de elementos).



(a) Velocidade Horizontal

(b) Velocidade Vertical

Figura 4.8: Comparação entre as soluções numéricas encontradas com a solução numérica descrita na literatura para a) Velocidade Horizontal e b) Velocidade Vertical na cavidade de *Lid Driven* com $Re = 10$.

Este comportamento observado para os casos com 450 passos, tanto da Velocidade Vertical quanto da Velocidade Horizontal ocorre pois, em ambos casos que não convergiram tanto quanto desejado, o tempo transcorrido após o início da análise não foi suficiente para que o escoamento atingisse condições próximas a estudada por Cunha [29]. Mas os demais testes apontam para convergência dos resultados encontrados, de modo que pode-se considerar esse caso duplamente validado, uma vez que Cunha [29], em seu trabalho, também validou seu trabalho com base em outras literaturas.

No caso de Reynolds igual a 10, os Testes 1, 2 e 4 foram os que mais se aproximaram dos valores desejados (apresentando erros entre si a partir do terceiro algarismo significativo, não mostrado neste trabalho). Considerando este fator, ainda que os resultados encontrados do Teste 4 foram satisfatórios, com uma boa qualidade de malha e com um tempo de processamento razoável, optou-se por realizar um Teste 6, para o caso de Reynolds igual a 100 apenas no caso do Teste 4. Ou seja, com uma malha de 1935 nós, 3868 elementos e 4500 passos. Os resultados do Teste 6 podem ser observados na figura 4.9

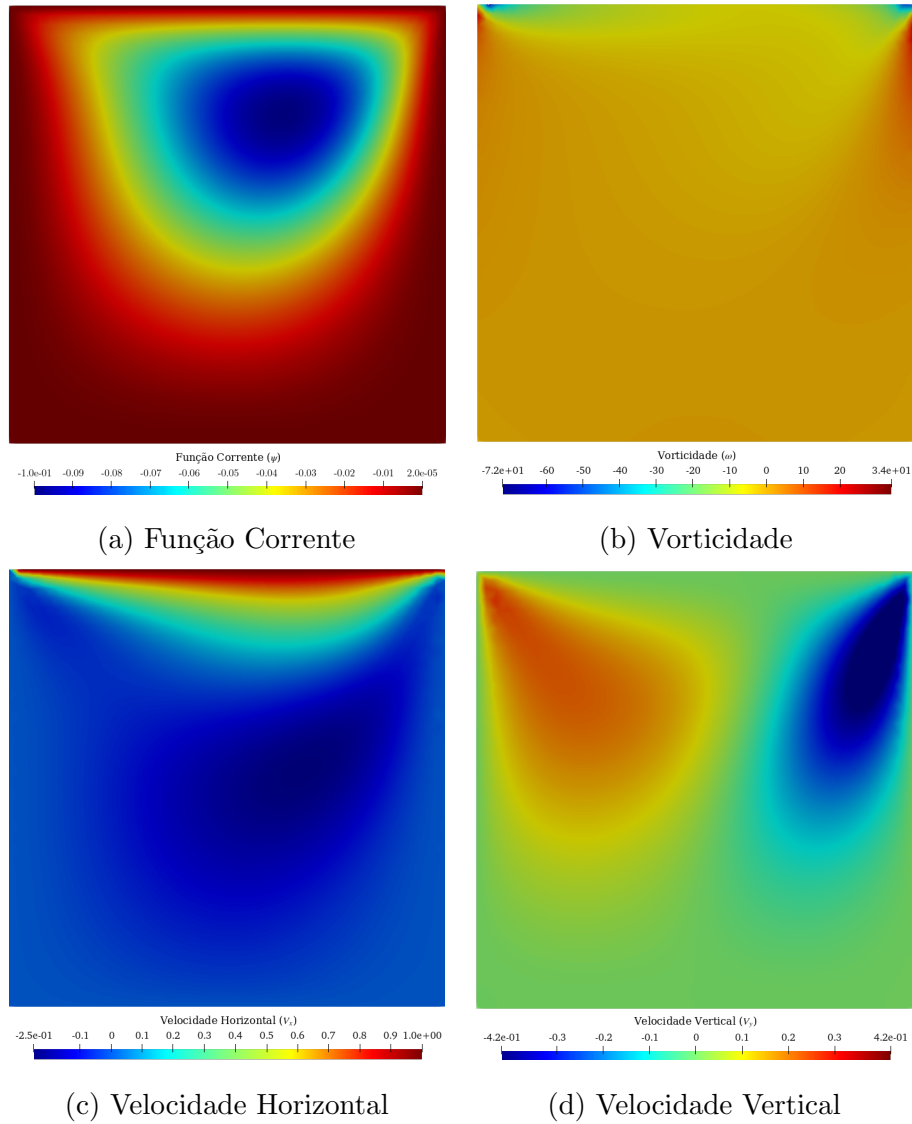


Figura 4.9: Teste 6 - Solução numérica do caso da cavidade de *Lid Driven* para $Re = 100$, malha com 1935 nós e 3868 elementos, com 4500 passos e $dt = 0.01$.

De forma análoga ao que foi feito no caso de Reynolds igual a 10, foi feita uma comparação entre os resultados obtidos e os resultados de Cunha [29]. Esta comparação é apresentada graficamente na figura 4.10.

Conforme pode ser observado, neste caso também há convergência dos resultados encontrados com os resultados descritos na literatura; logo o código também foi verificado para o caso de um escoamento na cavidade de *Lid Driven* para o número de Reynolds igual a 100.

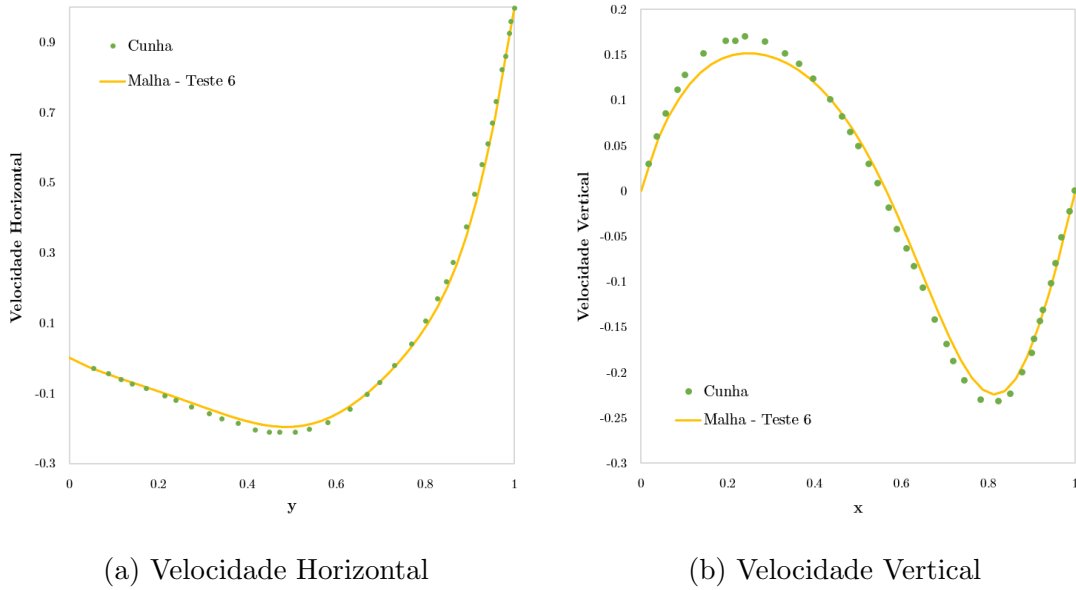


Figura 4.10: Comparação entre as soluções numéricas encontradas com a solução numérica descrita na literatura para a) Velocidade Horizontal e b) Velocidade Vertical na cavidade de *Lid Driven* com $Re = 100$.

4.2 Escoamento entre placas planas

Outro problema utilizado para a verificação do código, foi o escoamento entre placas planas. Este estudo foi importante para verificar a condição de contorno do tipo Neumann, na qual temos $\partial\psi/\partial x = 0$ e $\partial\psi/\partial y = 0$ na saída do escoamento das placas planas. Desta forma, como condição de contorno, além de Neumann na saída do escoamento (limite do lado direito do domínio), foi considerado as velocidades verticais (V_y) e horizontais (v_x) iguais a zero, exceto a velocidade horizontal de entrada do fluido (v_x do lado esquerdo do domínio) que é igual a um. A função corrente (ψ) assume valores constantes no limite inferior (c_1) e superior (c_2) do domínio, no qual $c_1 = 0$ e $c_2 = 1$. Cabe ressaltar que a condição de contorno da vorticidade é calculada dentro do código, a partir da equação 3.50.

Para a verificação deste caso, as condições de contorno utilizadas são apresentadas na figura 4.11, em duas placas de dimensão 5 no eixo x e distando 1 entre si no eixo y .

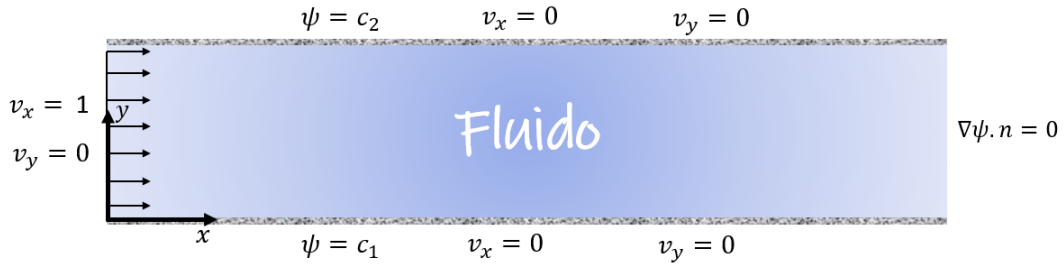


Figura 4.11: Condições de contorno do escoamento Entre Placas Planas. Fonte: Própria.

Os dados comparados foram obtidos em $x = 4$ (figura 4.12), uma vez que se deseja comparar os dados obtidos nas simulações com dados consolidados na literatura os quais se referem a um perfil parabólico de velocidade, e não mais linear, como é o caso do perfil de entrada do fluido. Também para garantir comparabilidade foi considerado um valor de $Re = 10$.



Figura 4.12: Eixo $x = 4$ selecionado para a avaliação dos valores.

Para este caso foram realizados os testes descritos na Tabela 4.2. Assim com ocorreu no caso da verificação no escoamento de *Lid Driven*, os números de nós e números de elementos são consequência de parâmetro de refino ao redor do domínio e dos obstáculos impostos no código desenvolvido para a geração de malha no *software* GMSH (descrito no apêndice A deste trabalho). Novamente, a malha gerada é uma malha não estruturada, de elementos triangulares. Para o cálculo da malha temporal, foi considerado a forma implícita *up-wind*.

Tabela 4.2: Parâmetros dos testes com diferentes malhas, passo de tempo e intervalos de tempo, para o caso de escoamento entre placas planas e com $Re = 10$.

Teste	Número de nós	Número de elementos	Passo de tempo	dt	Tempo de execução [s]
7	664	1330	450	0.01	36
8	664	1330	4500	0.01	451
9	2441	4884	450	0.01	896
10	2441	4884	4500	0.01	9833

Os resultados da Função Corrente, Vorticidade, e Velocidades Horizontal e Velocidade Vertical para os Testes 8 e 10, que são os casos com maior passo de tempo, podem ser visualizados nas figuras 4.13 e 4.14, respectivamente.

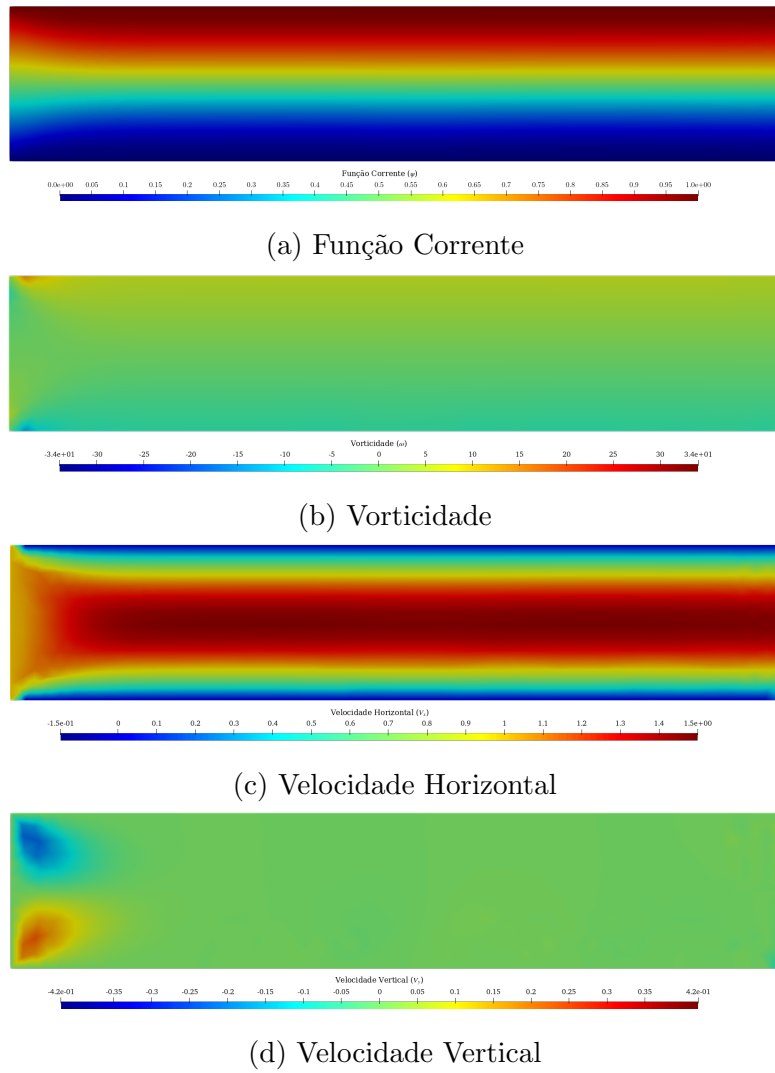


Figura 4.13: Teste 8 - Solução numérica do caso de placas paralelas, $Re = 10$, malha com 664 nós e 1330 elementos, com 4500 passos e $dt = 0.01$.

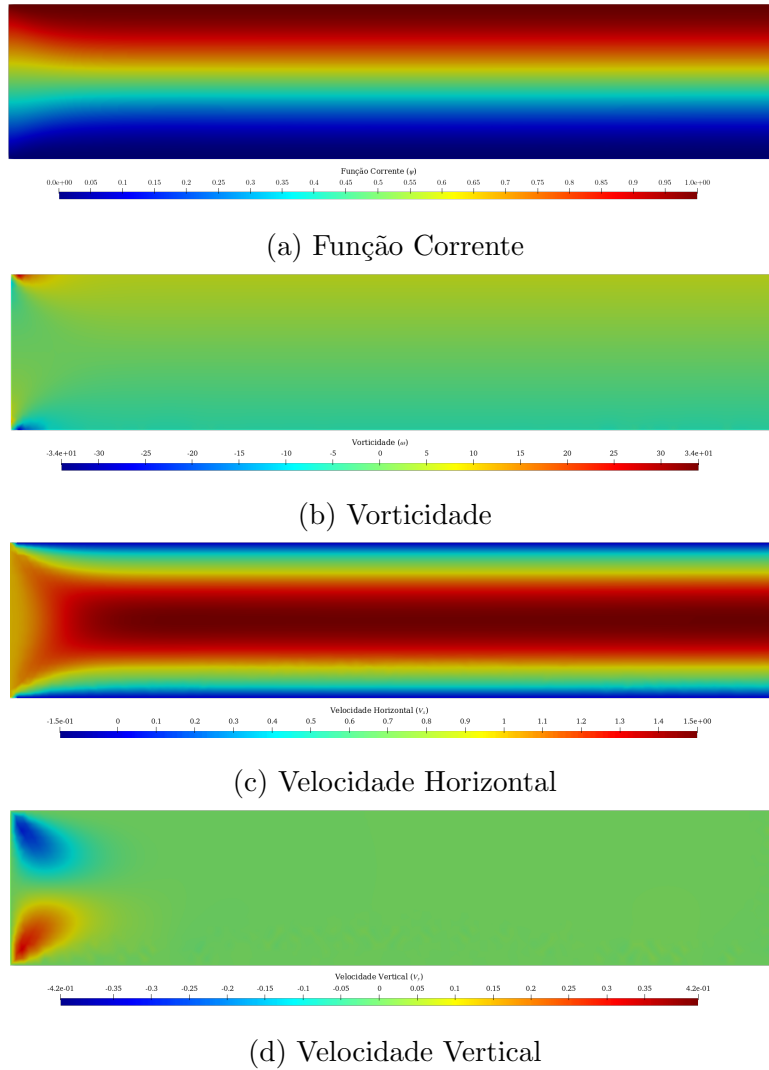


Figura 4.14: Teste 10 - Solução numérica do caso de placas paralelas, $Re = 10$, malha com 2441 nós e 4884 elementos, com 4500 passos e $dt = 0.01$.

Ao compararmos os resultados obtidos do perfil de velocidade, perfil da Função Corrente e perfil da Vorticidade com a literatura, obtemos os gráficos apresentados nas figuras 4.15, 4.16 e 4.17. Note que nos casos do perfil de Velocidade e de Vorticidade, as malhas mais refinadas se ajustaram um pouco melhor do que as malhas de menor definição.

Como podemos observar, os resultados encontrados para o caso de escoamento entre placas planas também converge com os resultados descritos na literatura. Como podemos observar, os resultados foram verificados uma vez que Cunha [29], autor utilizado como referência para comparação de dados, validou seus dados com base na solução analítica.

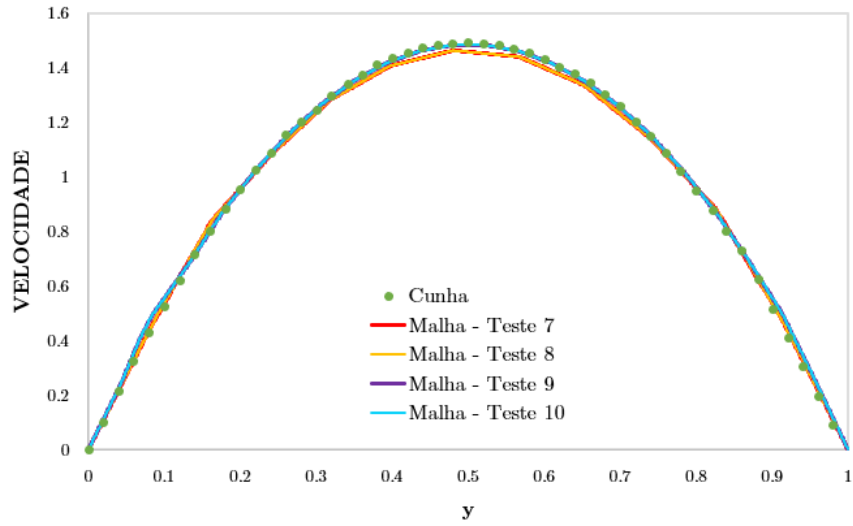


Figura 4.15: Comparação entre a solução analítica e a solução do perfil de velocidade para o caso de um escoamento entre placas planas.

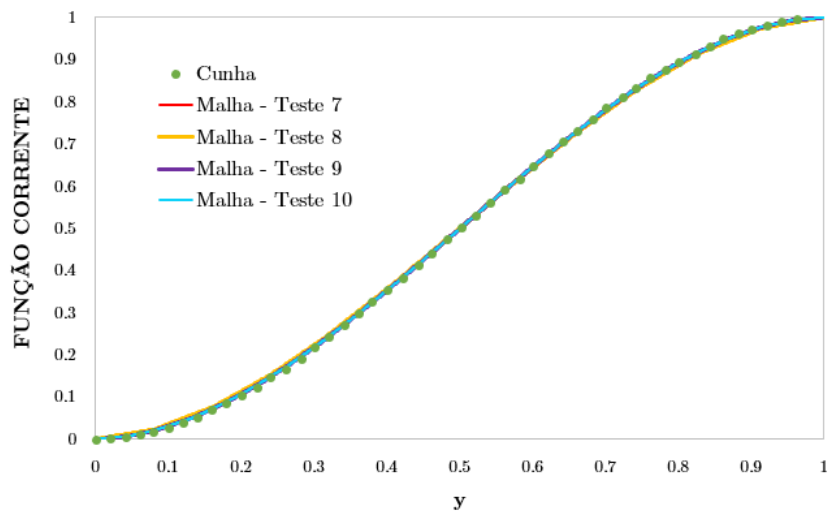


Figura 4.16: Comparação entre a solução analítica e a solução da Função Corrente para o caso de um escoamento entre placas planas.

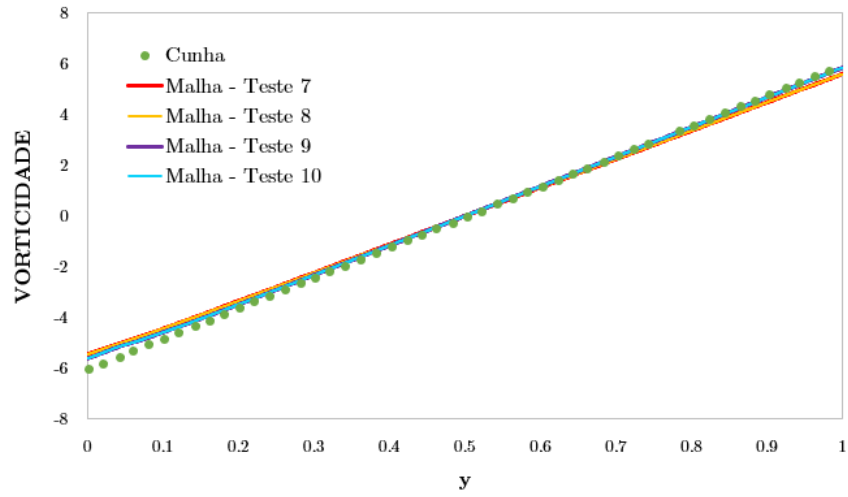


Figura 4.17: Comparação entre a solução analítica e a solução do perfil de Vorticidade para o caso de um escoamento entre placas planas.

Capítulo 5

Análise do escoamento sobre os obstáculos

Após a verificação do código desenvolvido, iniciou-se o estudo do comportamento do fluido no aquífero salino, a partir de uma simplificação do meio poroso. Foram estudados 19 casos. Os três primeiros casos com um obstáculo com a mesma geometria circular; dois casos um obstáculo de geometria elíptica, um com eixo maior horizontal e outro com eixo maior na vertical. Além disso foram estudados casos com arranjos com diferentes número de obstáculos alinhados, nas três geometrias citadas anteriormente. Foram estudados casos de dois obstáculos alinhados horizontalmente, sendo 3 casos com obstáculos circulares e dois casos com obstáculos elípticos; além disso, três casos com arranjos de obstáculos organizados três por três e; três casos com obstáculos organizados dez por três. A melhor descrição dos parâmetros de cada caso estudado pode ser observado na tabela 5.1.

Todos os casos foram simulados com o mesmo número de Reynolds, a saber $Re = 10$. Para o cálculo do número de Reynolds, foi considerado como comprimento característico o diâmetro do duto, ou seja a altura de entrada do escoamento ($y = 1$). E todos apresentam as mesmas condições de contorno, de modo que na saída do escoamento (limite a direita do domínio) seja a condição contorno de Neumann para a função corrente; nos limites inferior e superior a função corrente assuma valores constantes ($c_1 = 0$ e $c_2 = 1$) e, na entrada do domínio a função corrente é função de y ($\psi = y$). Quanto as velocidades, as velocidades verticais (v_y) e horizontais (v_x) são iguais a zero nos limites esquerdo, inferior e superior do domínio, exceto a velocidade

horizontal da entrada do fluido no domínio (limite a esquerda do domínio) no qual $v_x = 1$. Além disso nos obstáculos, tanto a velocidade vertical quanto horizontal são nulas e função corrente é função do y do centro do obstáculo.

Tabela 5.1: Parâmetros dos casos estudados.

Caso	Número de nós	Número de elementos	Passo de tempo	dt	Primeiro centro	Raio	Permeabilidade horizontal	Permeabilidade vertical	Tempo de processamento [s]
1	4347	8701	50	0.001	$x_c = 0.5$ $y_c = 0.5$	$r_x = 0.05$ $r_y = 0.05$	0.2	0.2	418
2	2459	4959	50	0.001	$x_c = 0.5$ $y_c = 0.5$	$r_x = 0.05$ $r_y = 0.05$	0.2	0.2	101
3	4347	8701	4000	0.001	$x_c = 0.5$ $y_c = 0.5$	$r_x = 0.05$ $r_y = 0.05$	0.2	0.2	33979
4	2601	5209	500	0.001	$x_c = 0.5$ $y_c = 0.5$	$r_x = 0.07$ $r_y = 0.03$	0.2	0.2	1281
5	2600	5207	500	0.001	$x_c = 0.5$ $y_c = 0.5$	$r_x = 0.03$ $r_y = 0.07$	0.2	0.2	1567
6	2852	5716	500	0.001	$x_c = 0.4$ $y_c = 0.5$	$r_x = 0.05$ $r_y = 0.05$	0.2	0.2	1804
7	2767	5564	500	0.001	$x_c = 0.425$ $y_c = 0.5$	$r_x = 0.05$ $r_y = 0.05$	0.15	0.2	1623
8	3243	6498	500	0.001	$x_c = 0.225$ $y_c = 0.5$	$r_x = 0.05$ $r_y = 0.05$	0.55	0.2	2133
9	2990	5992	500	0.001	$x_c = 0.4$ $y_c = 0.5$	$r_x = 0.07$ $r_y = 0.03$	0.2	0.2	1698
10	3136	6284	500	0.001	$x_c = 0.4$ $y_c = 0.5$	$r_x = 0.03$ $r_y = 0.07$	0.2	0.2	2160
11	3479	6980	500	0.001	$x_c = 0.4$ $y_c = 0.4$	$r_x = 0.05$ $r_y = 0.05$	0.2	0.2	2704
12	3847	7700	500	0.001	$x_c = 0.4$ $y_c = 0.4$	$r_x = 0.07$ $r_y = 0.03$	0.2	0.2	3318
13	3851	7724	500	0.001	$x_c = 0.4$ $y_c = 0.4$	$r_x = 0.03$ $r_y = 0.07$	0.2	0.2	3240
14	4892	9831	500	0.001	$x_c = 0.3$ $y_c = 0.3$	$r_x = 0.05$ $r_y = 0.05$	0.2	0.2	5702
15	5600	11247	500	0.001	$x_c = 0.3$ $y_c = 0.3$	$r_x = 0.07$ $r_y = 0.03$	0.2	0.2	8087
16	5570	11187	500	0.001	$x_c = 0.3$ $y_c = 0.3$	$r_x = 0.03$ $r_y = 0.07$	0.2	0.2	7928
17	6894	13940	1000	0.001	$x_c = 0.2$ $y_c = 0.15$	$r_x = 0.05$ $r_y = 0.05$	0.05	0.05	27978
18	6717	13586	1000	0.001	$x_c = 0.2$ $y_c = 0.15$	$r_x = 0.07$ $r_y = 0.03$	0.05	0.05	27440
19	6711	13574	1000	0.001	$x_c = 0.2$ $y_c = 0.15$	$r_x = 0.03$ $r_y = 0.07$	0.05	0.05	27243

As malhas foram criadas a partir do *software* GMSH através de um código desenvolvido (e que pode ser visualizado no apêndice A deste trabalho). Os números de nós e números de elementos são consequência de parâmetro de refino ao redor

do domínio e dos obstáculos impostos. Neste trabalho, em todos os casos foram utilizadas malhas não estruturadas de elementos triangulares. Além disso, todos os casos tiveram como refino igual a 0.03 ao redor do domínio e 0.01 ao redor do obstáculo. Na tabela 5.1, x_c e y_c são as posições, nos eixos x e y , respectivamente, do centro do primeiro obstáculo a ser inserido no código. Na descrição do raio dos obstáculos, r_x versa sobre o tamanho do raio horizontal do obstáculo, enquanto r_y é o tamanho vertical do raio. A permeabilidade é o tamanho imposto entre os centros dos obstáculos; desta forma, a permeabilidade horizontal é a distância horizontal (no eixo x) entre o centro dos obstáculos, enquanto a permeabilidade vertical é a distância vertical (no eixo y) entre os centros dos obstáculos. Para o cálculo da malha temporal, foi considerado a forma implícita, descrita no capítulo 3.

5.1 Escoamento ao redor de um obstáculo

5.1.1 Caso 1 - obstáculo circular em domínio retangular

O Caso 1 foi realizado com um obstáculo circular no meio do domínio, conforme pode ser visualizado na figura esquemática 5.1, que também mostra as condições de contorno utilizadas no problema. Como mencionado anteriormente, foram considerados $c_1 = 0$ e $c_2 = 1$. Cabe ressaltar, mais uma vez, que a condição de contorno da vorticidade é calculada dentro do código, considerando a equação 3.50. Observa-se, ainda, que a velocidade é informada em $[m/s]$ em todos os casos.

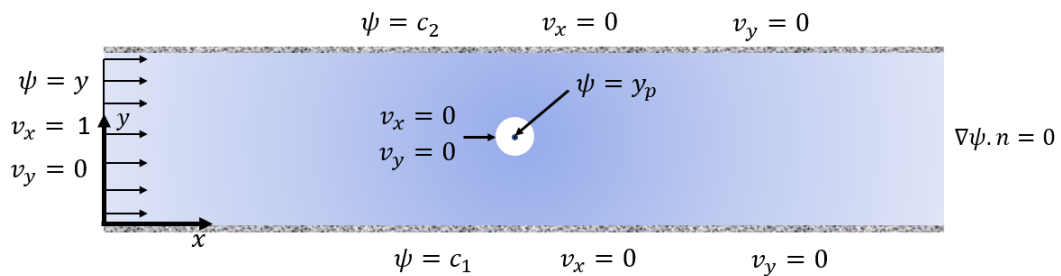
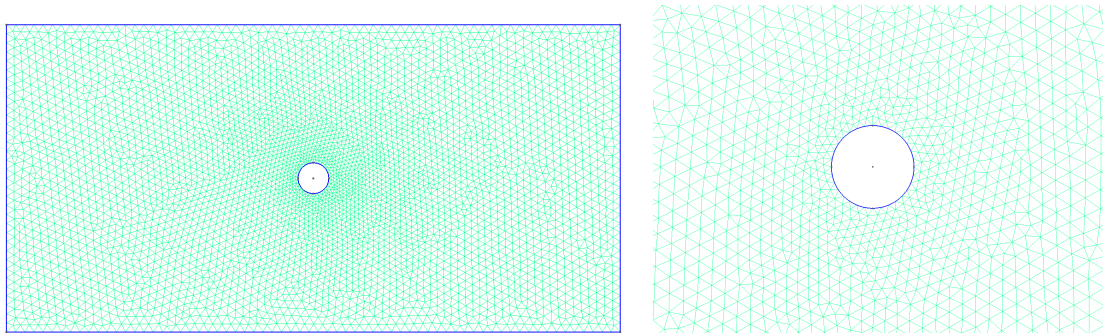


Figura 5.1: Desenho esquemático com condições de contorno.

Inicialmente testou-se um domínio retangular comprimento 2 e altura 1 ($x = 2$ e $y = 1$) e com um obstáculo de circunferência de 0.05 de raio, centrada em x e em y . Além disto a malha foi desenhada de tal forma que tivesse elementos triangulares

não estruturados e que ao redor do obstáculo circular houvesse mais definição do que nos limites do domínio, ou seja, que as arestas dos elementos triangulares fossem menores ao redor do obstáculo do que dos limites do domínio (figura 5.2). Esta malha testada apresenta 4347 nós e 8701 elementos, e o código demorou 418 segundos para executar um total de 50 passos. O resultado final da Função Corrente, Vorticidade, Velocidade Horizontal e Velocidade Vertical para esta situação pode ser visualizado na figura 5.3.



(a) Malha com domínio retangular

(b) Detalhe da malha ao redor do obstáculo

Figura 5.2: a) Malha de domínio retangular com elementos triangulares e não estruturada, com 4347 nós e 8701 elementos e b) detalhe da malha ao redor do obstáculo.

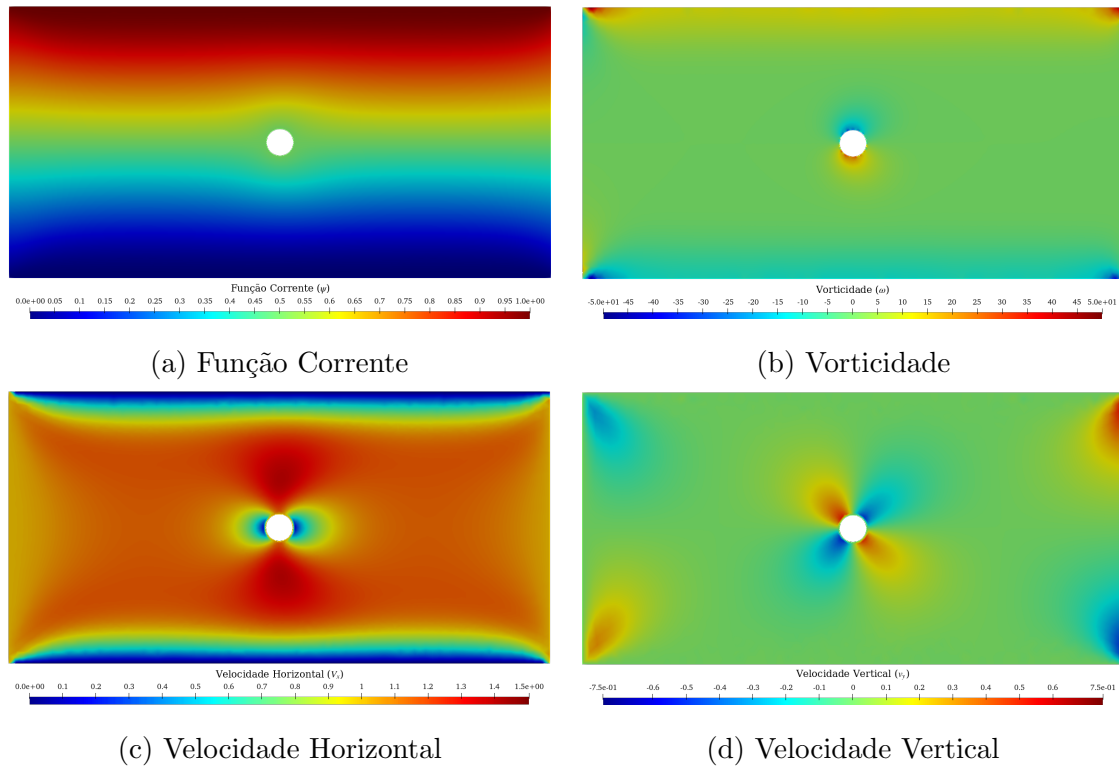


Figura 5.3: Solução numérica para o Caso 1 - Escoamento com um obstáculo circular, $Re = 10$, domínio retangular, com malha com 4347 nós e 8701 elementos, com 50 passos e $dt = 0.001$.

5.1.2 Caso 2 - obstáculo circular em domínio quadrada

Para tentar otimizar as próximas análises, foi testado o mesmo problema, mas agora com um domínio quadrado de 1 por 1 ($x = 1$ e $y = 1$) de obstáculo de circular de mesma dimensão que a anterior e também centrada (figura 5.5a). Isto fez com que a malha, com os mesmos critérios, reduzisse de 4347 nós para 2459 nós e de 8701 para 4959 elementos (figura 5.4).

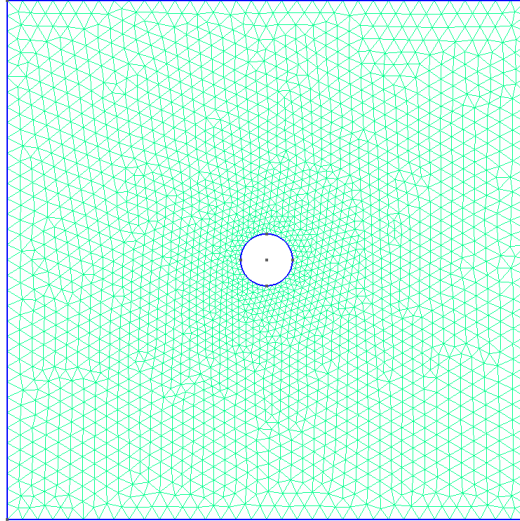


Figura 5.4: Malha de domínio quadrado com elementos triangulares e não estruturada, com 2459 nós e 4959 elementos.

Após 50 passos, o tempo de processamento foi de 418 segundos para 101 segundos. Contudo, comparando-se os resultados gerados pelo domínio retangular e quadrado 0.25 a jusante e 0.25 a montante do centro do obstáculo, notou-se que os resultados não convergem, pois o espaço que o fluido tem para percorrer, e consequentemente se desenvolver, antes de encontrar o obstáculo influencia no campo de velocidades (figura 5.6). Portanto, a geometria do domínio influencia nos resultados obtidos.

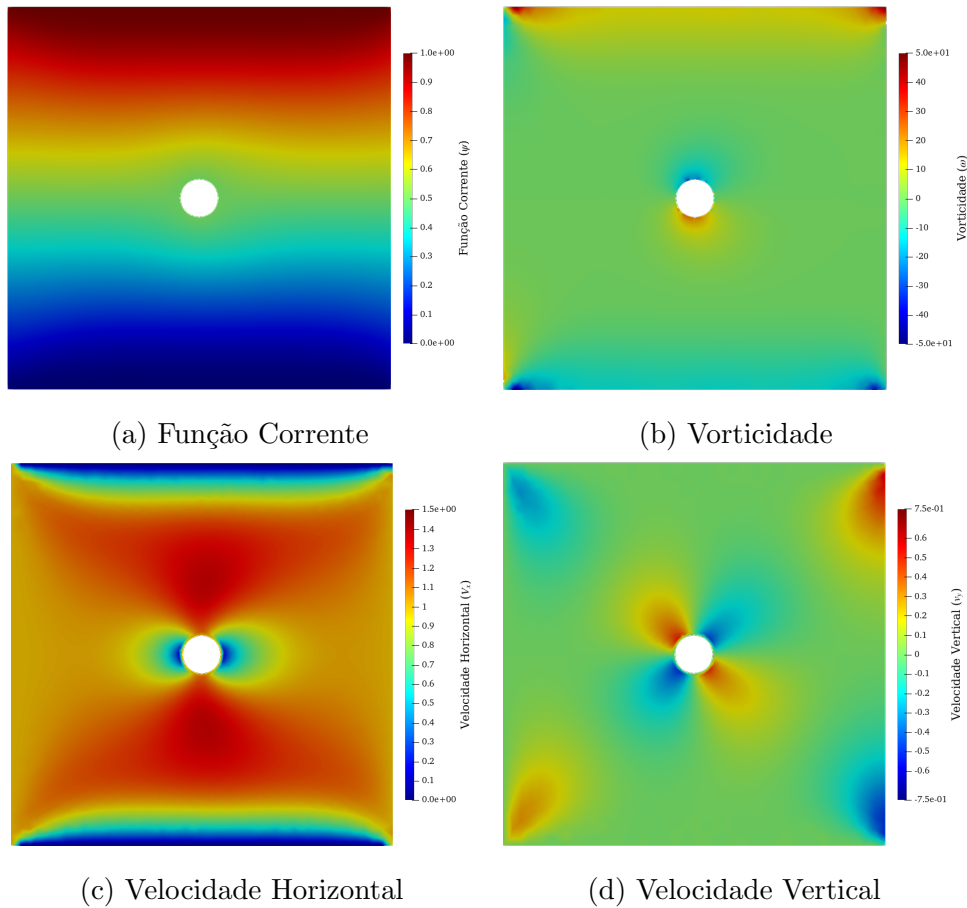


Figura 5.5: Solução numérica para um escoamento com um obstáculo circular, $Re = 10$, domínio quadrado, com malha com 2459 nós e 4959 elementos, com 50 passos e $dt = 0.001$.

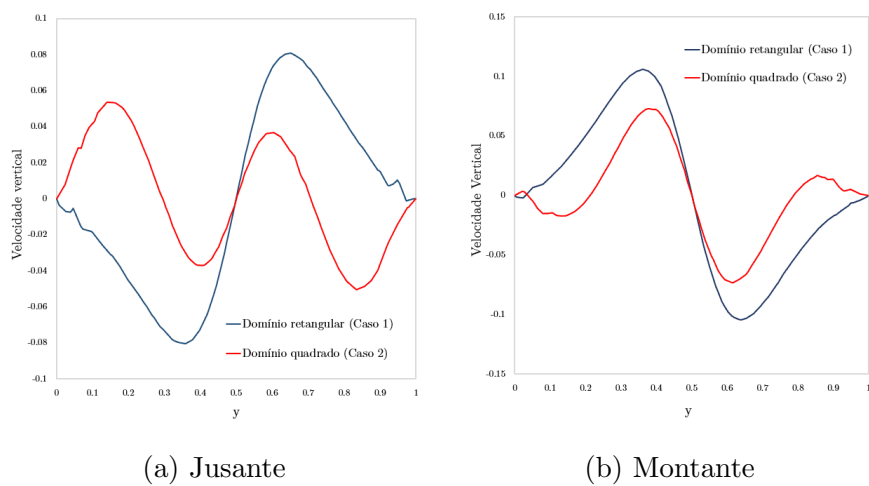


Figura 5.6: Gráfico comparativo dos resultados do domínio retangular (Caso 1) e quadrado (Caso 2) nos eixos a 0.25 a jusante e a montante do obstáculo.

5.1.3 Caso 3 - Influência do passo de tempo

A seguir, realizou-se o teste com o mesmo domínio retangular utilizada no Caso 1, mas aumentando o número total de passos para 4000. No total foram necessárias 33979 segundos para finalizar a simulação. Os resultados após 4000 passos são apresentados na figura 5.7.

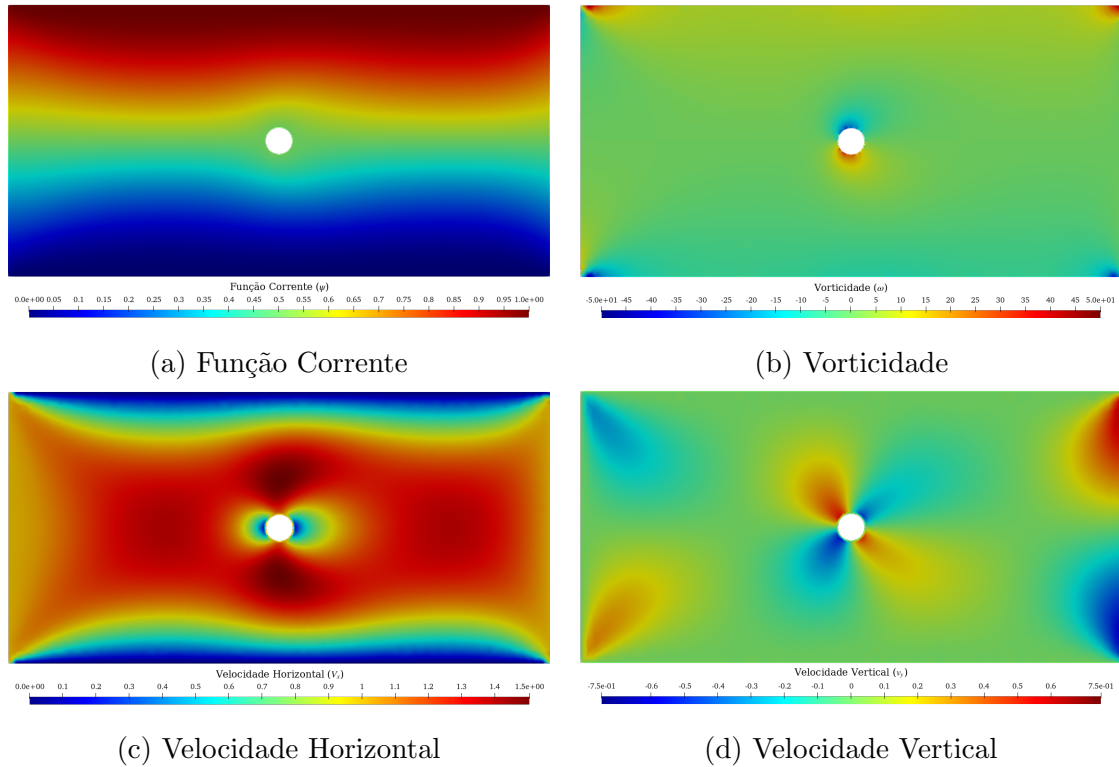


Figura 5.7: Solução numérica para um escoamento com um obstáculo circular, $Re = 10$, domínio retangular, com malha com 4347 nós e 8701 elementos, com 4000 passos e $dt = 0.001$.

Analisando os valores da Função Corrente, Vorticidade e Velocidades nos mesmos pontos ($x = 0.75$ e $x = 1.25$) da malha em diferentes tempos de iteração (50, 500 e 4000), foi possível notar que 50 passos não eram suficientes para estabelecer um resultado estável, ou seja para que o escoamento atingisse o regime permanente da simulação, onde não há mais alteração numérica significativa nos resultados obtidos. Mas os resultados com 500 passos e 4000 passos foram semelhantes. Note que nos gráficos da figura 5.8 as linhas nas cores verde e cinza, que se referem aos casos de 500 passos não aparecem. Isto porque elas estão sobrepostas pelas linhas de cores vermelha e roxa, referentes aos casos com 4000 passos. Desta forma, o passo de

tempo também afeta os resultados, mas 500 passos foram suficientes para que o regime permanente tenha sido alcançado (figura 5.8). Um resultado importante, dado que o tempo de execução das simulações está fortemente relacionado ao passo de tempo a serem realizadas.

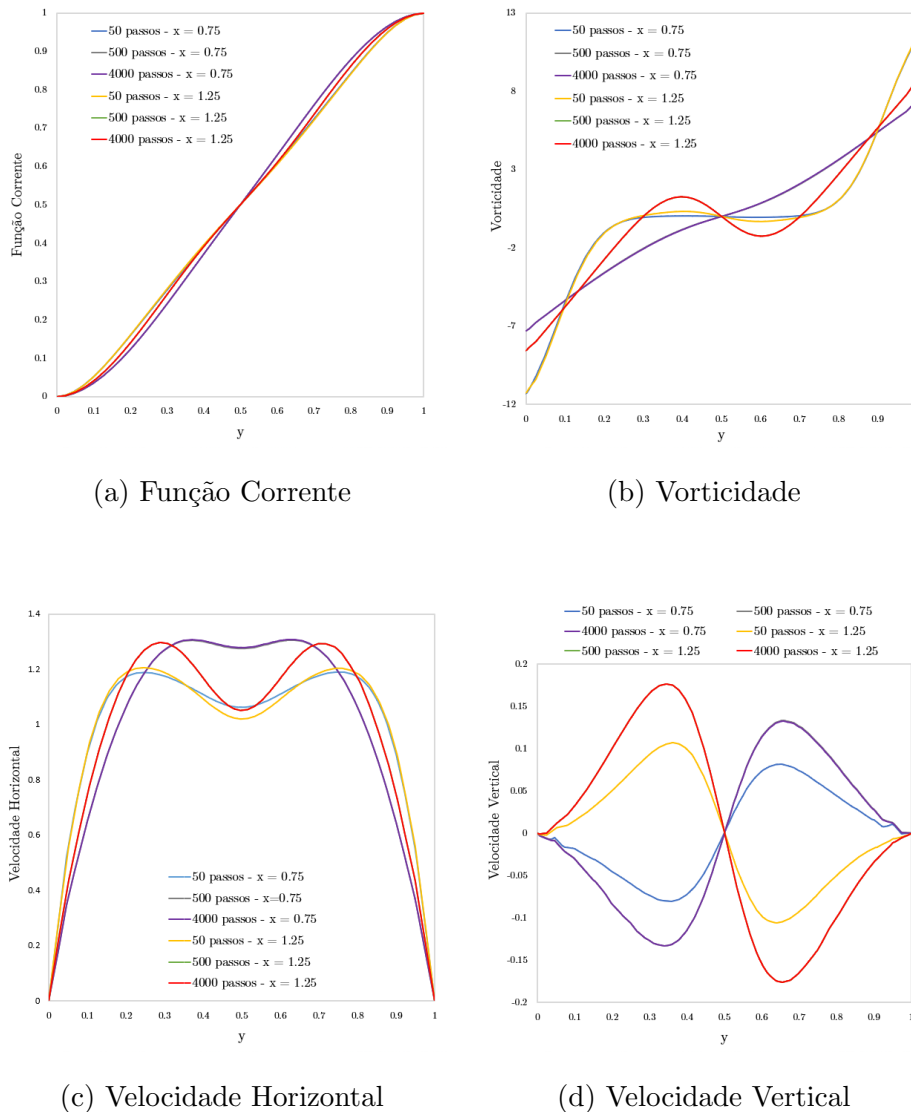
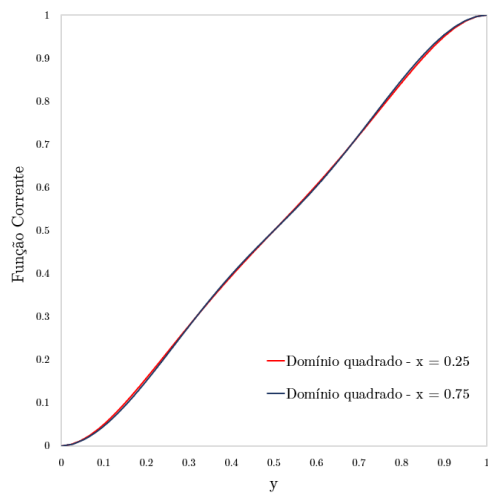


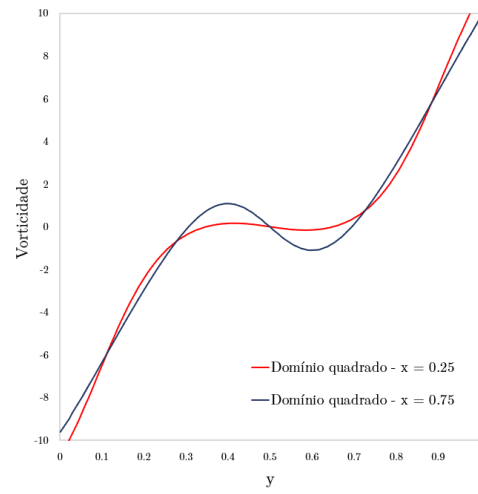
Figura 5.8: Comparação dos valores da Função Corrente, Vorticidade, e Velocidade Horizontal e Vertical no domínio retangular 0.25 antes a jusante do obstáculo ($x = 0.75$) e 0.25 a montante ($x = 1.25$) do obstáculo, de acordo com o passo de tempo.

Em seguida, para verificar o escoamento após seu desenvolvimento foi simulado no domínio quadrado com 500 passos (este valor de passos foi escolhido considerando os resultados encontrados anteriormente sobre a influência no passo de tempo nos resultados). Assim como ocorreu no caso do domínio retangular, os efeitos do

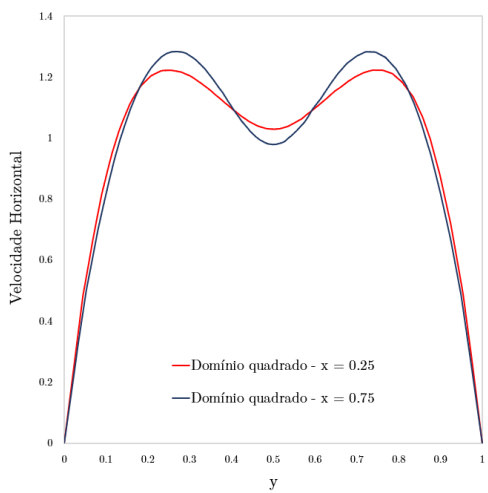
obstáculo podem ser observados quando comparamos os parâmetros a 0.25 a jusante e a 0.25 a montante do obstáculo, principalmente na Velocidade Vertical (figura 5.9).



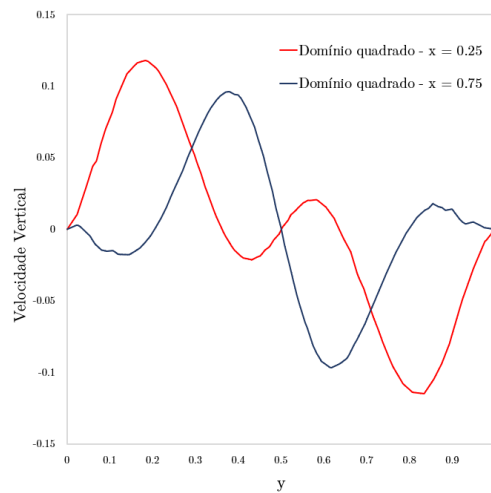
(a) Função Corrente



(b) Vorticidade



(c) Velocidade Horizontal



(d) Velocidade Vertical

Figura 5.9: Comparação dos valores da Função Corrente, Vorticidade, e Velocidade Horizontal e Vertical no domínio quadrado antes ($x = 0.25$) e depois ($x = 0.75$) do obstáculo, com 500 passos e $dt = 0.001$.

5.1.4 Caso 4 - obstáculo elíptico com maior eixo na horizontal

Após o estudo do escoamento ao redor de um obstáculo circular, optou-se por estudar a influência da geometria do obstáculo. Considerando os resultados satisfatórios encontrados com a utilização do refino da malha e passo de tempo do último teste realizado, partiu-se então para a simulação de um caso de um escoamento ao redor de um obstáculo com geometria elíptica com seu maior eixo na horizontal. A elipse foi centrada em x e y e tem o seu maior raio valor igual a 0.07 e o menor raio valor de 0.03. O refino permaneceu o mesmo (com os mesmos valores de tamanho das arestas dos triângulos ao redor do obstáculo e dos limites do domínio) e o domínio utilizado foi o domínio quadrado de 1×1 , dado o menor tempo necessário para o processamento, resultando em uma malha de 2601 nós e 5209 elementos. Os demais parâmetros permaneceram igual ao caso anterior ($Re = 10$, $dt = 0.001$, 500 passos). Os resultados finais são apresentados na figura 5.10.

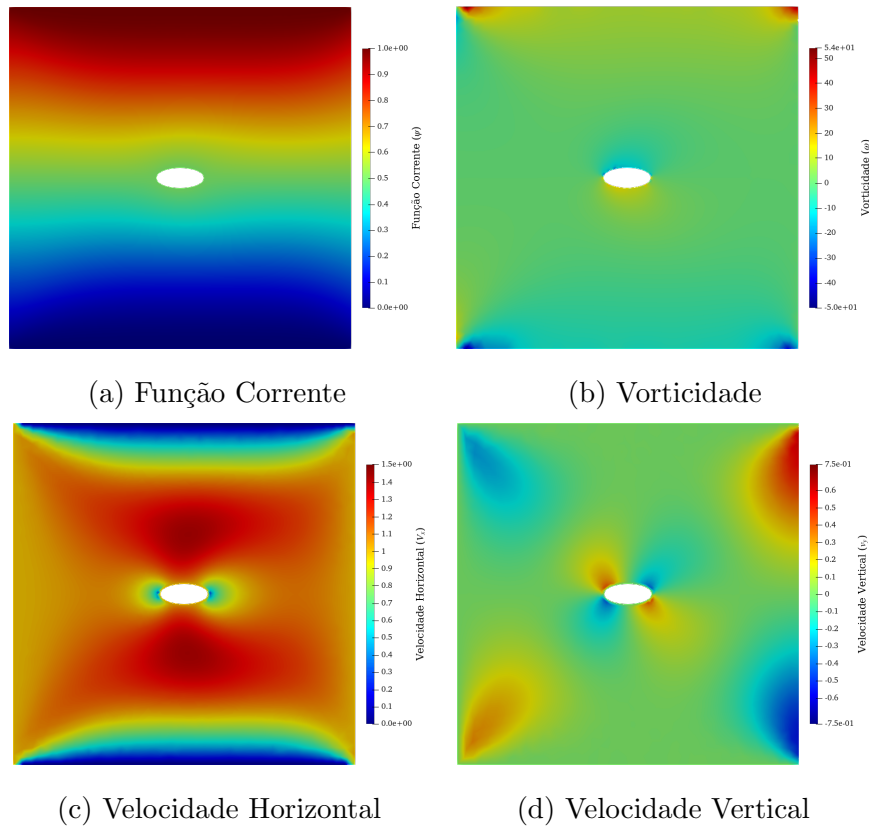


Figura 5.10: Solução numérica para um escoamento com um obstáculo elíptico com maior eixo na horizontal, $Re = 10$, domínio quadrado, com malha com 2601 nós e 5209 elementos, com 500 passos e $dt = 0.001$.

5.1.5 Caso 5 - obstáculo elíptico com maior eixo na vertical

Para finalizar o estudo do escoamento ao redor de um único obstáculo, mudando apenas a geometria do obstáculo foram invertidos os eixos maior e menor, mantendo-se os mesmos valores de refino e de raios utilizados no caso anterior e consequentemente alterando ligeiramente o número de nós (2600) e elementos (5207) da malha. Desta forma, realizou-se uma simulação de um escoamento ao redor de um obstáculo com geometria elíptica com maior eixo na vertical. Os resultados finais são apresentados a seguir, na figura 5.11.

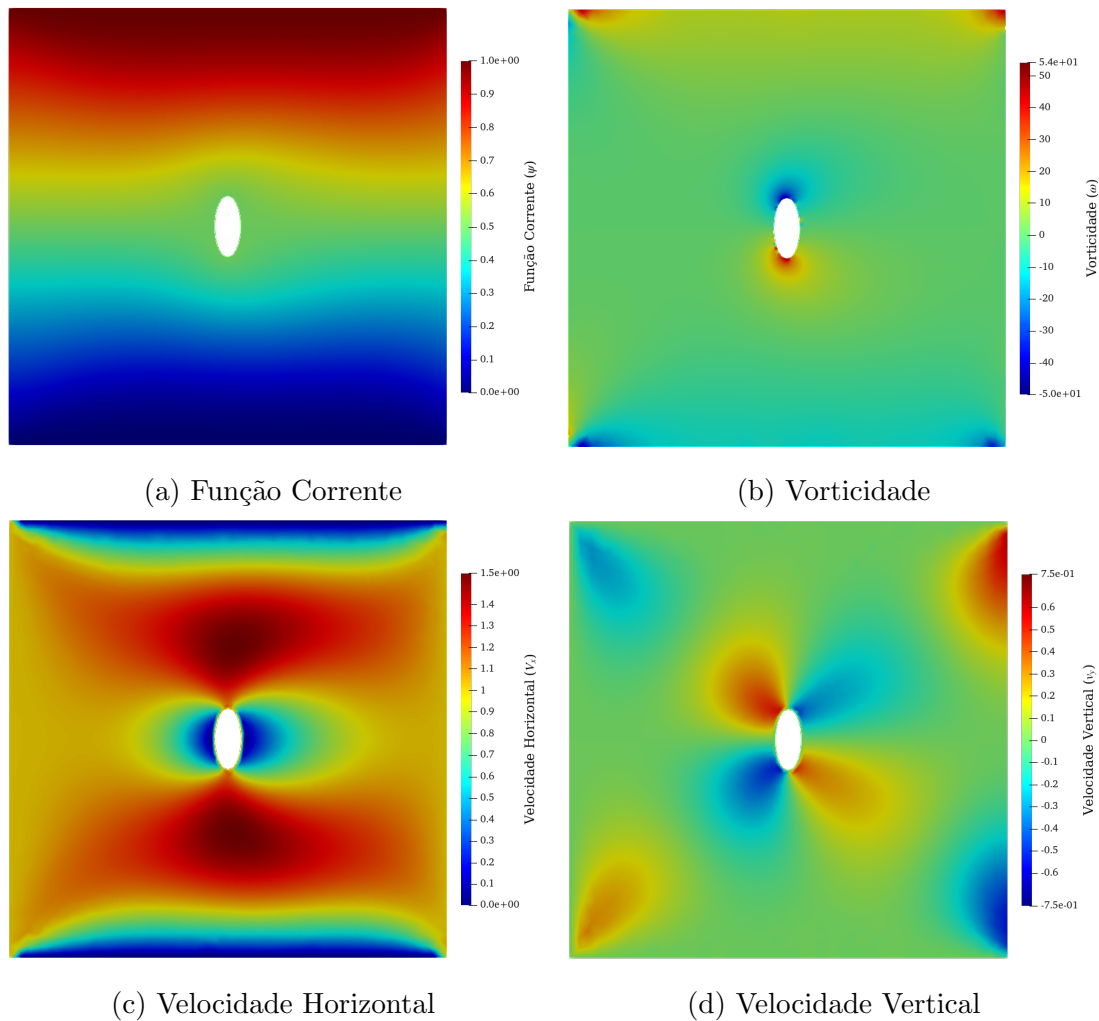


Figura 5.11: Solução numérica para um escoamento com um obstáculo elíptico com maior eixo na vertical, $Re = 10$, malha quadrada com 2600 nós e 5207 elementos, com 500 passos e $dt = 0.001$.

Comparando as Velocidades Verticais dos casos de obstáculo circular e nas geometrias elípticas pode-se notar a influência da geometria do obstáculo no escoamento (figura 5.12). No caso da elipse com maior eixo na vertical é interessante notar como a variação da velocidade Vertical aumenta após o escoamento passar pelo obstáculo (eixo $x = 0.75$); e que o oposto ocorre no caso do obstáculo de elipse horizontal que tem o valor da Velocidade Vertical diminuído após passar pelo obstáculo. Isso ocorre devido ao alargamento ou estreitamento da passagem pela qual o fluido escore, aumentando ou diminuindo a velocidade.

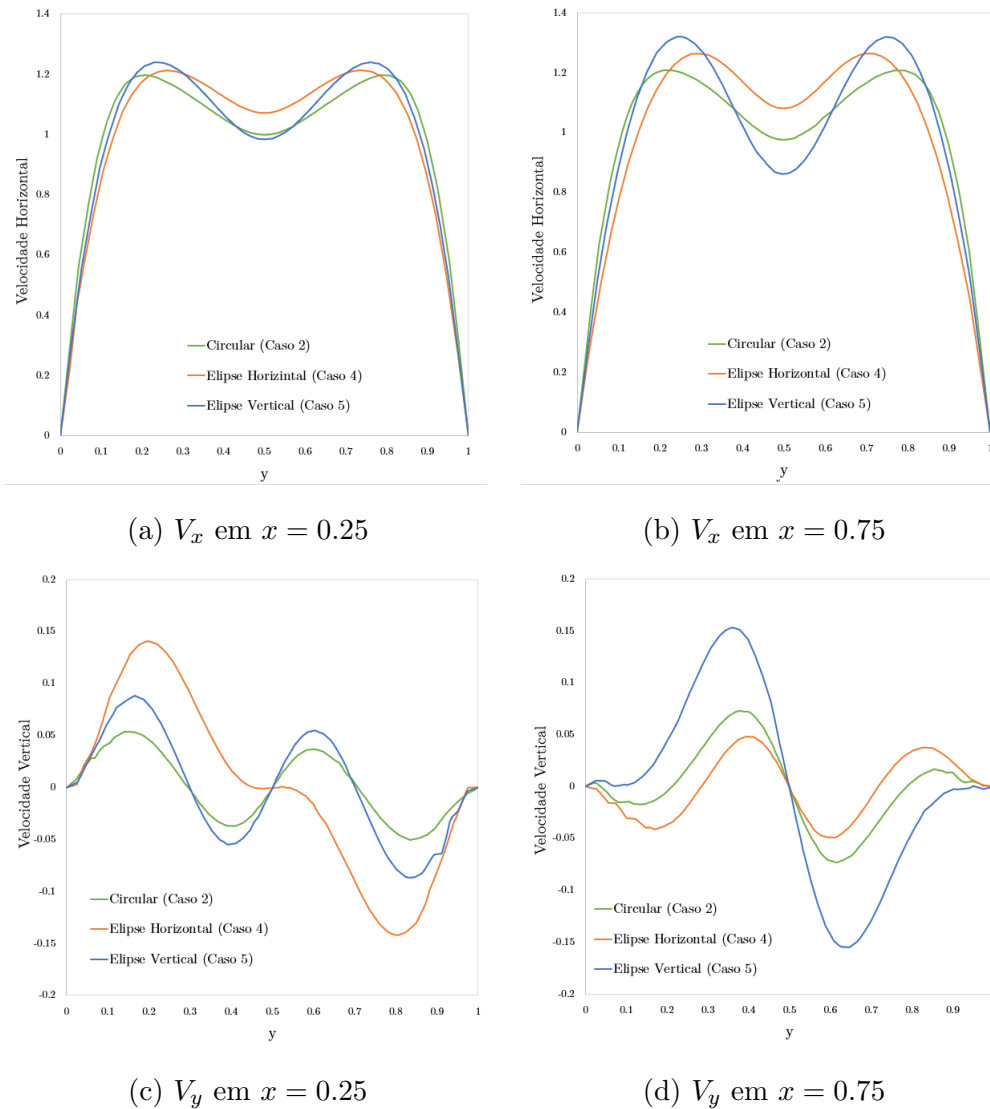


Figura 5.12: Gráfico comparativo da Velocidade Horizontal (a e b) e vertical (c e d) entre os casos de obstáculo circular (Caso 2), obstáculo elíptico com maior eixo na horizontal (Caso 4) e obstáculo elíptico com maior eixo na vertical (Caso 5).

5.2 Escoamento ao redor de dois alinhados horizontalmente

Após o estudo simplificado de um meio poroso com apenas um obstáculo, foi estudado uma configuração um pouco mais complexa, para compreender a possível interação entre obstáculos no escoamento. Iniciou-se, um estudo com dois obstáculos alinhados horizontalmente (no eixo y). As condições de contorno permanecem iguais as expostas na figura 5.1, mas todos os testes realizados para a análise da influência

dos dois obstáculos alinhados horizontalmente utilizaram o domínio quadrado. As condições de contorno podem ser visualizadas no desenho esquemático da figura 5.13. Novamente, a condição de contorno da vorticidade é calculada dentro do código, a partir da equação 3.50.

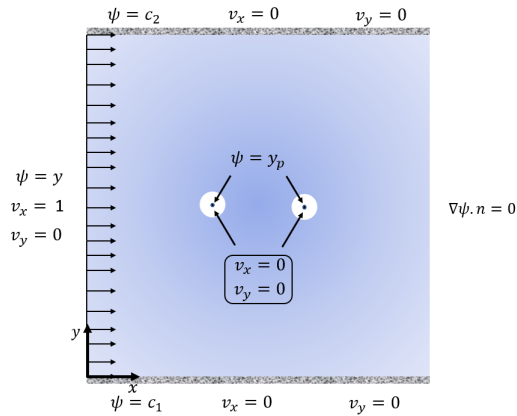


Figura 5.13: Desenho esquemático com condições de contorno com dois obstáculos alinhados horizontalmente.

5.2.1 Caso 6 - Dois obstáculos circulares

Assim como ocorreu na seção 5.1, ao estudar os caso com dois obstáculos alinhados horizontalmente, inicialmente foi simulada a situação para dois obstáculos de geometria circular. Com base nos resultados encontrados anteriormente, optou-se por testar um escoamento em um domínio quadrado de dimensões 1 em x e y , com $Re = 10$ e; com o mesmo refino de malha ao redor dos obstáculos e domínio. Além disso, os obstáculos foram centrados no eixo y e tem raio de 0.05. A distância horizontal entre os centros dos obstáculos (permeabilidade horizontal), bem como os demais parâmetros alterados entre cada teste simulado encontram-se descritos na tabela 5.2.

Tabela 5.2: Parâmetros dos casos com diferentes malhas, passo de tempo, posições dos centros e permeabilidades horizontais para o caso de dois obstáculos circulares em sequência.

Caso	Número de Nós	Número de Elementos	Passo de tempo	Primeiro Centro	Permeabilidade Horizontal	Tempo [s]
6	2852	5716	500	$x = 0.4$	0.2	1804
7	2767	5564	500	$x = 0.425$	0.15	1623
8	3243	6498	500	$x = 0.225$	0.55	2133

Os resultados para a Função Corrente, Vorticidade, Velocidade Horizontal e Velocidade Vertical após 500 passos com $dt = 0.001$ são apresentados para o Caso 6, considerado o teste com a distância de referência (ou padrão), na figura 5.14.

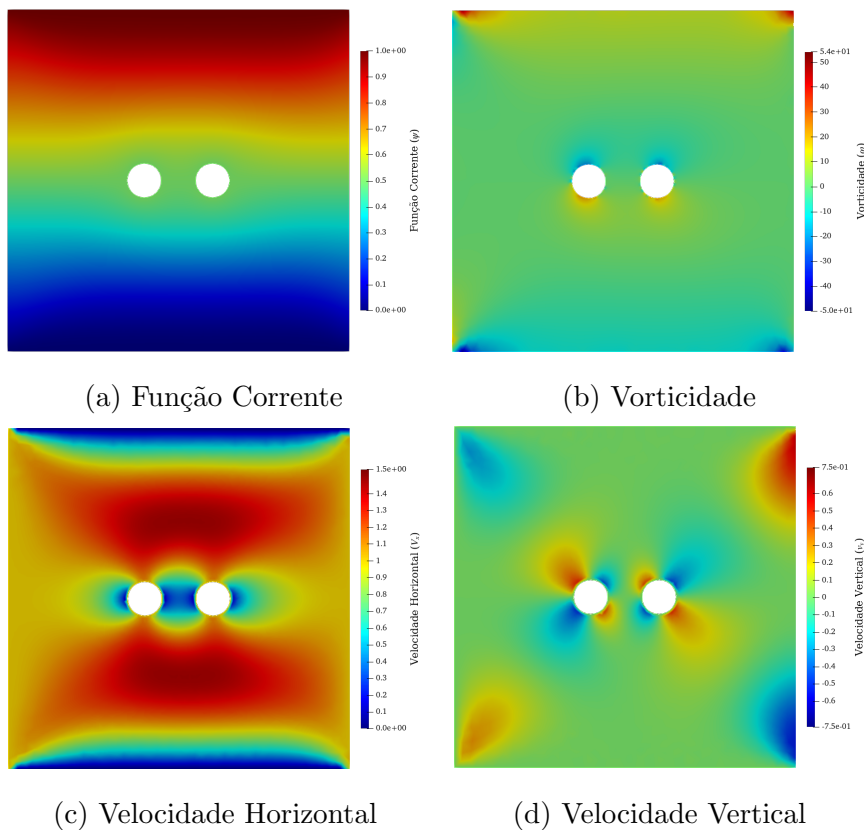


Figura 5.14: Solução numérica para um escoamento com dois obstáculos circulares alinhados horizontalmente, $Re = 10$, domínio quadrado, malha com 2852 nós e 5716 elementos, 500 passos e $dt = 0.001$.

5.2.2 Casos 7 e 8 - Influência da distância entre obstáculos

Estudou-se, também a influência da distância entre os obstáculos para o caso de obstáculos circulares. Para tal, foi simulado uma malha semelhante a anterior, ainda em um domínio quadrado, com os obstáculos mais próximos entre si (Caso 7 da tabela 5.2) e um caso com os obstáculos mais distantes entre si (Caso 8 da tabela 5.2). Nas figuras 5.15 a 5.18 são apresentados os resultados dos Casos 7 e 8 e o respectivo gráfico comparativo entre os três Casos (6, 7 e 8) de cada uma das variáveis (Função Corrente, Vorticidade, Velocidade Horizontal e Velocidade Vertical).

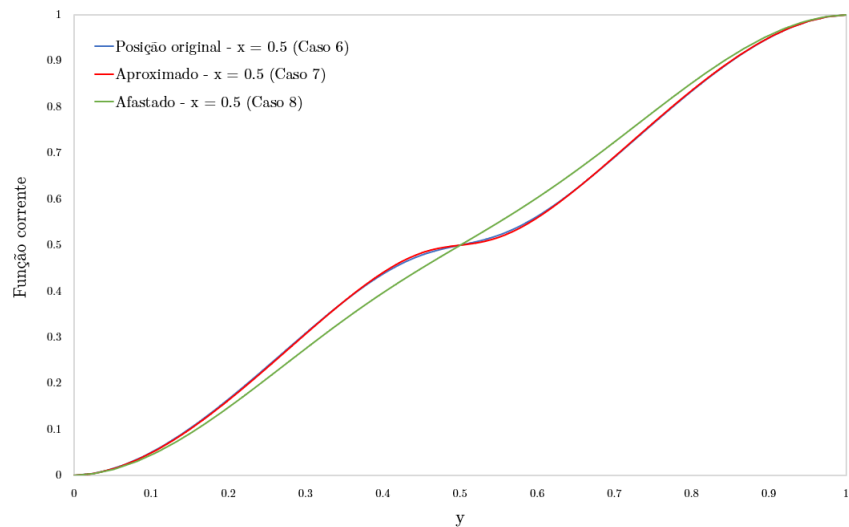
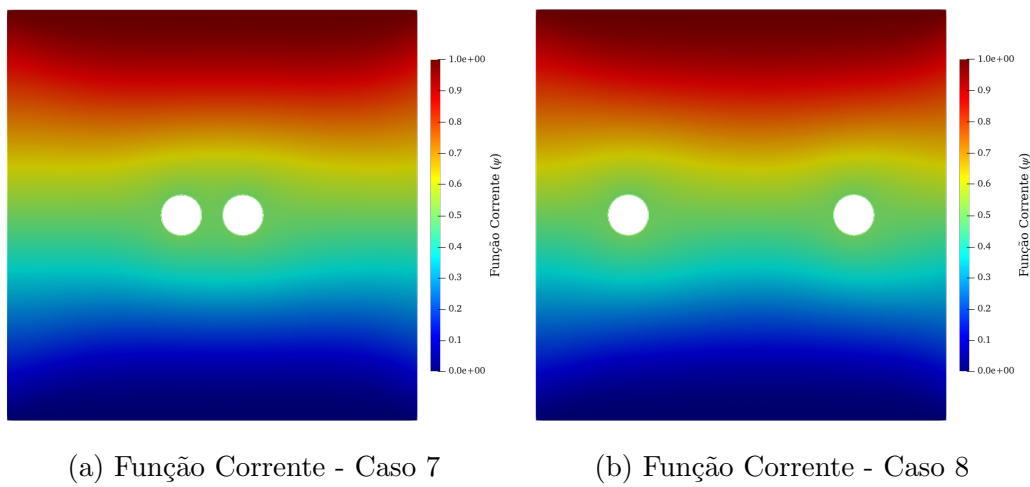
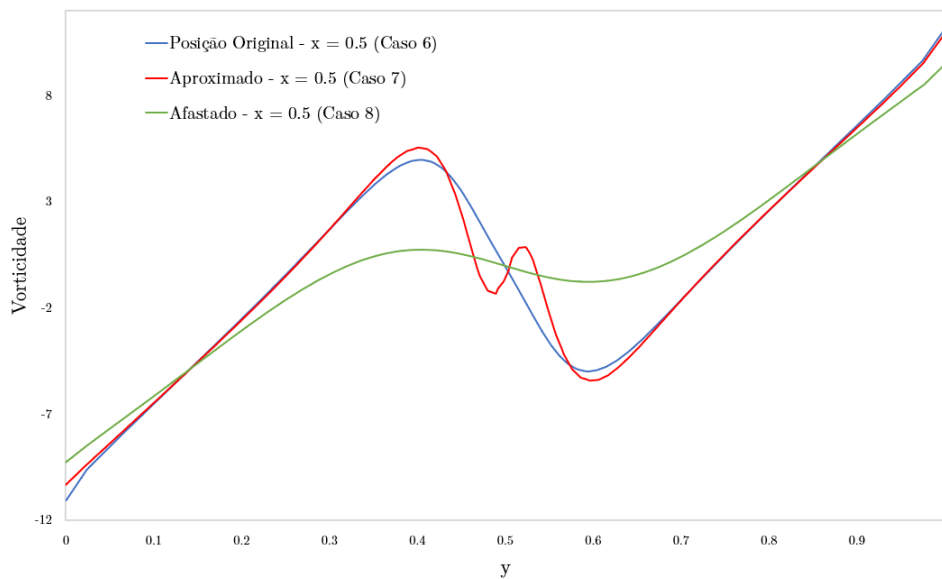
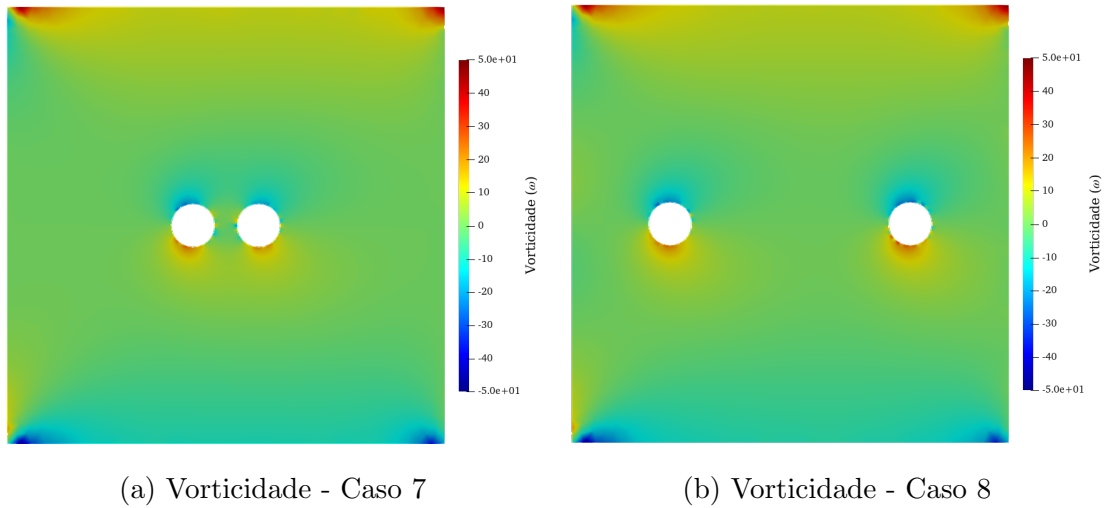


Figura 5.15: Solução numérica da Função Corrente para os casos (a) Caso 7, (b) Caso 8 e (c) gráfico comparativo da solução da Função Corrente para os 3 casos, obtidos no eixo $x = 0.5$.

Quanto a da função corrente, podemos observar que a maior diferença ocorreu no caso do Caso 8, no qual os obstáculos encontram-se mais afastados entre si. Neste caso, o resultado encontrados se assemelha ao resultado encontrado no caso com apenas um obstáculo. Ou seja, os obstáculos estão afastados suficientemente entre si para existir diferença pouco significativa entre o Caso 8 e o caso da seção 5.1, Caso 3 (figura 5.8).

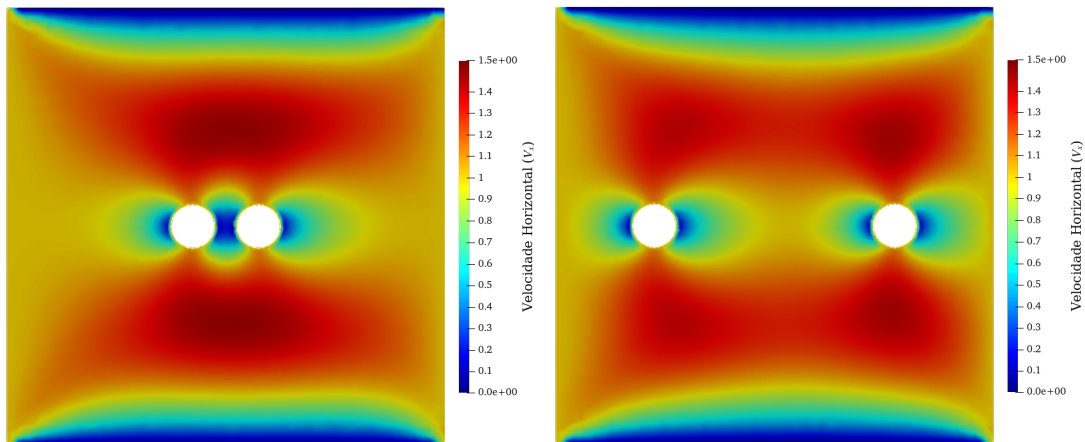


(c) Gráfico comparativo da Vorticidade entre os Casos 6, 7 e 8.

Figura 5.16: Solução numérica da Vorticidade para os casos (a) Caso 7, (b) Caso 8 e (c) gráfico comparativo da solução da Vorticidade para os 3 casos, obtidos no eixo $x = 0.5$.

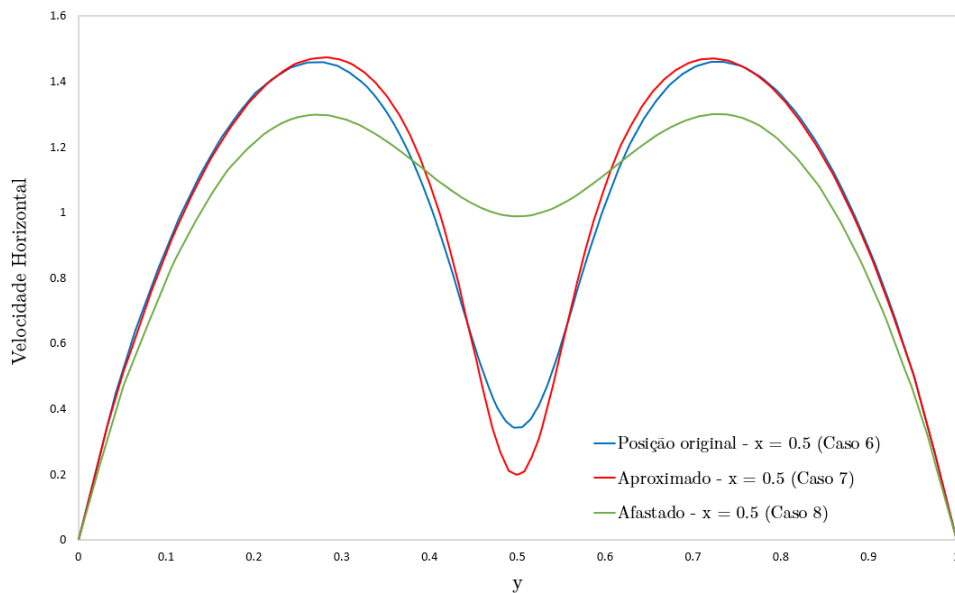
Ao observamos a figura 5.16c notamos que a aproximação dos obstáculos tem

forte influência sobre a Vorticidade, tendendo a se comportar cada vez mais com o comportamento no caso de apenas um único obstáculo (figura 5.8) a medida que os obstáculos se afastam.



(a) Velocidade Horizontal - Caso 7

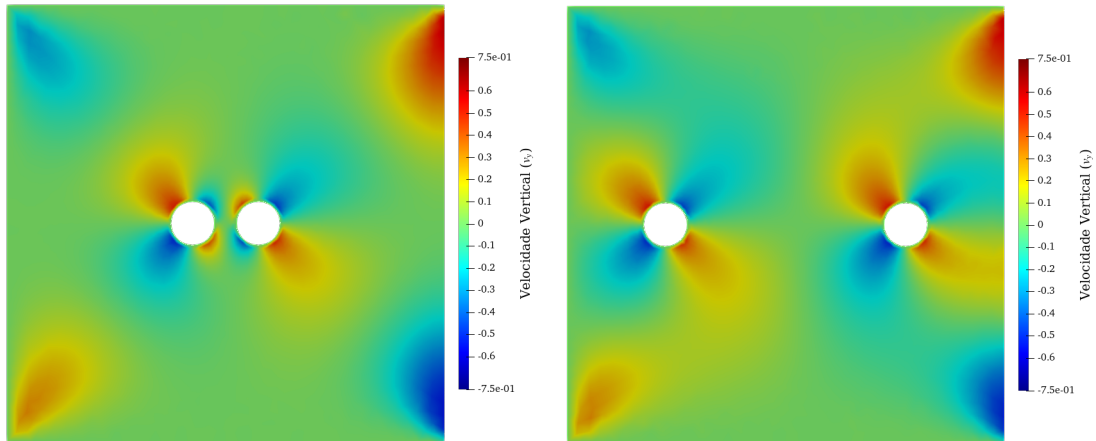
(b) Velocidade Horizontal - Caso 8



(c) Gráfico comparativo da Velocidade Horizontal entre os Casos 6, 7 e 8.

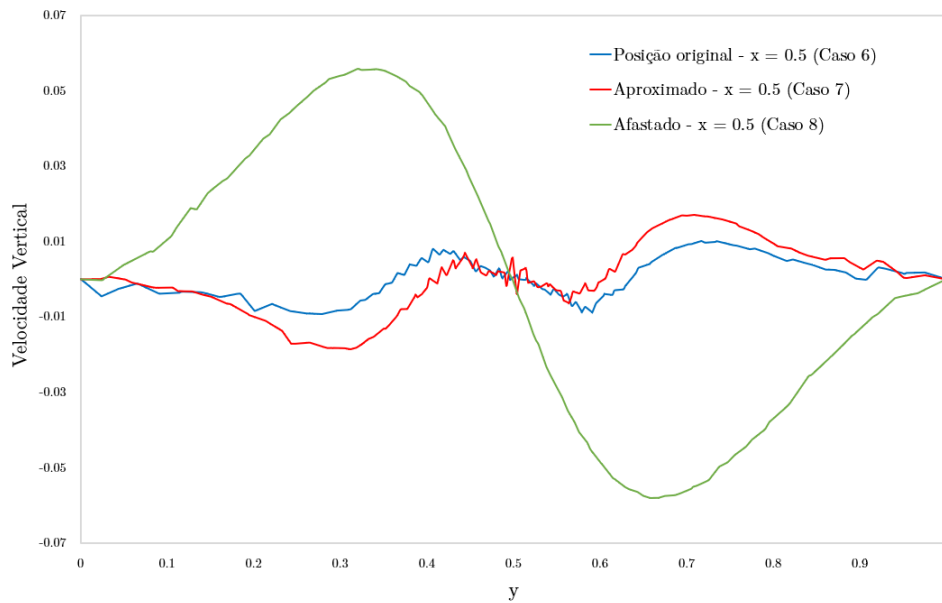
Figura 5.17: Solução numérica da Velocidade Horizontal para os casos (a) Caso 7, (b) Caso 8 e (c) gráfico comparativo da solução da Velocidade Horizontal para os 3 casos, obtidos no eixo $x = 0.5$.

Analisando as Velocidades (figuras 5.17 e 5.18) notamos que os resultados seguem a mesma tendência encontradas na Função Corrente e Vorticidade. Ou seja, quanto mais afastamos os obstáculos, mais eles tendem a se comportar como apenas um único obstáculo (figura 5.8), exercendo menos influência um sobre o outro.



(a) Velocidade Vertical - Caso 7

(b) Velocidade Vertical - Caso 8



(c) Gráfico comparativo da Velocidade Vertical entre os Casos 6, 7 e 8.

Figura 5.18: Solução numérica da Velocidade Vertical para os casos (a) Teste 2, (b) Teste 3 e (c) gráfico comparativo da solução da Velocidade Vertical para os 3 testes obtidos no eixo $x = 0.5$.

Assim como ocorreu com todas as outras variáveis estudadas, a Velocidade Vertical tende a se comportar como o caso de apenas um único obstáculo (figura 5.8) conforme os obstáculos são afastados.

5.2.3 Caso 9 - Dois obstáculos elípticos com maior eixo na horizontal

Como foi visto anteriormente na seção 5.1, a geometria do obstáculo também influencia no resultado. Desta forma, foram simuladas duas situações, de modo a permitir a compreensão da mistura da influência de dois obstáculos alinhados horizontalmente de geometrias distintas a circular. O primeiro caso (Caso 9) simulado foi de obstáculo elíptico com o maior eixo na horizontal, sendo o primeiro centrado em $x = 0.4$ e permeabilidade horizontal de 0.2, tal como foi realizado no teste 1 (tabela 5.2). Assim como nos casos anteriores, os raio menor da elipse é de 0.03 e o maior raio tem valor igual a 0.07. Os resultados são apresentados a seguir, na figura 5.19.

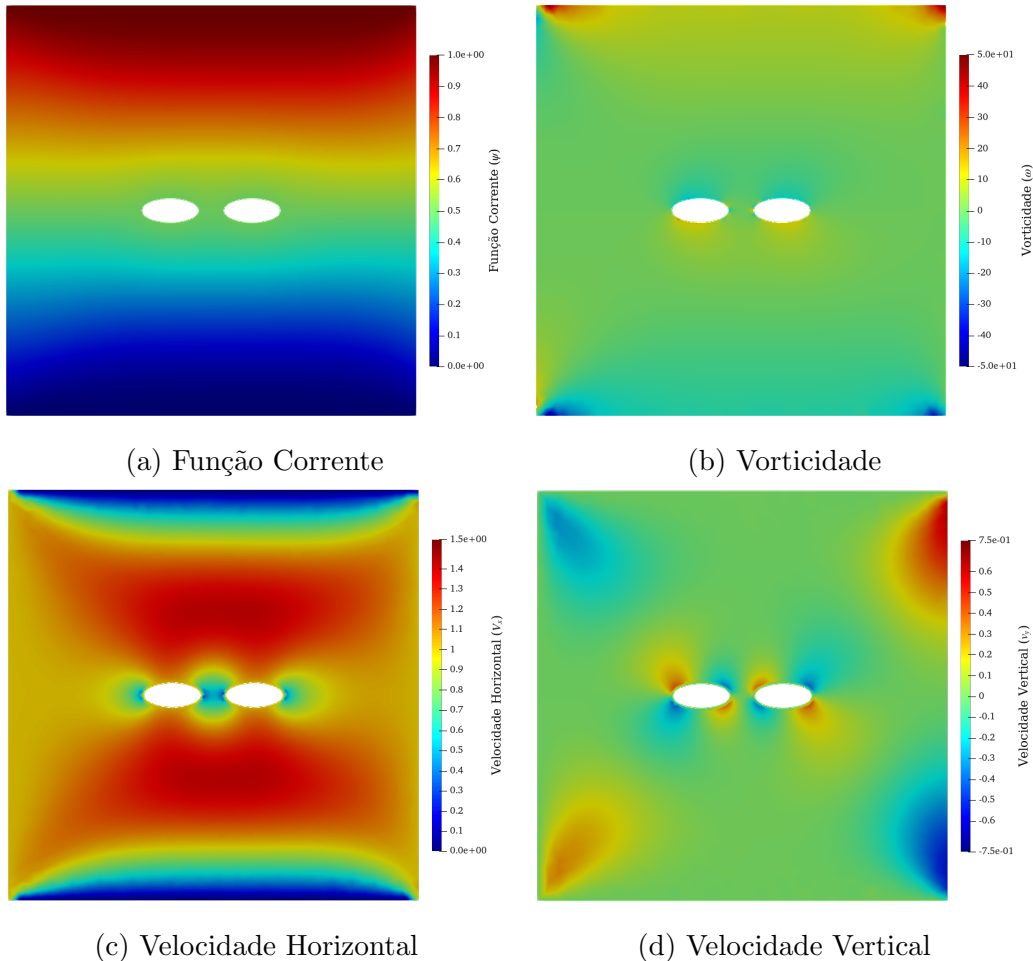


Figura 5.19: Solução numérica para um escoamento ao redor de dois obstáculos elípticos com maior eixo na horizontal alinhados horizontalmente, $Re = 10$, domínio quadrado, malha com 2990 nós e 5992 elementos, 500 passos e $dt = 0.001$.

5.2.4 Caso 10 - Dois obstáculos elípticos com maior eixo na vertical

Em seguida, foi simulado o caso para dois obstáculos elípticos alinhados horizontalmente com maior eixo na vertical (Caso 10), sendo a solução final apresentada na figura 5.20. O centro em x da primeira elipse e permeabilidade horizontal foram também os mesmos utilizados no teste 1 (tabela 5.2), e os valores de raios foram mantidos, invertendo-se apenas os eixos.

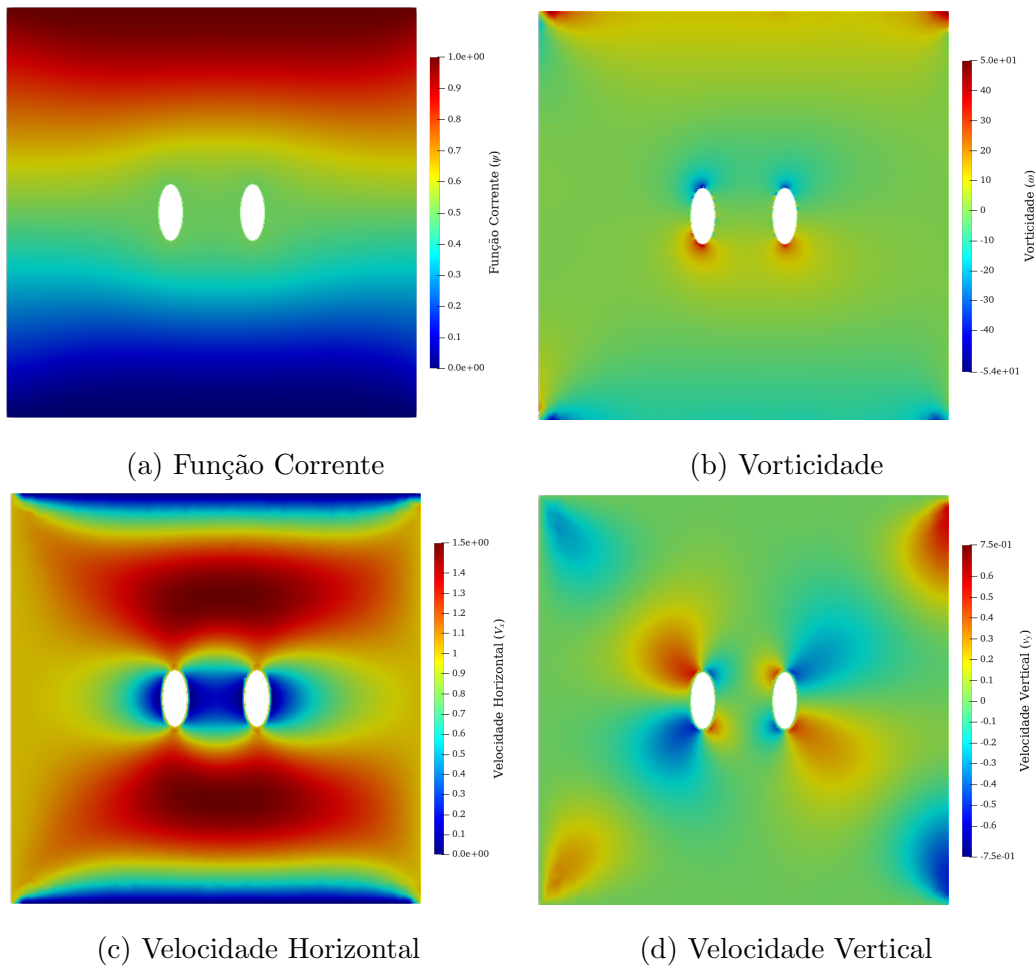


Figura 5.20: Solução numérica para um escoamento com dois obstáculos elípticos com maior eixo na vertical, $Re = 10$, domínio quadrado, malha de 3136 nós e 6284 elementos, 500 passos e $dt = 0.001$.

Após a análise destes casos, foi comparado os valores entres os casos de dois obstáculos circulares alinhados horizontalmente (Caso 6), com o de dois obstáculos elípticos com maior eixo na horizontal alinhados horizontalmente (Caso 9) e com o

dois obstáculos elípticos com maior eixo na vertical alinhados horizontalmente (Caso 10) para a Função Corrente, Vorticidade, e Velocidades Horizontal e Vertical. Em todos os casos para valores comparados foram obtidos no eixo vertical $x = 0.5$. Essa comparação pode ser visualizada nos gráficos apresentados na figura 5.21.

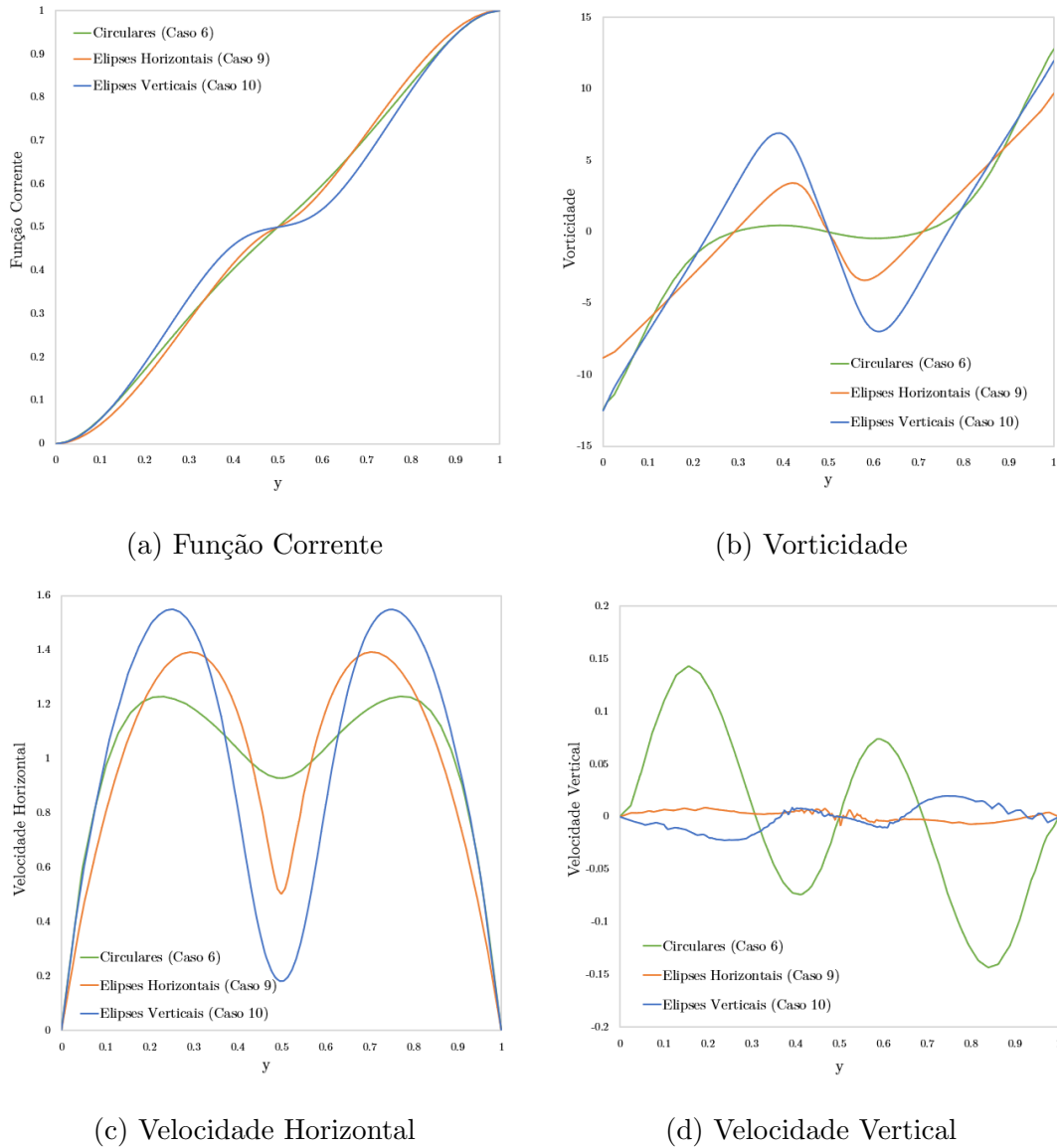


Figura 5.21: Comparação das soluções numéricas de 2 obstáculos alinhados horizontalmente entre os casos de obstáculos circular (Caso 6), elípticos com maior eixo na horizontal (Caso 9) e elípticos com maior eixo na vertical (Caso 10), no eixo vertical $x = 0.5$.

Como pode ser observado nos gráficos da figura 5.21 os obstáculos com geometria elíptica de maior eixo na vertical são os que mais causam “distúrbios” nas variáveis

analisadas, como podia-se esperar, dado que isso significa que o escoamento encontra um obstáculo maior a sua frente. Na Função Corrente (figura 5.21a) o comportamento tende a deixar de ser linear ao longo de y , passando a se assemelhar a uma função de segundo grau entre as bordas e os obstáculos. Para a Vorticidade e Velocidade Horizontal (figuras 5.21b e 5.21c, respectivamente) a mudança da geometria aumenta as amplitudes das variações observadas quando comparada com o caso de obstáculos circulares, principalmente para obstáculos elípticos com maior eixo na vertical. Já a Velocidade Vertical (figura 5.21d) apresenta uma maior quantidade de variações das direções de velocidade, mas as amplitudes de velocidade são diminutas quando comparamos os casos de obstáculos elípticos com o caso de obstáculos circulares.

5.3 Escoamento em meio poroso - 2x2

A configuração estudada em seguida foi um caso com um pouco mais de complexidade geométrica do que os casos até então apresentados. Partiu-se para o estudo de quatro obstáculos dispostos dois a dois, já sendo considerado este arranjo um meio poroso. Conforme pode ser visualizado na figura esquemática 5.22, as condições de contorno permaneceram as mesmas utilizadas até então, inclusive os valores de $c_1 = 0$ e $c_2 = 1$ e, a condição de contorno da vorticidade sendo calculada dentro do código a partir da equação 3.50; sendo a única diferença o número e a localização dos obstáculos.

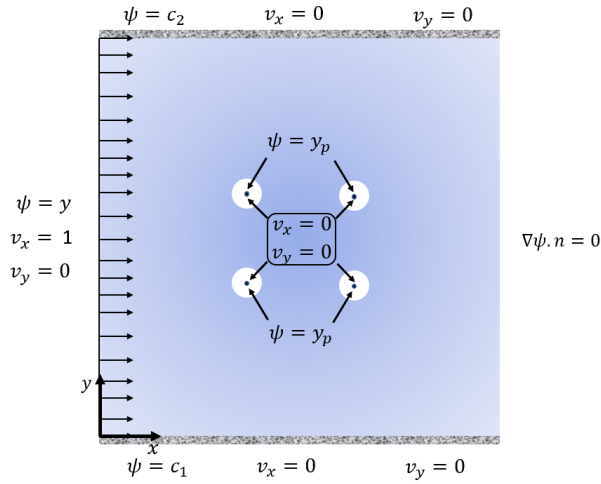


Figura 5.22: Desenho esquemático com condições de contorno com quatro obstáculos, organizados dois a dois.

Seguindo o mesmo raciocínio aplicado anteriormente, foram estudados 3 casos. Um com obstáculos circulares (Caso 11), outro com obstáculos elípticos com o maior eixo horizontal (Caso 12), e um terceiro sendo com obstáculos de geometria elíptica com maior eixo na vertical (Caso 13). Os parâmetros de cada caso, inclusive a distância horizontal e vertical do centro de cada obstáculo (permeabilidade horizontal e vertical, respectivamente) são listados na tabela 5.3.

Tabela 5.3: Parâmetros dos três casos com quatro obstáculos testados em uma domínio quadrado de 1×1 ($x = 0.1$ e $y = 0.1$).

Caso	Elemento	Primeiro Centro	Raio Horizontal	Raio Vertical	Permeabilidade Horizontal	Permeabilidade Vertical
11	Circular	$x = 0.4;$ $y = 0.4$	0.05	0.05	0.2	0.2
12	Elipse Horizontal	$x = 0.4;$ $y = 0.4$	0.07	0.03	0.2	0.2
13	Elipse Vertical	$x = 0.4;$ $y = 0.4$	0.03	0.07	0.2	0.2

5.3.1 Caso 11 - obstáculos circulares

Como realizado anteriormente, a primeira situação estudada foi para o caso de obstáculos circulares. Para tal, utilizou-se uma malha com uma qualidade semelhante aos casos vistos anteriormente; e as condições de contorno utilizadas estão expostas na figura 5.22. Os resultados encontrados podem ser visualizados na figura 5.23.

Cabe chamar atenção para a influência dos obstáculos dispostos em um y constante nos obstáculos dispostos em um y diferente. Isto pode ser facilmente visualizado na Velocidade Horizontal final (figura 5.23c), onde v_x tem grande variação em seu módulo entre os obstáculos em $y = 0.45$ e $y = 0.65$ (representado por uma intensa coloração avermelhada nesta região).

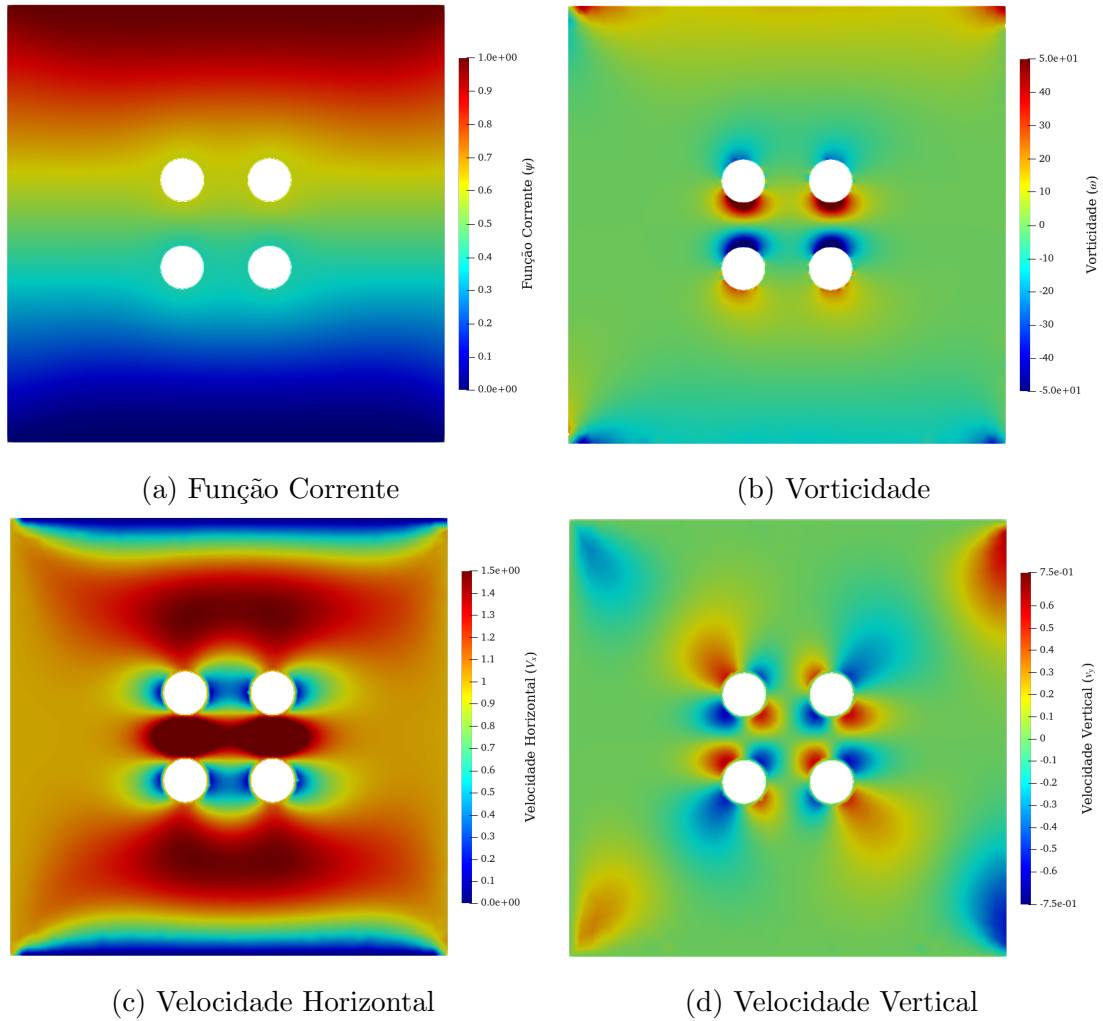


Figura 5.23: Solução numérica para um escoamento em um meio poroso com obstáculos circulares organizados 2 a 2, com $Re = 10$, domínio quadrado, malha com 3479 nós e 6980 elementos, 500 passos e $dt = 0.001$.

5.3.2 Caso 12 - obstáculos elípticos com maior eixo na horizontal

Em seguida estudou-se o mesmo caso, alterando apenas a geometria do obstáculo de circulares para elíptico com maior eixo na horizontal. A qualidade da malha foi mantida, tal como as condições de contorno. Os resultados para as variáveis estudadas neste caso podem ser observados na figura 5.24.

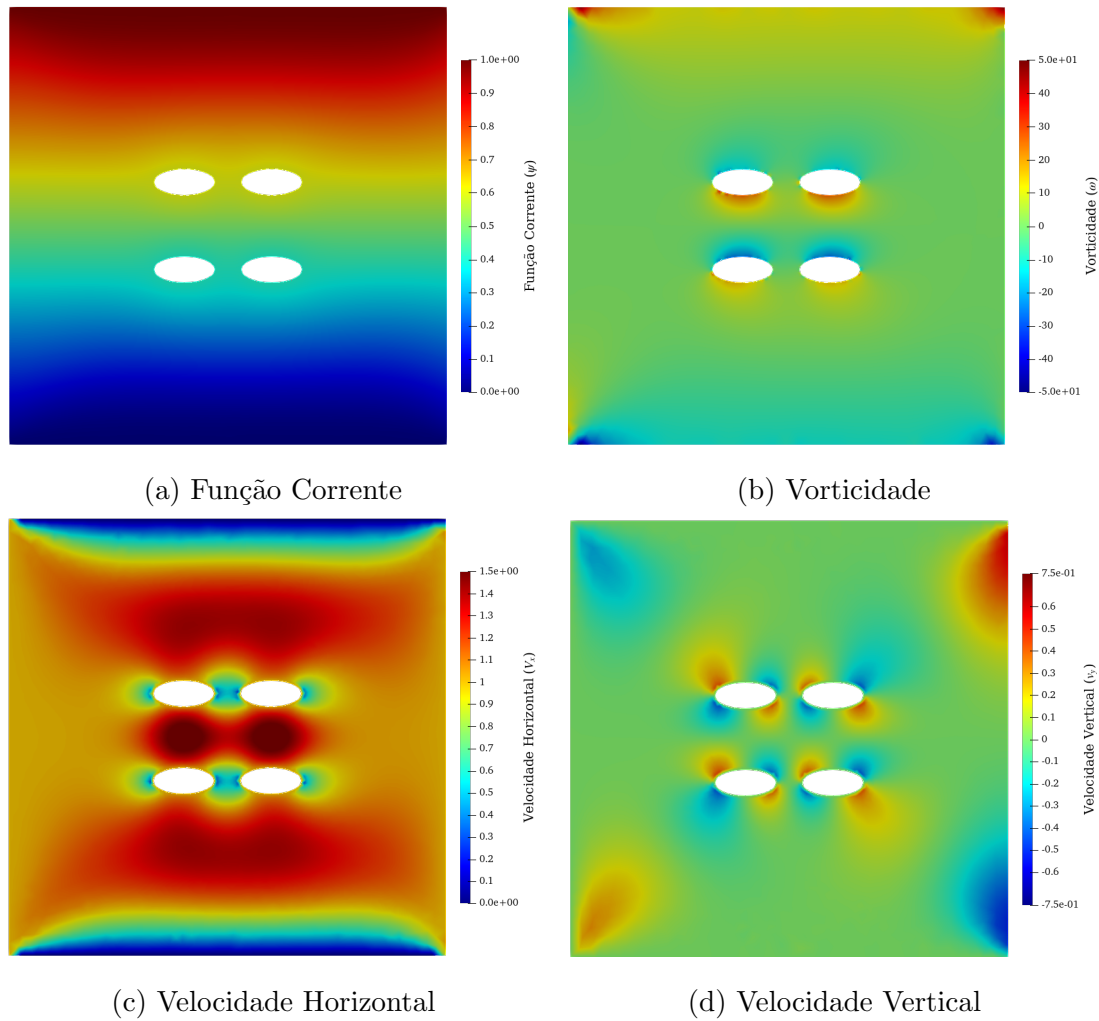


Figura 5.24: Solução numérica para um escoamento em um meio poroso, com obstáculos elípticos com maior eixo na horizontal organizados 2 a 2, com $Re = 10$, domínio quadrado, malha com 3847 nós e 7700 elementos, 500 passos e $dt = 0.001$.

5.3.3 Caso 13 - obstáculos elípticos com maior eixo na vertical

Para concluir o estudo de uma arranjo de quatro obstáculos organizados dois a dois, estudou-se o caso de obstáculos elípticos com maior eixo na vertical, obtendo-se os resultados expostos na figura 5.25.

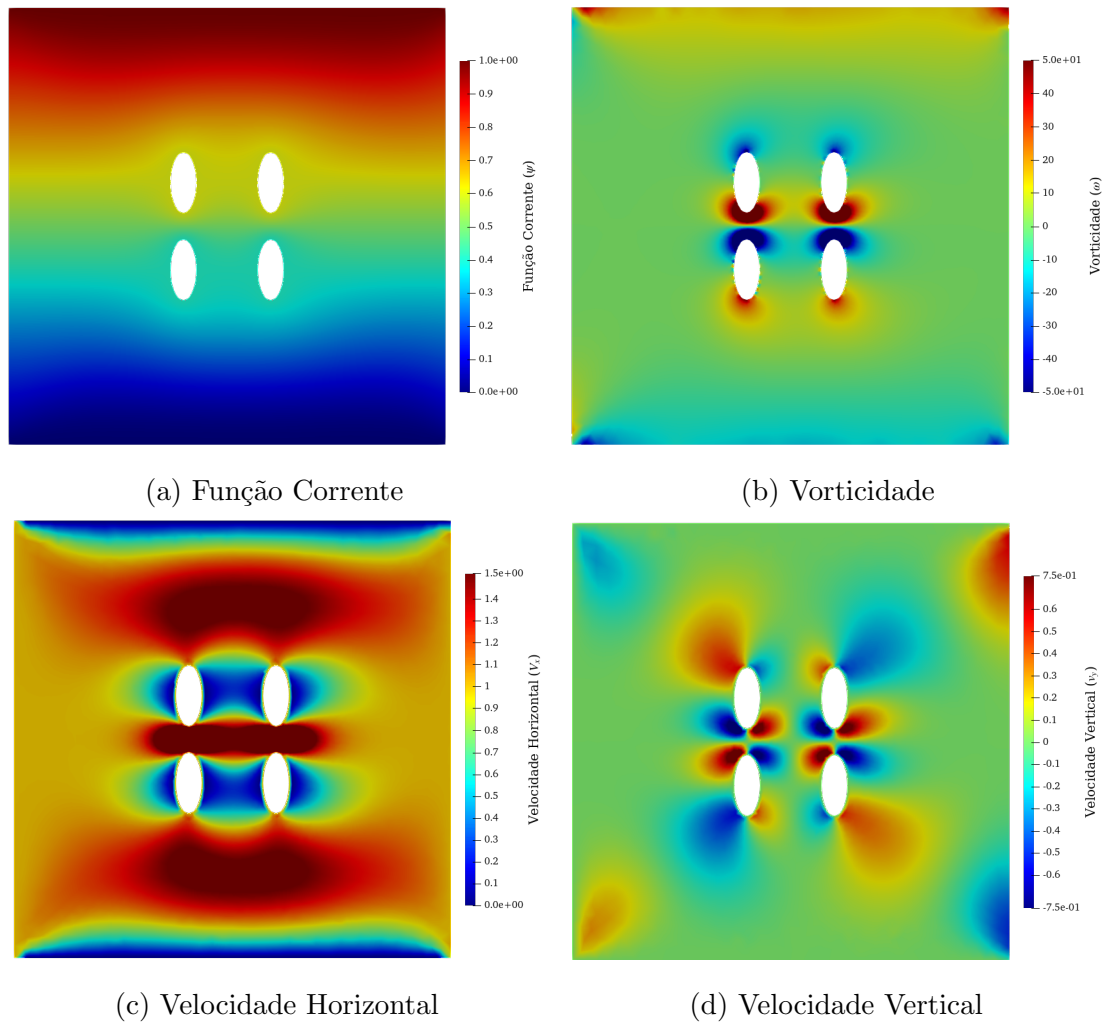
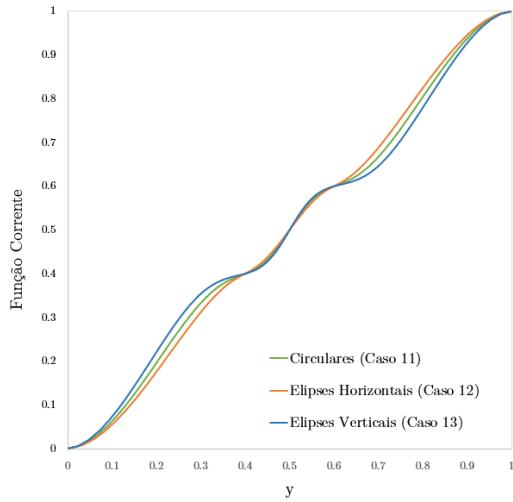
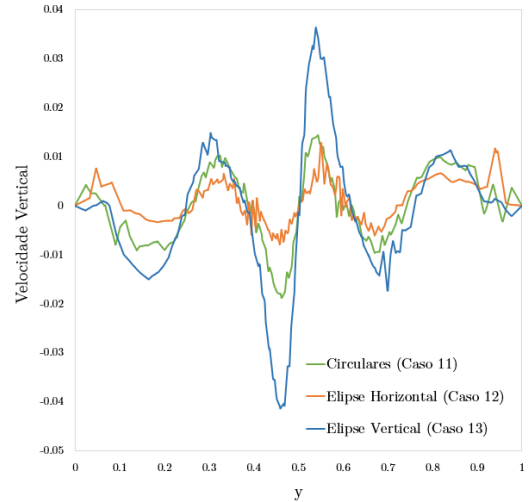


Figura 5.25: Solução numérica para um escoamento com um obstáculo de forma de elipses verticais, organizados 2 a 2, com $Re = 10$, malha quadrada com 3851 nós e 7724 elementos, com 500 passos e $dt = 0.001$.

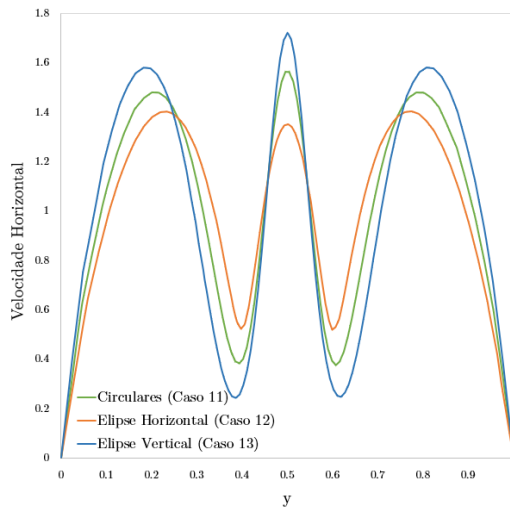
Após o estudos dos três casos com quatro obstáculos arranjados dois a dois (Casos 11, 12 e 13), os resultados dos três casos foram comparados. Apenas observando as figuras 5.23, 5.24 e 5.25, pode-se notar que a forma influencia na distância entre os obstáculos e conseqüentemente a distância entre os obstáculos tem grande influência nas variáveis estudadas. Isso fica claro ao observarmos, por exemplo, a Velocidade Horizontal (v_x) nos três casos (figuras 5.23c, 5.24c e 5.25c). Além disso, podemos visualizar essa influência ao analisarmos os gráficos comparativos entre os três casos em $x = 0.5$ e $y = 0.5$ expostos nas figuras 5.26 e 5.27, respectivamente.



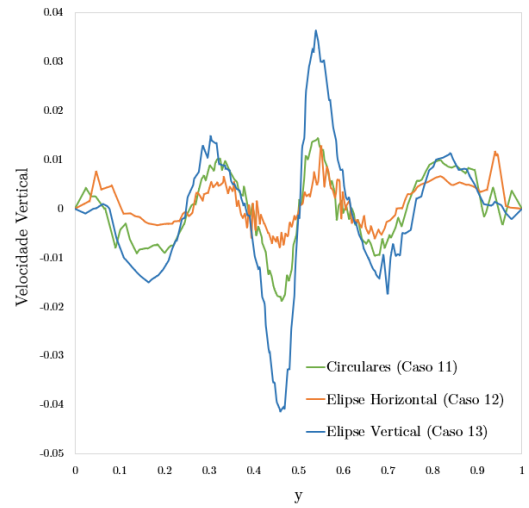
(a) Função Corrente



(b) Vorticidade



(c) Velocidade Horizontal

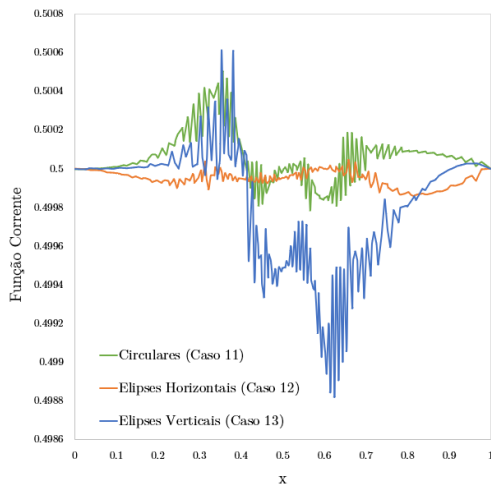


(d) Velocidade Vertical

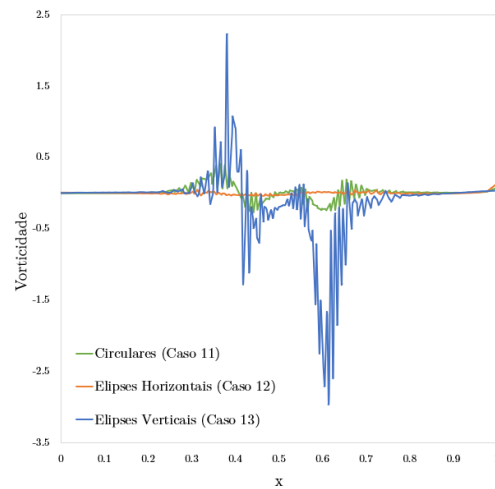
Figura 5.26: Comparação das soluções numéricas de obstáculos organizados 2 a 2 entre os casos de obstáculos circulares (Caso 11), elipse com maior eixo na horizontal (Caso 12) e elipse com maior eixo na vertical (Caso 13), no eixo vertical $x = 0.5$.

Note que mais uma vez, podemos observar que a geometria influencia consideravelmente nos resultados obtidos das quatro variáveis analisadas. Novamente, a amplitude das variações está atrelada ao espaço disponível para que o escoamento ocorra, de modo que as amplitudes das variações são menores nos casos de obstáculos elípticos com maior eixo na horizontal (como ocorreu no Caso 12), e maior para o caso de obstáculos elípticos com maior eixo na vertical (como visto no Caso 13), como podemos visualizar nos gráficos comparativos das Velocidade Horizontais e

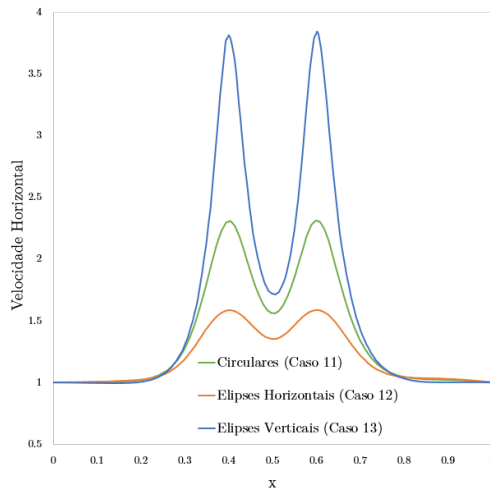
Verticais apresentadas nas figuras 5.26 e 5.27, respectivamente.



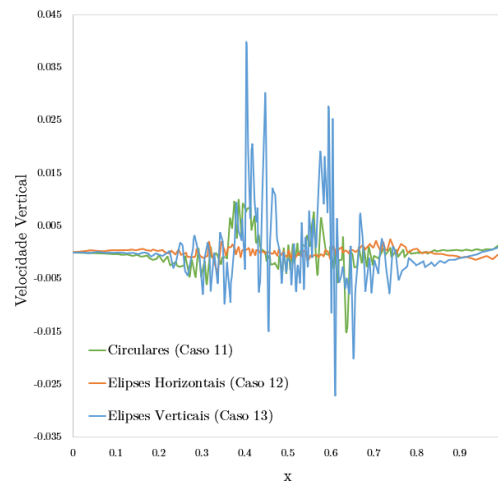
(a) Função Corrente



(b) Vorticidade



(c) Velocidade Horizontal



(d) Velocidade Vertical

Figura 5.27: Comparação das soluções numéricas de obstáculos organizados 2 a 2 entre os casos de obstáculos circulares (Caso 11), elipse com maior eixo na horizontal (Caso 12) e elipse com maior eixo na vertical (Caso 13), no eixo horizontal $y = 0.5$.

5.4 Escoamento em meio poroso - 3x3

5.4.1 Caso 14 - obstáculos circulares

Após os estudos com uma geometria de quatro obstáculos organizados dois a dois, continuou-se com o estudo incrementando mais um pouco a complexidade da geometria do meio poroso, e iniciou-se o estudo do escoamento em um meio

do domínio quadrado ($x = 1$ e $y = 1$) agora com nove obstáculos, dispostos três a Três. Assim como foi realizado até agora foi estudado primeiramente um meio poroso com obstáculos circulares. O refino da malha foi mantido igual aos casos até aqui estudados, resultando em uma malha com 4892 nós e 9831 elementos. A solução encontrada após 500 passos com um intervalo de tempo (dt) de 0.001 pode ser visualizada na figura 5.28.

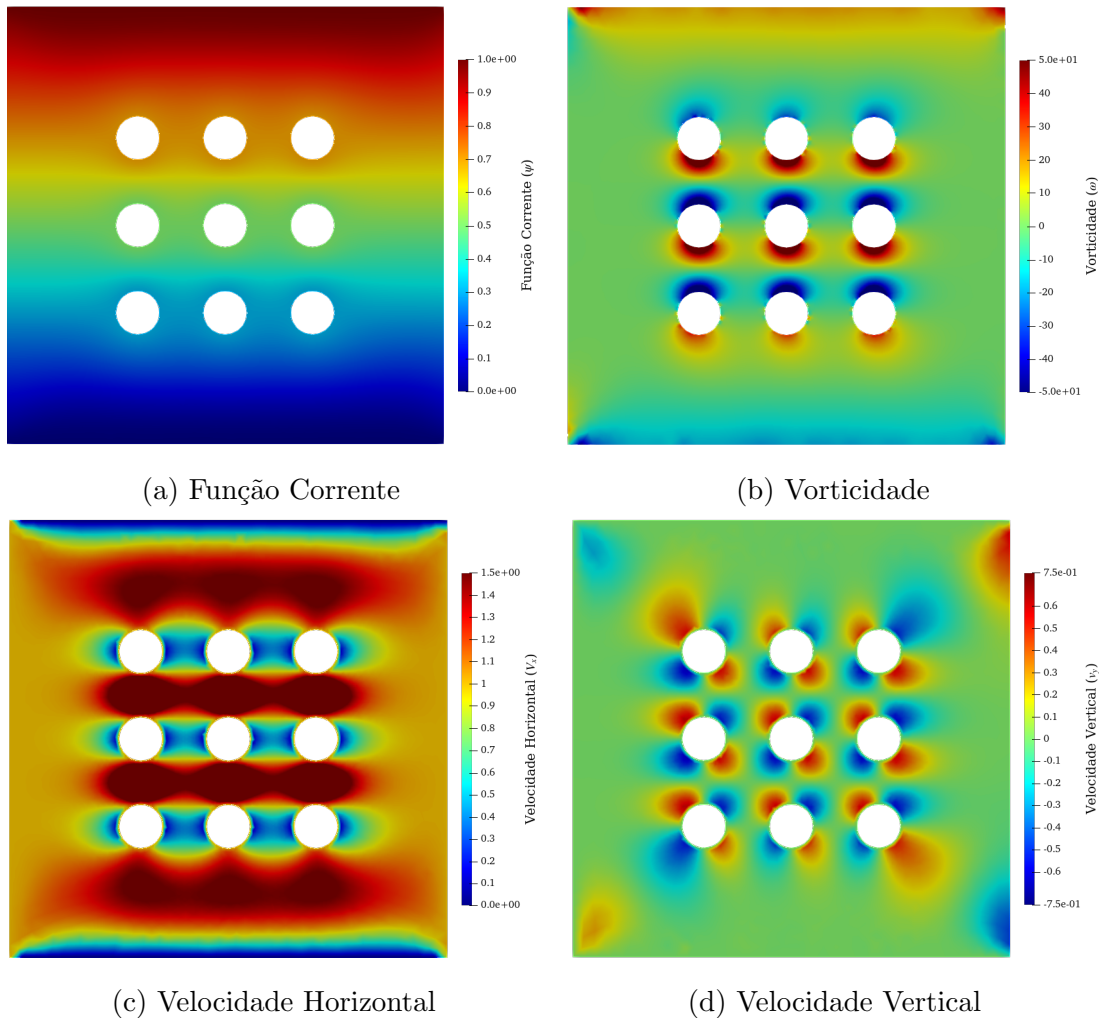


Figura 5.28: Solução numérica para um escoamento com obstáculo circulares, organizados 3 a 3, com $Re = 10$, domínio quadrado, malha com 4892 nós e 9831 elementos, com 500 passos e $dt = 0.001$.

Ao compararmos as figuras 5.23 e 5.28 podemos notar uma influência semelhante de um obstáculo com os obstáculos em sua adjacência entre o domínio quadrado com quatro obstáculos organizados dois a dois e o domínio quadrado com nove obstáculos organizados três a Três.

5.4.2 Caso 15 - obstáculos elípticos com maior eixo na horizontal

Dando sequencia aos testes com nove obstáculos, estudou-se o mesmo arranjo, com o mesmo domínio e refinamento de malha, com a única diferença sendo a geometria dos obstáculos, que desta vez são elipses com maior eixo na horizontal. O menor raio sendo igual a 0.03 e o maior raio sendo igual a 0.07. Os resultados deste estudo podem ser visualizados na figura 5.29.

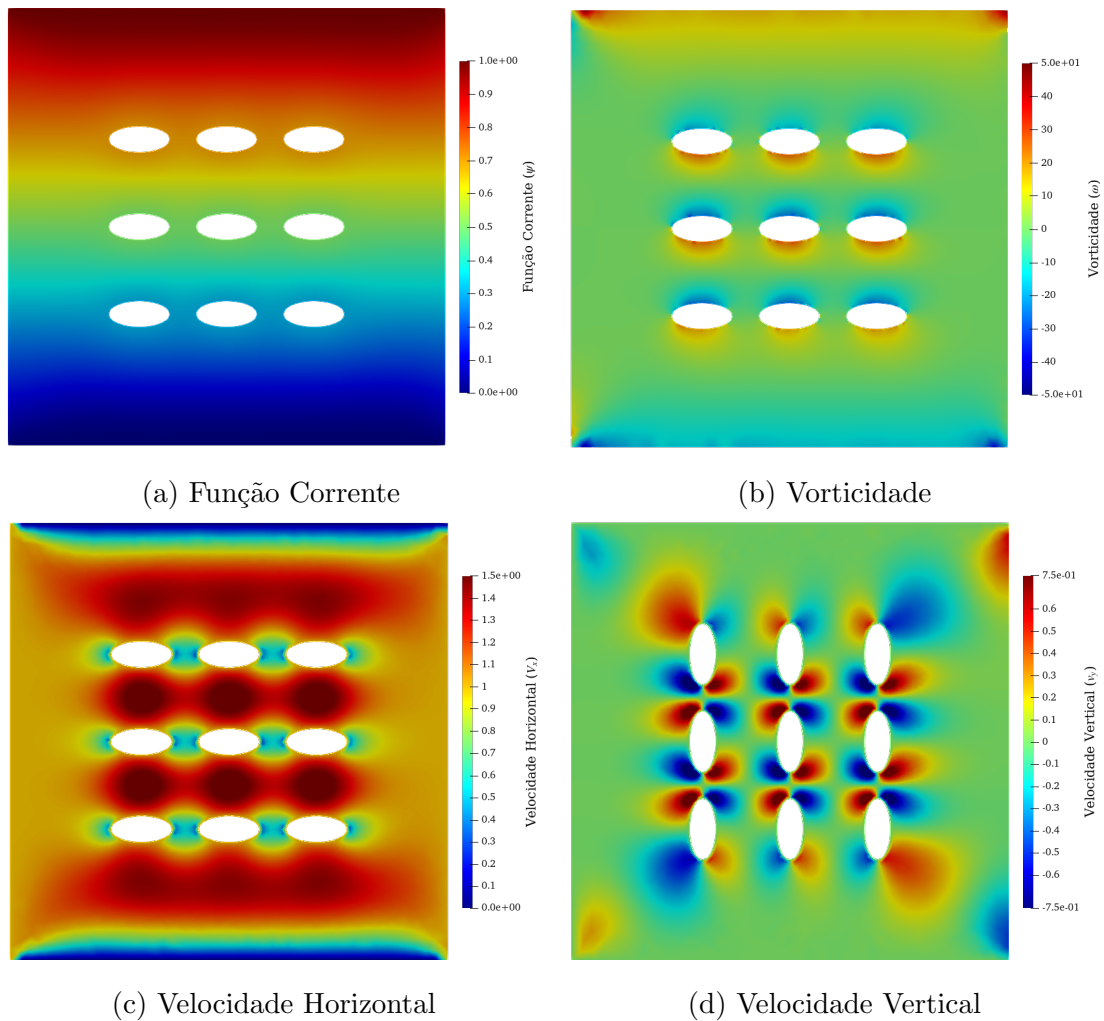


Figura 5.29: Solução numérica para um escoamento com obstáculo elípticos com maior eixo na horizontal, organizados 3 a 3, com $Re = 10$, domínio quadrado, malha com 5600 nós e 11247 elementos, com 500 passos e $dt = 0.001$.

5.4.3 Caso 16 - obstáculos elípticos com maior eixo na vertical

Finalizando os estudos de um meio poroso com nove obstáculos organizados três a três, alterou-se novamente a geometria do obstáculo, desta vez com o maior eixo na vertical, os demais parâmetros permaneceram iguais ao caso anterior, inclusive os valores dos raios maior e menor. O resultado final pode ser visualizado na figura 5.30.

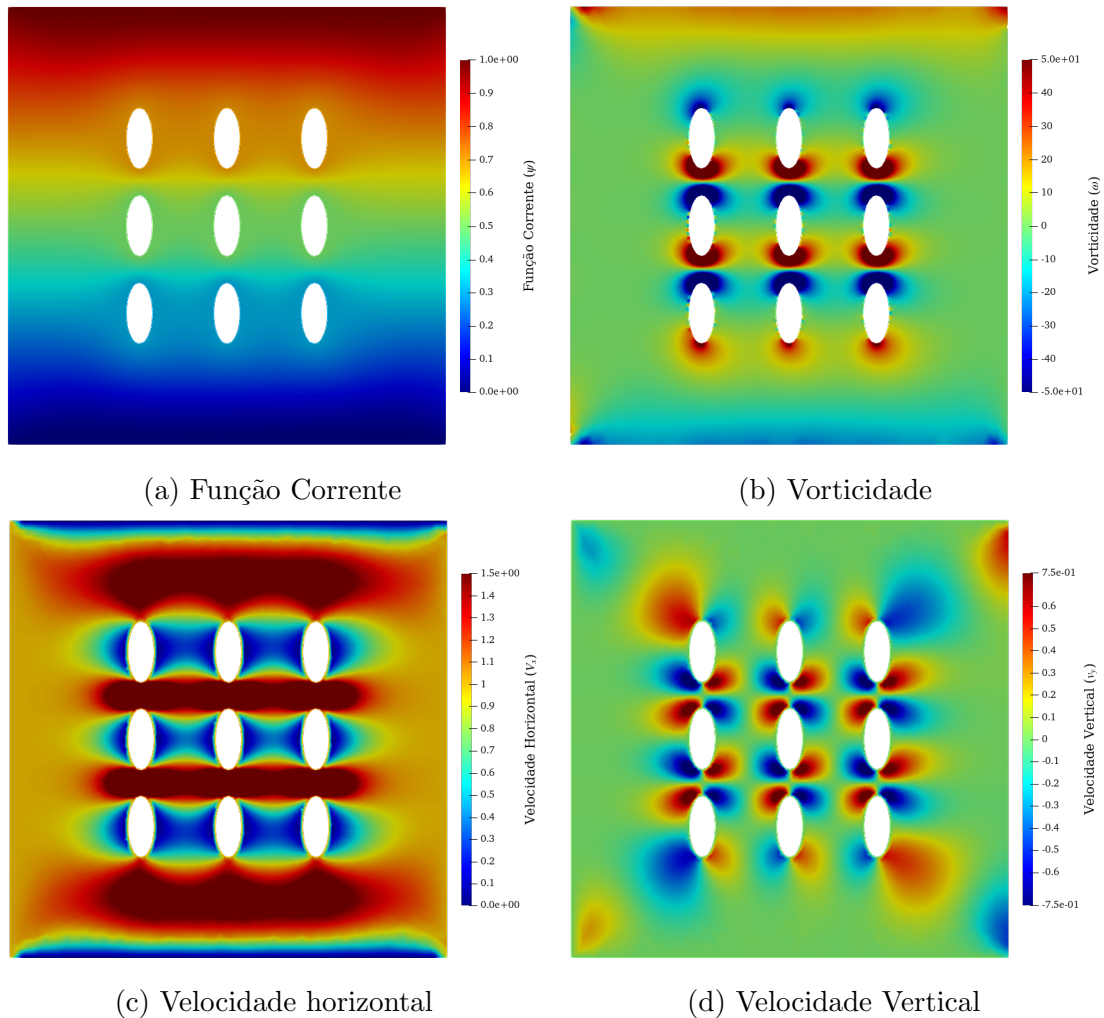


Figura 5.30: Solução numérica para um escoamento com obstáculo elípticos com maior eixo na vertical, organizados 3 a 3, com $Re = 10$, domínio quadrado, malha com 5570 nós e 11187 elementos, com 500 passos e $dt = 0.001$.

Após estudados os três casos, de geometrias distintas, de nove obstáculos organizados três a três foi verificado os valores da Função Corrente, Vorticidade, Velocidade Horizontal e Velocidade Vertical nos eixos vertical $x = 0.4$ (figura 5.31), localizado

no meio entre a primeira e segunda coluna de obstáculos e; horizontal $y = 0.4$ (figura 5.32), localizado no meio entre a primeira e segunda linha de obstáculos.

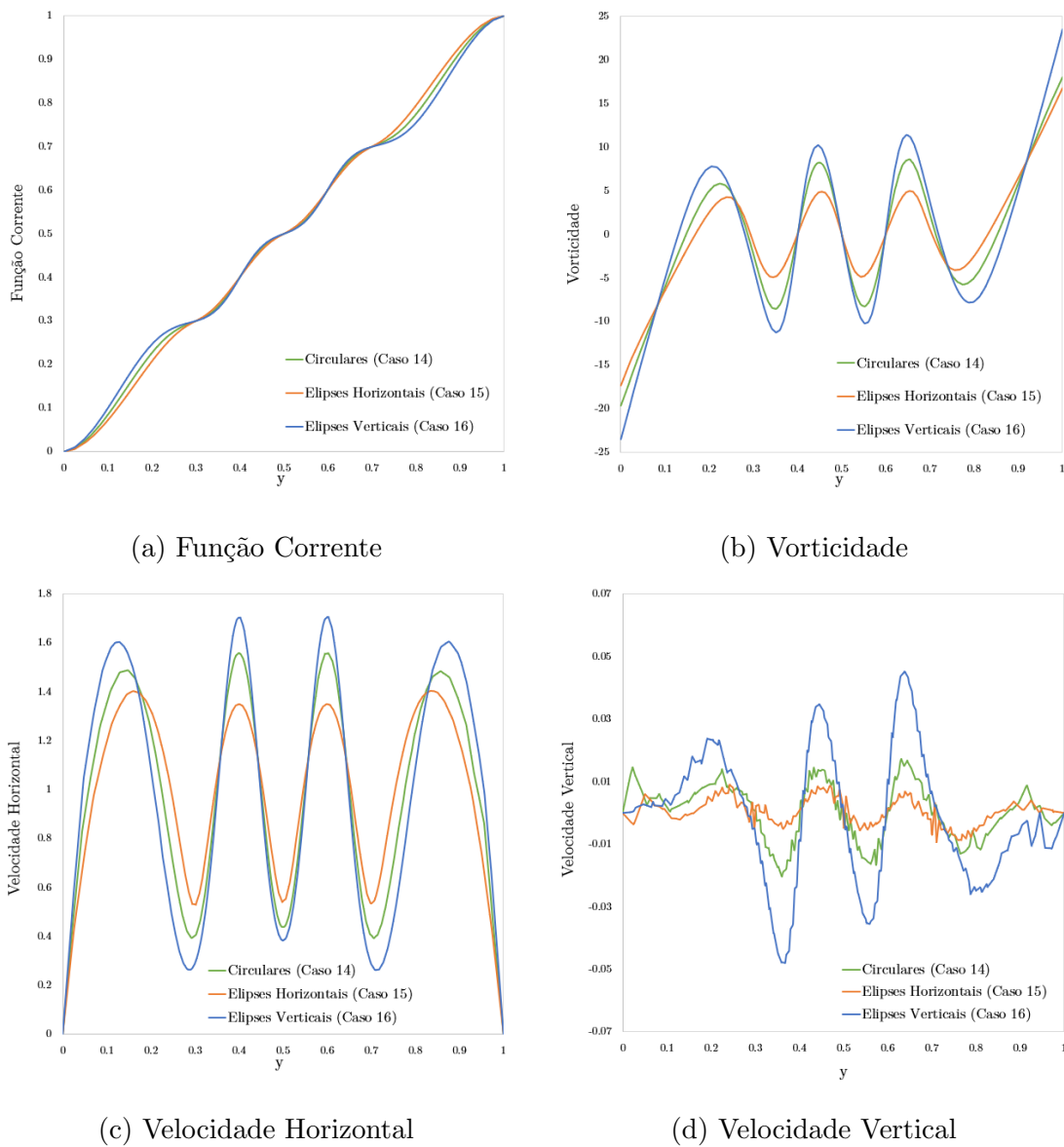
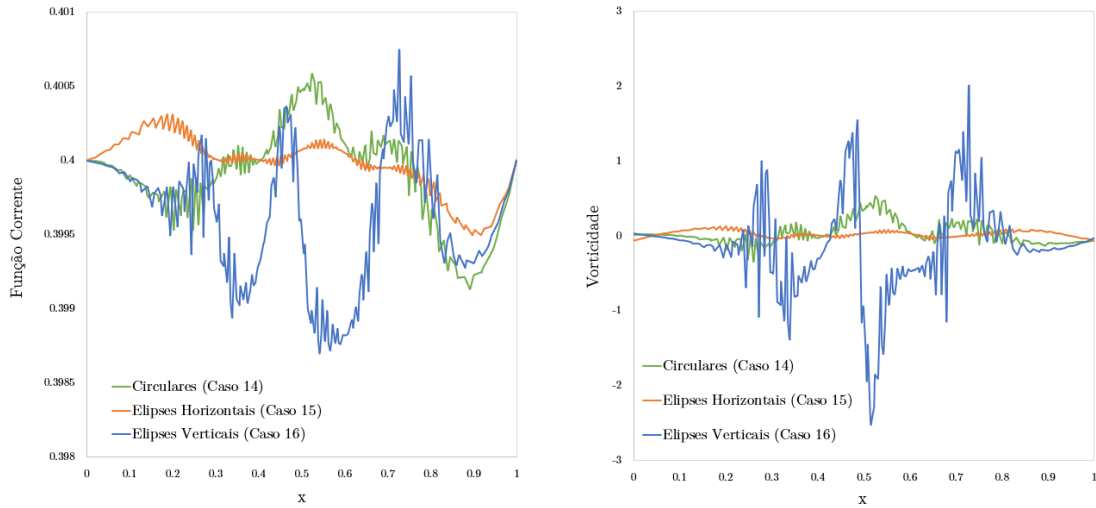


Figura 5.31: Comparação das soluções numéricas de obstáculos organizados 3 a 3 entre os casos de obstáculos circular (Caso 14), elipse com maior eixo na horizontal (Caso 15) e elipse com maior eixo na vertical (Caso 16), no eixo vertical $x = 0.4$.

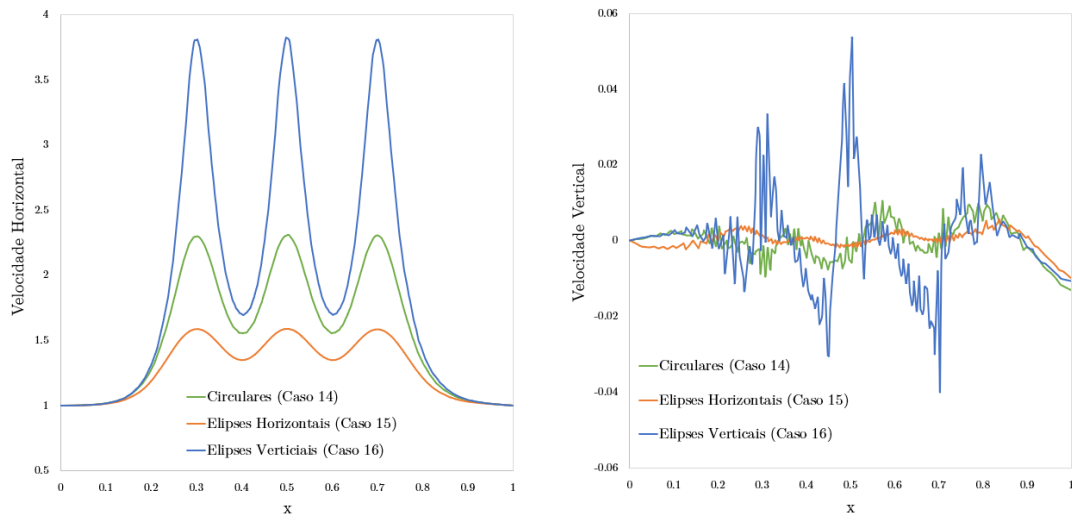
Assim como ocorreu nos casos anteriores, podemos observar que existe uma maior variação de amplitude para o caso de obstáculos elípticos com maior eixo na vertical (Caso 16), seguido do caso com obstáculos circulares (Caso 14), sendo o caso de obstáculos elípticos com maior eixo na horizontal (Caso 15) aquele que apresenta menor variação de amplitudes. Isto pode ser observado em todos os casos apresen-

tados nas figuras 5.31 e 5.32, mas fica fortemente evidente na figura 5.32c na qual podemos ver que a Velocidade Horizontal varia de 1 a cerca de 1.5 para o caso de obstáculos elípticos com maior eixo na horizontal e a Velocidade Horizontal no caso de obstáculos elípticos varia de 1 a quase 4.



(a) Função Corrente

(b) Vorticidade



(c) Velocidade Horizontal

(d) Velocidade Vertical

Figura 5.32: Comparação das soluções numéricas de obstáculos organizados 3 a 3 entre os casos de obstáculos circulares (Caso 14), elipse com maior eixo na horizontal (Caso 15) e elipse com maior eixo na vertical (Caso 16), no eixo horizontal $y = 0.4$.

5.5 Escoamento em meio poroso - 10x3

Dando continuidade ao incremento de complexidade do meio, seguiu-se para o estudo do escoamento em um meio com trinta obstáculos dispostos na distribuição dez (no eixo x) por três (no eixo y) (10×3). As condições de contorno permaneceram as mesmas estudadas anteriormente. Exceto que neste caso optou-se por um domínio retangular de dimensões $x = 2$ e $y = 1$, tal como foi realizado no primeiro caso deste capítulo (Caso 1 - figura 5.1), de modo a facilitar a distribuição dos obstáculos e seus espaçamentos na malha. A figura esquemática pode ser visualizada na figura 5.33 a seguir. Como ocorreram em todos os casos apresentados, a condição de contorno da vorticidade é calculada dentro do código, a partir da equação 3.50.

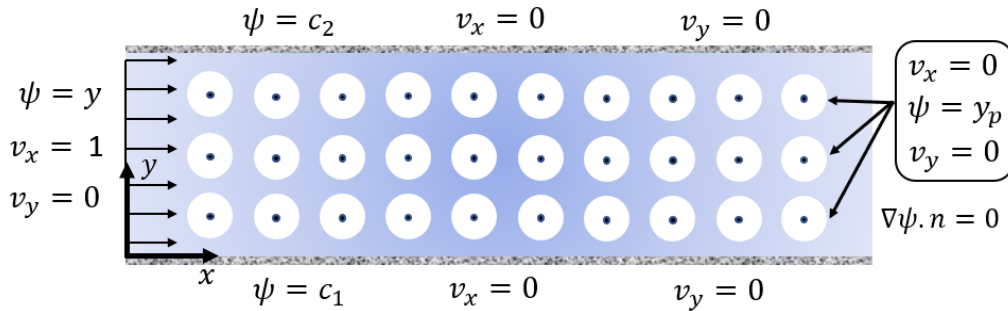


Figura 5.33: Desenho esquemático com condições de contorno com trinta obstáculos organizados em 10×3 .

5.5.1 Caso 17 - obstáculos circulares

Iniciou-se, novamente, o estudo dentro do arranjo proposto, com obstáculos circulares. Inicialmente foi realizada uma tentativa de simular uma malha com refino igual aos casos até então trabalhados, que resultou em uma malha de 46509 nós e 93170 elementos. Entretanto, para a simulação desta malha são necessários mais de 16 Gb de memória apenas para a geração de matrizes que necessitam ser geradas dentro do código para a solução deste problema com esse nível de detalhe. Esta é uma quantidade de memória maior do que a quantidade de memória disponível no computador utilizado. Desta forma, optou-se por reduzir a qualidade da malha deste caso, aumentando o tamanho das arestas dos elementos triangulares ao redor dos obstáculos, deixando-os do mesmo tamanho das arestas dos elementos triangulares ao redor do domínio (figura 5.34).

A figura 5.35 apresenta os resultados encontrados nesta situação para obstáculos circulares, após 1000 passos e um total de 27978 segundos de processamento.

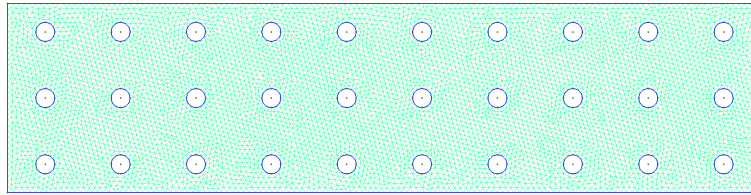
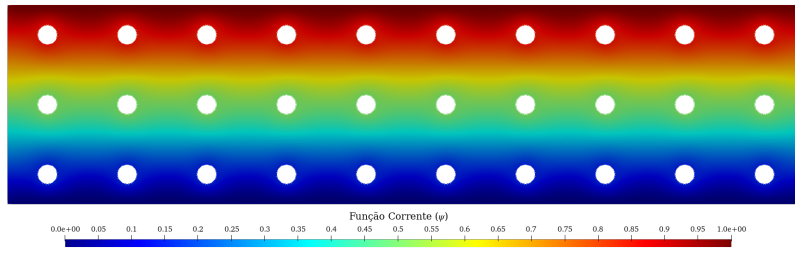
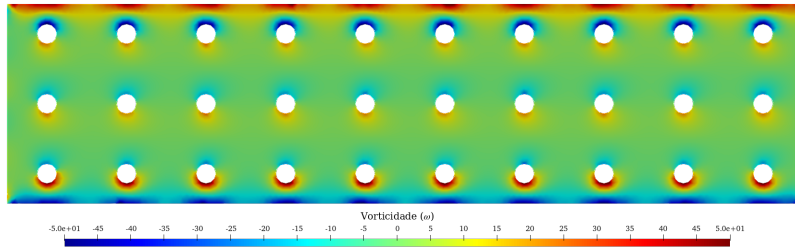


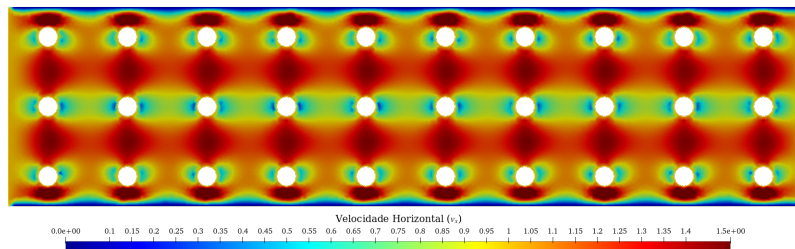
Figura 5.34: Malha do meio poroso com obstáculos circulares, organizados em 10 x 3 e refinamento de 0.03 ao redor dos dos obstáculos e domínio.



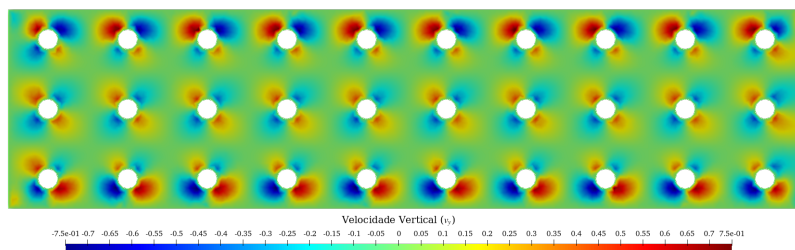
(a) Função Corrente



(b) Vorticidade



(c) Velocidade Horizontal



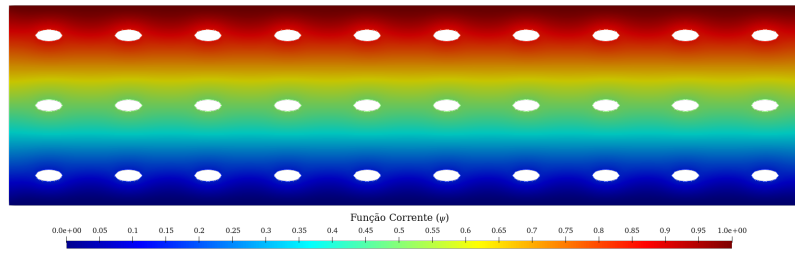
(d) Velocidade Vertical

Figura 5.35: Solução numérica para um escoamento em meio poroso com obstáculos circulares, organizados 10 por 3, com $Re = 10$, domínio retangular, malha com 6894 nós e 13940 elementos, 1000 passos e $dt = 0.001$.

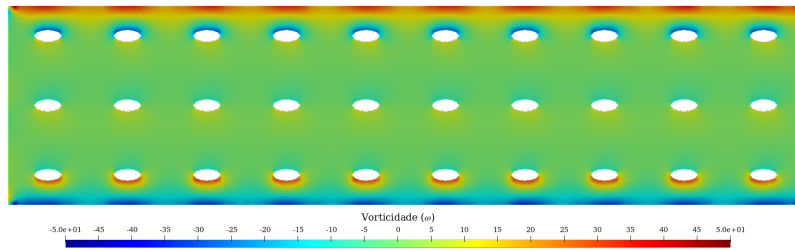
5.5.2 Caso 18 - obstáculos elípticos com maior eixo na horizontal

Em sequencia foi realizada a simulação para o caso semelhante, tendo como a diferença da geometria dos obstáculos, que desta vez eram elipses com maior eixo na horizontal. A malha utilizada neste caso tem 6717 nós e 13586 elementos, em

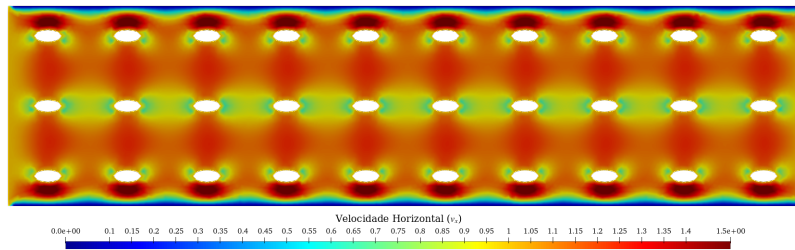
um domínio também retangular de dois por um ($x = 2$ e $y = 1$). O resultado após 1000 passos e 27440 segundos pode ser visualizado na figura 5.36.



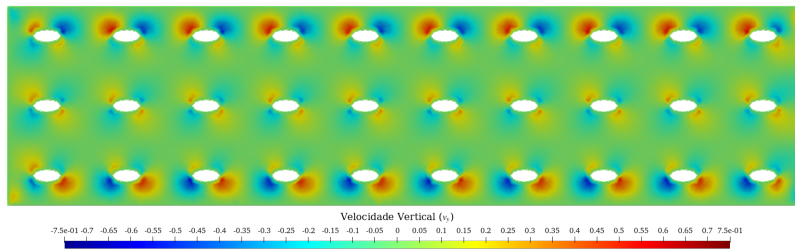
(a) Função Corrente



(b) Vorticidade



(c) Velocidade Horizontal

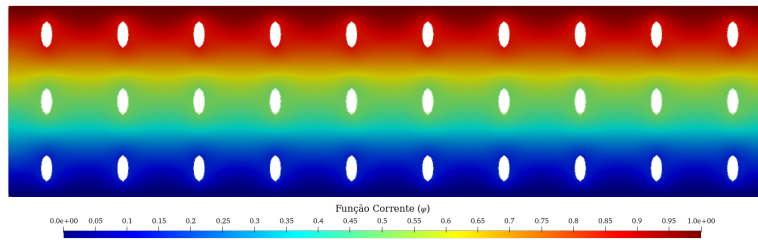


(d) Velocidade Vertical

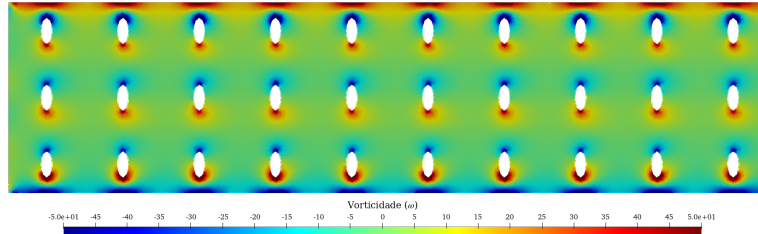
Figura 5.36: Solução numérica para um escoamento em meio poroso com obstáculos elípticos com maior eixo na horizontal, organizados 10 por 3, com $Re = 10$, domínio retangular, malha com 6717 nós e 13586 elementos, 1000 passos e $dt = 0.001$.

5.5.3 Caso 19 - obstáculos elípticos com maior eixo na vertical

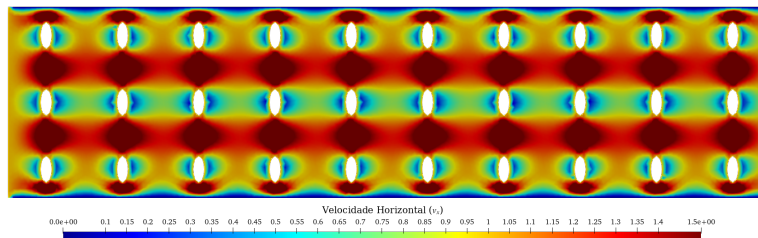
Finalizando os estudos do escoamento em meio poroso deste trabalho, foi realizada a simulação de trinta obstáculos, organizados em dez por três, com obstáculos de geometrias elípticas, com o maior eixo na vertical. As condições exceto a geometria dos obstáculos foram mantidas as mesmas do caso anterior, gerando uma malha de 6711 nós e 13574 elementos, e a simulação demorou 27243 segundos para ser finalizada. O resultado após 1000 passos é exposto na figura 5.37.



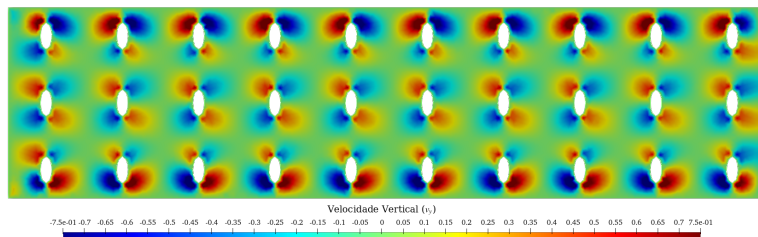
(a) Função Corrente



(b) Vorticidade



(c) Velocidade Horizontal



(d) Velocidade Vertical

Figura 5.37: Solução numérica para um escoamento em meio poroso com obstáculos elípticos com maior eixo na horizontal, organizados 10 por 3, com $Re = 10$, domínio retangular, malha com 6711 nós e 13574 elementos, 1000 passos e $dt = 0.001$.

Ao observarmos qualitativamente os três casos de trinta obstáculos estudados (Casos 17, 18 e 19) podemos observar que as tendências encontradas até aqui são mantidas. Ou seja, a proximidade dos obstáculos e a geometria tem influencia nas variações da Função Corrente, Vorticidade, Velocidade Horizontal e Velocidade Vertical do escoamento, após decorrido o tempo de necessário para que o regime permanente seja alcançado.

No caso da geometria, aquelas que encurtam o espaço livre para o escoamento

resultam em maiores variações das funções estudadas. Este fato fica mais evidente na visualização das Velocidades Horizontais encontradas, dada a variação da intensidade de vermelho expostas nas figuras.

As distâncias tentem a também seguir um mesmo padrão. Quanto mais distantes menor a influência de um obstáculo em outro (e obviamente essas distâncias também são influenciadas não somente pelo número de obstáculos no domínio e suas localizações, como pela geometria, que faz com que os limites de cada obstáculo estejam mais próximos ou mais afastados um dos outros).

Além disso, podemos notar também que o “alinhamento” dos obstáculos no eixo y formam uma espécie de “corredor” para o escoamento, facilitando o mesmo em algumas alturas de y (aquelas que não possuem obstáculos em y).

Dados os encontrados até aqui, e as limitações computacionais encontradas nestes últimos casos, o arranjo de 30 obstáculos foi o último a ser estudado no presente trabalho.

Capítulo 6

Conclusões

Neste trabalho foi apresentado uma breve revisão de literatura sobre biocombustíveis e sobre como a associação da produção deste tipo de combustível com utilização da captura e armazenamento de carbono em aquíferos salinos pode ajudar a reduzir a concentração de gás carbônico presente atualmente na nossa atmosfera.

Encontra-se também presente neste trabalho uma rápida revisão de mecânica dos fluidos e sua leis governantes; além das primeiras noções de Método de Elementos Finitos.

Em sequência, foi apresentada a formulação teórica da função Corrente-Vorticidade a partir da Equação de Navier-Stokes, e a sua transformação para um modelo numérico a ser utilizado como base das análises da mesma a partir do Método de Elementos Finitos.

Para iniciar os estudos desejados, primeiramente foi realizado a verificação do código escrito em *Python* em casos já amplamente conhecidos e desenvolvidos na literatura, como o escoamento de Lid Driven e entre placas planas. Após a validação, iniciou-se o estudo do escoamento em um meio poroso, inicialmente simplificado para um meio com apenas um único obstáculo, e em seguida aumentou-se gradativamente a complexidade do meio poroso, alternado-se a geometria do meio estudado, o número de obstáculos, suas disposições no domínio e a geometria dos obstáculos até uma situação em um domínio com trinta obstáculos.

Os estudos realizados indicam grande influência dos parâmetros variados. A geometria do domínio influencia nas respostas obtidas, uma vez que o fluido tem mais tempo/espço para se desenvolver antes de encontrar o obstáculo. A distância

entre os obstáculos também influencia nas variáveis estudadas (função corrente, vorticidade, velocidade horizontal e velocidade vertical). No quesito da distância, quanto mais distantes os obstáculos encontram-se entre si, menor é a influência entre eles, de modo que quanto mais os obstáculos encontram-se afastados, mais o escoamento do fluido se aproxima ao escoamento observado em um meio com apenas um obstáculo. A geometria obstáculo também influencia no comportamento do escoamento. Obstáculos mais alongados perpendicularmente ao escoamento geram maior “barreiras”, dificultando a passagem do fluido pelos vãos livres.

Dadas algumas limitações, sejam de poder computacional, sejam por outros motivos, algumas ideias iniciais não puderam ser implementadas e estudadas neste trabalho. Sendo assim, a seguir são listadas algumas sugestões de trabalhos futuros e melhorias para este trabalho:

- Testar casos com diferentes números de Reynolds;
- Incluir a influência da temperatura no escoamento;
- Testar situações onde os obstáculos estejam desalinhados entre si;
- Testar situações nas quais as geometrias dos obstáculos sejam diferentes entre si e situações nas quais os obstáculos tenham geometrias distintas das estudadas no presente trabalho;
- Testar situações de distribuição e geometrias aleatórias dos obstáculos;
- Implementar a Lei de Darcy no código elaborado e comparar os resultados com os resultados obtidos neste trabalho;
- Modular o código desenvolvido;
- Implementar uma interface gráfica (GUI - *Graphical User Interface*), a fim de facilitar a utilização por parte de usuários que não tenham ou tenham pouca noção de linguagem de programação (em específico *Python*, que foi a linguagem na qual o código deste trabalho foi desenvolvido).

Referências Bibliográficas

- [1] BACHU, S., “CO₂ storage in geological media: Role, means, status and barriers to deployment”, *Progress in energy and combustion science*, v. 34, n. 2, pp. 254–273, 2008.
- [2] GOUGH, C., UPHAM, P., “Biomass energy with carbon capture and storage (BECCS or Bio-CCS)”, *Greenhouse Gases: Science and Technology*, v. 1, n. 4, pp. 324–334, 2011.
- [3] GAZZONI, D. L., “Balanço de emissões de CO₂ por biocombustíveis no Brasil: histórico e perspectivas.” *Embrapa Soja-Livro técnico (INFOTECA-E)*, 2014.
- [4] CHANGE, I. C., “Mitigation of Climate Change. Contribution of Working Group III to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change”, 2022.
- [5] ONARHEIM, K., ARASTO, A., “Sustainability and ghg impact of bio-cc(u)s”, *Technical Report 41, IEA Bioenergy*, 2016.
- [6] “United Nations Framework Convention on Climate Change.” *The Paris agreement*, 2017.
- [7] MACHADO, C. X., OTHERS, “Potencial brasileiro para implementação de projetos de armazenamento geológico de carbono da biomassa a partir de usinas de álcool”, 2015.
- [8] DEL ÁLAMO, G., SANDQUIST, J., VREUGDENHIL, B., et al., “Implementation of bio-CCS in biofuels production”. In: *IEA Bioenergy Task*, v. 33, 2015.

- [9] MALODE, S. J., PRABHU, K. K., MASCARENHAS, R. J., et al., “Recent advances and viability in biofuel production”, *Energy Conversion and Management: X*, v. 10, pp. 100070, 2021.
- [10] AMBAYE, T. G., VACCARI, M., BONILLA-PETRICIOLET, A., et al., “Emerging technologies for biofuel production: a critical review on recent progress, challenges and perspectives”, *Journal of environmental management*, v. 290, pp. 112627, 2021.
- [11] REHAM, S., MASJUKI, H. H., KALAM, M. A., et al., “Study on stability, fuel properties, engine combustion, performance and emission characteristics of biofuel emulsion”, *Renewable and Sustainable Energy Reviews*, v. 52, pp. 1566–1579, 2015.
- [12] XU, L., CHENG, J.-H., LIU, P., et al., “Production of bio-fuel oil from pyrolysis of plant acidified oil”, *Renewable energy*, v. 130, pp. 910–919, 2019.
- [13] GUEDES, J. M., SANTOS, A. G. D., DOS SANTOS, H. S., “Uso da biomassa como fonte energética para produção de biocombustíveis.” *Ambiente: Gestão e Desenvolvimento*, 2021.
- [14] HUPPERT, H. E., NEUFELD, J. A., “The fluid mechanics of carbon dioxide sequestration”, *Annual review of fluid mechanics*, v. 46, pp. 255–272, 2014.
- [15] MOREIRA, J. R., ROMEIRO, V., FUSS, S., et al., “BECCS potential in Brazil: Achieving negative emissions in ethanol and electricity production based on sugar cane bagasse and other residues”, *Applied Energy*, v. 179, pp. 55–63, 2016.
- [16] ZIÓŁKOWSKI, P., MADEJSKI, P., AMIRI, M., et al., “Thermodynamic analysis of negative CO₂ emission power plant using Aspen Plus, Aspen Hysys, and Epsilon software”, *Energies*, v. 14, n. 19, pp. 6304, 2021.
- [17] IGLESIAS, R. S., KETZER, J. M., MELO, C. L., et al., “Carbon capture and geological storage in Brazil: an overview”, *Greenhouse Gases: Science and Technology*, v. 5, n. 2, pp. 119–130, 2015.

- [18] BECK, B., CUNHA, P., KETZER, M., et al., “The current status of CCS development in Brazil”, *Energy Procedia*, v. 4, pp. 6148–6151, 2011.
- [19] CIVIL, C., “LEI Nº 12.187, DE 29 DE DEZEMBRO DE 2009”, *Institui a Política Nacional sobre Mudança do Clima-PNMC e dá outras providências. DOU de*, v. 30, 2009.
- [20] PRITCHARD, P. J., MITCHELL, J. W., *Fox and McDonald’s introduction to fluid mechanics*. John Wiley & Sons, 2011.
- [21] ÇENCEL, Y. A., CIMBALA, J. M., “Mecânica dos Fluidos–Fundamentos e Aplicações”, *SÃO PAULO, McGraw-Hill*, 2012.
- [22] TEIXEIRA-DIAS, F., PINHO-DA CRUZ, J., VALENTE, R. F., et al., “Método dos elementos finitos: técnicas de simulação numérica em engenharia”, 2010.
- [23] MELCONIAN, M. V., *Modelagem numérica e computacional com similitude e elementos finitos*. Editora Edgard Blücher, 2014.
- [24] ANJOS, G. R., ***Computação científica para engenheiros***. Brasil, disponível em: ‘<https://drive.google.com/file/d/1F3hddkpuzMoLg7nFrfrRq0jxB7JGMD8NR/view>’, 2019.
- [25] LOGAN, D. L., *A first course in the finite element method*. Cengage Learning, 2016.
- [26] PONTES, J., MANGIAVACCHI, N., “Fenômenos de Transferência com Aplicações às Ciências Físicas e à Engenharia”, 2016.
- [27] GEUZAINÉ, C., REMACLE, J.-F., “Gmsh: a three-dimensional finite element mesh generator with built-in pre-and post-processing facilities”, 2008.
- [28] AHRENS, J., GEVECI, B., LAW, C., “Paraview: An end-user tool for large data visualization”, *The visualization handbook*, v. 717, n. 8, 2005.
- [29] CUNHA, L. H. C. D., OTHERS, “ALE finite element method for simulating flows with the streamfunction-vorticity formulation”, 2020.

- [30] MICHEL, R., “Fluid Dynamics: An Introduction”, 2015.
- [31] SILVA, J. A., “Método de elementos finitos em python com a abordagem Lagrangiana Euleriana para oscilações de um cilindro em um escoamento transversal.” 2020.
- [32] CUPERTINO, S. A., *A responsabilidade civil na estocagem de carbono no Brasil*, Ph.D. Thesis, Universidade de São Paulo, 2019.

Apêndice A

Códigos Fonte

Neste apêndice são apresentados os códigos desenvolvidos no GMSH e em *Python* utilizados para geração de malha, verificação do código e resolução dos casos apresentados neste trabalho.

A.1 Script para geração das malhas no GMSH

A seguir, encontra-se descrito o código utilizado para geração das malhas no *software* GMSH, versão 4.10.0.

```
1 lc = 0.03; // refino na parede (dominio)
2 le = 0.01; // refino ao redor da elipse (obstaculo)
3 //+
4 //Pontos do dominio
5 Point(1) = {0, 0, 0, lc};
6 Point(2) = {1.0, 0, 0, lc};
7 Point(3) = {1.0, 1, 0, lc};
8 Point(4) = {0, 1, 0, lc};
9 //+
10 Line(1) = {1, 2};
11 Line(2) = {2, 3};
12 Line(3) = {3, 4};
13 Line(4) = {4, 1};
```

```

14 //+
15 Curve Loop(1) = {3, 4, 1, 2};
16
17 xc = 0.5; // x centro da 1a. elipse
18 yc = 0.5; // y centro da 1a. elipse
19 rh = 0.05; // raio horizontal
20 rv = 0.05; // raio vertical
21 kh = 0.2; // permeabilidade horizontal
22 kv = 0.2; // permeabilidade vertical
23 nh = 1; // total de obstaculos horizontais
24 nv = 1; // total de obstaculos verticais
25
26 // elipses horizontais (nh) e verticais (nv)
27 n=5; // iterador
28 count=0;counte=0;countl=2;
29 For v In {0:nv-1}
30 For h In {0:nh-1}
31 //+
32 Point(5+count) = { xc+kh*(h), yc+kv*(v), 0, le};
33 Point(6+count) = {xc+rh+kh*(h), yc+kv*(v), 0, le};
34 Point(7+count) = { xc+kh*(h), yc+rv+kv*(v), 0, le};
35 Point(8+count) = {xc-rh+kh*(h), yc+kv*(v), 0, le};
36 Point(9+count) = { xc+kh*(h), yc-rv+kv*(v), 0, le};
37
38 //+
39 Ellipse(5+(counte)) = {7+count, 5+count, 5+count, 6+count};
40 Ellipse(6+(counte)) = {6+count, 5+count, 5+count, 9+count};
41 Ellipse(7+(counte)) = {9+count, 5+count, 5+count, 8+count};
42 Ellipse(8+(counte)) = {8+count, 5+count, 5+count, 7+count};
43
44 Curve Loop(countl) = {8+counte, 5+counte, 6+counte, 7+counte};
45 count+=5;counte+=4;countl+=1;
46 EndFor
47 EndFor

```

```

48
49 Plane Surface(1) = {1:nh*nv+1};
50
51 Physical Line("inferior", 1) = {1};
52 Physical Line("superior", 2) = {3};
53 Physical Line("esquerda", 3) = {4};
54 Physical Line("direita", 4) = {2};
55 Physical Line("obstaculo", 5) = {5:nv*nh*4+4};
56 //+
57 Physical Surface("surface") = {1};

```

A.2 Script dos casos de verificação

Os códigos *Python* completos utilizado para verificação do código, apresentados no capítulo 4 são escritos descritivamente a seguir. Caso prefira, o leitor pode acessar o arquivo .py do código clicando no ícone de clipe respectivo ao código desejado.

A.2.1 Escoamento de *Lid Driven*



```

1 import meshio
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import matplotlib.tri as mtri
5
6 '''
7 #####
8 #           Parametros de input do Simulador
9 #####
10 '''
11 rho = 1.0           # densidade
12 mu = 0.01          # viscosidade dinamica
13 nu = mu/rho        # viscosidade cinematica

```

```

14 cp = 1.0          # capacidade termica a pressao constante
15 k = 1.0          # condutividade termica
16 alpha = k / (rho * cp) # difusividade termica do material
17
18 V_cc_inf = 0.0    # vorticidade cc inferior
19 V_cc_sup = 0.0    # vorticidade cc superior
20 V_cc_esq = 0.0    # vorticidade cc esquerda
21 V_cc_dir = 0.0    # vorticidade cc direita
22
23 vx_cc_inf = 0.0   # velocidade em x cc inferior
24 vx_cc_sup = 1.0   # velocidade em x cc superior
25 vx_cc_esq = 0.0   # velocidade em x cc esquerda
26 vx_cc_dir = 0.0   # velocidade em x cc direita
27
28 vy_cc_inf = 0.0   # velocidade em y cc inferior
29 vy_cc_sup = 0.0   # velocidade em y cc superior
30 vy_cc_esq = 0.0   # velocidade em y cc esquerda
31 vy_cc_dir = 0.0   # velocidade em y cc direita
32
33 n_inter = 200     # numero de iteracoes
34 dt = 0.01         # intervalo de tempo
35 t = 0.0           # tempo inicial
36
37 '''
38 #####
39 #             LEITOR DE MALHAS e IEN
40 #####
41 '''
42 msh = meshio.read('../cavidade.msh')
43 X = msh.points[:, 0]
44 Y = msh.points[:, 1]
45 IEN = msh.cells['triangle'] # malha bidimensional
46 IENbound = msh.cells['line'] # malha de contorno
47 IENboundTypeElem = list(msh.cell_data['line']['gmsh:physical'] - 1)

```

```

48 boundNames = list(msh.field_data.keys())
49 IENboundElem = [boundNames[elem] for elem in IENboundTypeElem]
50 npoints = len(X) # obtendo o numero de pontos
51 ne = IEN.shape[0] # obtendo o numero de elementos
52
53 # Cria lista de nos do contorno
54 cc = IENbound.reshape(IENbound.size) # lista de nos do contorno
55 ccName = [[] for i in range(len(cc))] # lista com os nomes das cc's
56 for elem in range(len(IENbound)):
57     ccName[IENbound[elem][0]] = IENboundElem[elem]
58     ccName[IENbound[elem][1]] = IENboundElem[elem]
59
60 ## Plot malha e pontos de contorno
61 # plt.triplot(X,Y,IEN,'k-') # plot da malha
62 # plt.plot(X[cc],Y[cc],'b.') # plot dos pontos do contorno
63 # ax = plt.axes()
64 # ax.set_aspect('equal')
65 # plt.show()
66
67 '''
68 #####
69 # Criando Matrizes - K, M, vx, vy, psi, w
70 #####
71 '''
72 K = np.zeros((npoints, npoints), dtype='float') # Matriz K
73 M = np.zeros((npoints, npoints), dtype='float') # Matriz M
74 Gx = np.zeros((npoints, npoints), dtype='float') # Matriz Gx
75 Gy = np.zeros((npoints, npoints), dtype='float') # Matriz Gy
76 vx = np.zeros((npoints), dtype='float') # vetor vx
77 vy = np.zeros((npoints), dtype='float') # vetor vy
78 psi = np.zeros((npoints), dtype='float') # funcao corrente
79 w_z = np.zeros((npoints), dtype = 'float') # funcao vorticidade
80
81 '''

```

```

82 #####
83 #           Jacobiano e area do triangulo
84 #####
85 '''
86 for e in range(0, ne):
87     v = IEN[e]
88
89     # Jacobiano - triangulo
90     jacobian = X[v[2]] * (Y[v[0]] - Y[v[1]]) \
91                 + X[v[0]] * (Y[v[1]] - Y[v[2]]) \
92                 + X[v[1]] * (-Y[v[0]] + Y[v[2]])
93
94     # area do triangulo
95     area = jacobian / 2.0
96
97     '''
98 #####
99     #           Matrizes B (rigidez - Laplaciano) e B transposta
100 #####
101     '''
102     b1 = Y[v[1]] - Y[v[2]]
103     b2 = Y[v[2]] - Y[v[0]]
104     b3 = Y[v[0]] - Y[v[1]]
105
106     c1 = X[v[2]] - X[v[1]]
107     c2 = X[v[0]] - X[v[2]]
108     c3 = X[v[1]] - X[v[0]]
109
110     B = (1.0 / (2.0 * area)) * np.array([[b1, b2, b3],
111                                         [c1, c2, c3]])
112
113     # transposta de B
114     BT = B.transpose()
115

```

```

116     '''
117     #####
118     #           Matrizes k_elemento e m_elemento
119     #####
120     '''
121     kelem = area * np.dot(BT, B)
122
123     melem = (area / 12.0) * np.array([[2.0, 1.0, 1.0],
124                                       [1.0, 2.0, 1.0],
125                                       [1.0, 1.0, 2.0]])
126
127     '''
128     #####
129     #           Matrizes gx_elemento e gy_elemento
130     #####
131     '''
132     gxelem = (1.0 / 6.0) * np.array([[b1, b2, b3],
133                                       [b1, b2, b3],
134                                       [b1, b2, b3]])
135
136     gyelem = (1.0 / 6.0) * np.array([[c1, c2, c3],
137                                       [c1, c2, c3],
138                                       [c1, c2, c3]])
139
140     '''
141     #####
142     #           Matriz A
143     #####
144     '''
145     ## i e j local sao para "passear dentro da "mini" matriz(kelem)
146     ## o global eh para passear dentro da matriz maior (K)
147     for ilocal in range(0, 3): # de zero ao numero de elementos
148         iglobal = IEN[e, ilocal]
149         for jlocal in range(0, 3):

```

```

150         jglobal = IEN[e, jlocal]
151         K[iglobal, jglobal] += alpha * kelem[ilocal, jlocal]
152         M[iglobal, jglobal] += melem[ilocal, jlocal]
153         Gx[iglobal, jglobal] += gxelem[ilocal, jlocal]
154         Gy[iglobal, jglobal] += gyelem[ilocal, jlocal]
155
156     Aw = (M/dt) + (nu * K.copy())
157     Ac = -K.copy()
158     Minv = np.linalg.inv(M)
159
160     for j in cc:
161         # if ccName[j] != 'direita':
162         Aw[j, :] = 0.0 # Zerando as linhas j      # A = Aw
163         Aw[j, j] = 1.0 # Igualando a 1 diagonal
164         Ac[j, :] = 0.0 # Zerando as linhas j
165         Ac[j, j] = 1.0 # Igualando a 1 diagonal
166
167     '''
168     #####
169     #                               Vetor b
170     #####
171     '''
172     b = np.zeros((npoints), dtype='float')
173     bc = np.zeros((npoints), dtype='float')
174
175     ''' CC velocidade '''
176     for i in cc:
177         if ccName[i] == 'inferior':
178             vx[i] = vx_cc_inf
179             vy[i] = vy_cc_inf
180         if ccName[i] == 'superior':
181             vx[i] = vx_cc_sup
182             vy[i] = vy_cc_sup
183         if ccName[i] == 'esquerda':

```

```

184     vx[i] = vx_cc_esq
185     vy[i] = vy_cc_esq
186     if ccName[i] == 'direita':
187         vx[i] = vx_cc_dir
188         vy[i] = vy_cc_dir
189
190 for n in range(1,n_inter): # looping no tempo
191     vxDiag = np.diag(vx)
192     vyDiag = np.diag(vy)
193     vxGx = vxDiag @ Gx
194     vyGy = vyDiag @ Gy
195
196     wp = Minv @ (Gx @ vy - Gy @ vx) # vorticidade (wz) na parede
197
198     b = ((M/dt) ) @ w_z # vetor b, da solucao linear via algelin
199
200     ''' Ajustes das CC no vetor b (vorticidade wz) '''
201     for i in (cc):
202         if ccName[i] == 'inferior':
203             b[i] = wp[i]
204         if ccName[i] == 'superior':
205             b[i] = wp[i]
206         if ccName[i] == 'esquerda':
207             b[i] = wp[i]
208         if ccName[i] == 'direita':
209             b[i] = wp[i]
210         if ccName[i] == 'circulo':
211             b[i] = wp[i]
212
213     Ainv = np.linalg.inv(Aw)
214     w_z = Ainv @ b
215     bc = - M @ w_z
216
217     ''' Ajustes das CC no vetor bc (corrente - psi) '''

```

```

218     for i in cc:
219         if ccName[i] == 'inferior':
220             bc[i] = V_cc_inf
221         if ccName[i] == 'superior':
222             bc[i] = V_cc_sup
223         if ccName[i] == 'esquerda':
224             bc[i] = V_cc_esq
225         if ccName[i] == 'direita':
226             bc[i] = V_cc_dir
227
228     Ac_inv = np.linalg.inv(Ac)
229     psi = Ac_inv @ bc
230     vx = Minv @ (Gy @ psi)
231     vy = Minv @ (-Gx @ psi)
232
233     ''' Ajustes das CC nos vetores vx e vy (velocidades horizontal e
234         vertical) '''
235     for i in cc:
236         if ccName[i] == 'inferior':
237             vx[i] = vx_cc_inf
238             vy[i] = vy_cc_inf
239         if ccName[i] == 'superior':
240             vx[i] = vx_cc_sup
241             vy[i] = vy_cc_sup
242         if ccName[i] == 'esquerda':
243             vx[i] = vx_cc_esq
244             vy[i] = vy_cc_esq
245         if ccName[i] == 'direita':
246             vx[i] = vx_cc_dir
247             vy[i] = vy_cc_dir
248
249     # point_data = {'vx' : vx}
250     # data_vy = {'vy' : vy}
251     # data_psi = { 'psi' : psi}

```

```

251     # data_wzx = {'wzx' : wzx}
252     # point_data.update(data_vy)
253     # point_data.update(data_psi)
254     # point_data.update(data_wzx)
255     # meshio.write_points_cells('solucao-'+str(n)+' .vtk', msh.points,
        msh.cells, point_data=point_data,)
256
257     '''
258     #####
259     #                               GRAFICO
260     #####
261     '''
262     plt.figure(1)
263     triang = mtri.Triangulation(X, Y, IEN)
264     ax = plt.axes()
265     ax.set_aspect('equal')
266     ax.triplot(triang, 'k.-', linewidth=0.5, alpha=0.1)
267     psix = ax.tricontourf(triang, psi, cmap='jet')
268     plt.colorbar(psix)
269     ax.set_title('Corrente (psi)')
270     plt.show()
271
272     plt.figure(2)
273     triang = mtri.Triangulation(X, Y, IEN)
274     ax = plt.axes()
275     ax.set_aspect('equal')
276     ax.triplot(triang, 'k.-', linewidth=0.5, alpha=0.1)
277     vxx = ax.tricontourf(triang, vx, cmap='jet')
278     plt.colorbar(vxx)
279     ax.set_title('Velocidade Horizontal (Vx)')
280     plt.show()
281
282     plt.figure(3)
283     triang = mtri.Triangulation(X, Y, IEN)

```

```

284 ax = plt.axes()
285 ax.set_aspect('equal')
286 ax.triplot(triang, 'k.-', linewidth=0.5, alpha=0.1)
287 vyx = ax.tricontourf(triang, vy, cmap='jet')
288 plt.colorbar(vyx)
289 ax.set_title('Velocidade Vertical (Vy)')
290 plt.show()
291
292 plt.figure(4)
293 triang = mtri.Triangulation(X, Y, IEN)
294 ax = plt.axes()
295 ax.set_aspect('equal')
296 ax.triplot(triang, 'k.-', linewidth=0.5, alpha=0.1)
297 wzx = ax.tricontourf(triang, w_z, cmap='jet')
298 plt.colorbar(wzx)
299 ax.set_title('Vorticidade (wz)')
300 plt.show()

```

A.2.2 Escoamento entre placas planas



```

1 import meshio
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import matplotlib.tri as mtri
5
6 '''
7 #####
8 #           Parametros de input do Simulador
9 #####
10 '''
11 rho = 1.0           # densidade
12 mi = 0.01          # viscosidade dinamica

```

```

13 nu = mi/rho          # viscosidade cinematica
14 cp = 1.0            # capacidade termica a pressao constante
15 k = 1.0             # condutividade termica
16 alpha = k / (rho * cp) # difusividade termica do material
17
18 V_cc_inf = 0.0      # vorticidade cc inferior
19 V_cc_sup = 1.0      # vorticidade cc superior
20
21 vx_cc_inf = 0.0     # velocidade em x cc inferior
22 vx_cc_sup = 0.0     # velocidade em x cc superior
23 vx_cc_esq = 1.0     # velocidade em x cc esquerda
24
25 vy_cc_inf = 0.0     # velocidade em y cc inferior
26 vy_cc_sup = 0.0     # velocidade em y cc superior
27 vy_cc_esq = 0.0     # velocidade em y cc esquerda
28
29 n_inter = 200       # numero de iteracoes
30 dt = 0.01          # intervalo de tempo
31 t = 0.0            # tempo inicial
32
33 '''
34 #####
35 #             LEITOR DE MALHAS e IEN
36 #####
37 '''
38 msh = meshio.read('./retang.msh')
39 X = msh.points[:, 0]
40 Y = msh.points[:, 1]
41 IEN = msh.cells['triangle'] # malha bidimensional
42 IENbound = msh.cells['line'] # malha de contorno
43 IENboundTypeElem = list(msh.cell_data['line']['gmsh:physical'] - 1)
44 boundNames = list(msh.field_data.keys())
45 IENboundElem = [boundNames[elem] for elem in IENboundTypeElem]
46 npoints = len(X) # obtendo o numero de pontos

```

```

47 ne = IEN.shape[0] # obtendo o numero de elementos
48
49 # Cria lista de nos do contorno
50 cc = IENbound.reshape(IENbound.size) # lista de nos do contorno
51 ccName = [[] for i in range(len(cc))] # lista com os nomes das cc's
52 # for elem in range(0, len(cc)):
53 for elem in range(len(IENbound)):
54     ccName[IENbound[elem][0]] = IENboundElem[elem]
55     ccName[IENbound[elem][1]] = IENboundElem[elem]
56
57 ## Plot malha e pontos de contorno
58 # plt.triplot(X,Y,IEN,'k-')      # plot da malha
59 # plt.plot(X[cc],Y[cc],'b.')    # plot dos pontos do contorno
60 # ax = plt.axes()
61 # ax.set_aspect('equal')
62 # plt.show()
63
64 '''
65 #####
66 #           Criando Matrizes - K, M, vx, vy, psi, w
67 #####
68 '''
69 K = np.zeros((npoints, npoints), dtype='float') # Matriz K
70 M = np.zeros((npoints, npoints), dtype='float') # Matriz M
71 Gx = np.zeros((npoints, npoints), dtype='float') # Matriz Gx
72 Gy = np.zeros((npoints, npoints), dtype='float') # Matriz Gy
73 vx = np.zeros((npoints), dtype='float')         # vetor vx
74 vy = np.zeros((npoints), dtype='float')         # vetor vy
75 psi = np.zeros((npoints), dtype='float')        # funcao corrente
76 w_z = np.zeros((npoints), dtype = 'float')     # funcao vorticidade
77
78 '''
79 #####
80 #           Jacobiano e area do triangulo

```

```

81 #####
82 '''
83 for e in range(0, ne):
84     v = IEN[e]
85
86     # Jacobiano - triangulo
87     jacobian = X[v[2]] * (Y[v[0]] - Y[v[1]]) \
88               + X[v[0]] * (Y[v[1]] - Y[v[2]]) \
89               + X[v[1]] * (-Y[v[0]] + Y[v[2]])
90
91     # area do triangulo
92     area = jacobian / 2.0
93
94     '''
95 #####
96     #           Matrizes B (rigidez - Laplaciano) e B transposta
97     #####
98     '''
99     b1 = Y[v[1]] - Y[v[2]]
100    b2 = Y[v[2]] - Y[v[0]]
101    b3 = Y[v[0]] - Y[v[1]]
102
103    c1 = X[v[2]] - X[v[1]]
104    c2 = X[v[0]] - X[v[2]]
105    c3 = X[v[1]] - X[v[0]]
106
107    B = (1.0 / (2.0 * area)) * np.array([[b1, b2, b3],
108                                         [c1, c2, c3]])
109
110    # transposta de B
111    BT = B.transpose()
112
113    '''
114 #####

```

```

115 #           Matrizes k_elemento e m_elemento
116 #####
117 '''
118 kelem = area * np.dot(BT, B)
119
120 melem = (area / 12.0) * np.array([[2.0, 1.0, 1.0],
121                                   [1.0, 2.0, 1.0],
122                                   [1.0, 1.0, 2.0]])
123
124 '''
125 #####
126 #           Matrizes gx_elemento e gy_elemento
127 #####
128 '''
129 gxelem = (1.0 / 6.0) * np.array([[b1, b2, b3],
130                                   [b1, b2, b3],
131                                   [b1, b2, b3]])
132
133 gyelem = (1.0 / 6.0) * np.array([[c1, c2, c3],
134                                   [c1, c2, c3],
135                                   [c1, c2, c3]])
136
137 '''
138 #####
139 #           Matriz A
140 #####
141 '''
142 ## i e j local sao para "passear dentro da "mini" matriz(kelem)
143 ## o global eh para passear dentro da matriz maior (K)
144 for ilocal in range(0, 3): # de zero ao numero de elementos
145     iglobal = IEN[e, ilocal]
146     for jlocal in range(0, 3):
147         jglobal = IEN[e, jlocal]
148         K[iglobal, jglobal] += alpha * kelem[ilocal, jlocal]

```

```

149         M[iglobal, jglobal] += melem[ilocal, jlocal]
150         Gx[iglobal, jglobal] += gxelem[ilocal, jlocal]
151         Gy[iglobal, jglobal] += gyelem[ilocal, jlocal]
152
153     Aw = (M/dt) + (nu * K.copy())
154     Ac = -K.copy()
155     Minv = np.linalg.inv(M)
156
157     for j in cc:
158         if ccName[j] != 'direita':
159             Aw[j, :] = 0.0 # Zerando as linhas j      # A = Aw
160             Aw[j, j] = 1.0 # Igualando a 1 diagonal
161             Ac[j, :] = 0.0 # Zerando as linhas j
162             Ac[j, j] = 1.0 # Igualando a 1 diagonal
163
164     '''
165     #####
166     #                               Vetor b
167     #####
168     '''
169     b = np.zeros((npoints), dtype='float')
170     bc = np.zeros((npoints), dtype='float')
171
172     ''' CC velocidade '''
173     for i in cc:
174         if ccName[i] == 'inferior':
175             vx[i] = vx_cc_inf
176             vy[i] = vy_cc_inf
177         if ccName[i] == 'superior':
178             vx[i] = vx_cc_sup
179             vy[i] = vy_cc_sup
180         if ccName[i] == 'esquerda':
181             vx[i] = vx_cc_esq
182             vy[i] = vy_cc_esq

```

```

183
184 for n in range(1,n_inter): # looping no tempo
185     vxDiag = np.diag(vx)
186     vyDiag = np.diag(vy)
187     vxGx = vxDiag @ Gx
188     vyGy = vyDiag @ Gy
189
190     wp = Minv @ (Gx @ vy - Gy @vx) # vorticidade (wz) na parede
191
192     b = ((M/dt) ) @ w_z
193
194     ''' Ajustes das CC no vetor b (vorticidade wz) '''
195     for i in (cc):
196         if ccName[i] == 'inferior':
197             b[i] = wp[i]
198         if ccName[i] == 'superior':
199             b[i] = wp[i]
200         if ccName[i] == 'esquerda':
201             b[i] = wp[i]
202
203     Ainv = np.linalg.inv(Aw)
204     w_z = Ainv @ b
205     bc = - M @ w_z
206
207     ''' Ajustes das CC no vetor bc (corrente) '''
208     for i in cc:
209         if ccName[i] == 'inferior':
210             bc[i] = V_cc_inf
211         if ccName[i] == 'superior':
212             bc[i] = V_cc_sup
213         if ccName[i] == 'esquerda':
214             bc[i] = Y[i]
215
216     Ac_inv = np.linalg.inv(Ac)

```

```

217     psi = Ac_inv @ bc
218     vx = Minv @ (Gy @ psi)
219     vy = Minv @ (-Gx @ psi)
220
221     ''' Ajustes das CC nos vetores vx e vy (velocidades horizontal e
222         vertical) '''
223     for i in cc:
224         if ccName[i] == 'inferior':
225             vx[i] = vx_cc_inf
226             vy[i] = vy_cc_inf
227         if ccName[i] == 'superior':
228             vx[i] = vx_cc_sup
229             vy[i] = vy_cc_sup
230         if ccName[i] == 'esquerda':
231             vx[i] = vx_cc_esq
232             vy[i] = vy_cc_esq
233
234     # point_data = {'vx' : vx}
235     # data_vy = {'vy' : vy}
236     # data_psi = { 'psi' : psi}
237     # data_wzx = {'wzx' : wxz}
238     # point_data.update(data_vy)
239     # point_data.update(data_psi)
240     # point_data.update(data_wzx)
241     # meshio.write_points_cells('solucao-'+str(n)+' .vtk', msh.points,
242     #                             msh.cells, point_data=point_data,)
243
244     '''
245     #####
246     #                               GRAFICO
247     #####
248     '''
249     plt.figure(1)
250     triang = mtri.Triangulation(X, Y, IEN)

```

```

249 ax = plt.axes()
250 ax.set_aspect('equal')
251 ax.triplot(triang, 'k.-', linewidth=0.5, alpha=0.1)
252 psix = ax.tricontourf(triang, psi, cmap='jet')
253 plt.colorbar(psix)
254 ax.set_title('Corrente (psi)')
255 plt.show()
256
257 plt.figure(2)
258 triang = mtri.Triangulation(X, Y, IEN)
259 ax = plt.axes()
260 ax.set_aspect('equal')
261 ax.triplot(triang, 'k.-', linewidth=0.5, alpha=0.1)
262 vxx = ax.tricontourf(triang, vx, cmap='jet')
263 plt.colorbar(vxx)
264 ax.set_title('Velocidade Horizontal (Vx)')
265 plt.show()
266
267 plt.figure(3)
268 triang = mtri.Triangulation(X, Y, IEN)
269 ax = plt.axes()
270 ax.set_aspect('equal')
271 ax.triplot(triang, 'k.-', linewidth=0.5, alpha=0.1)
272 vyx = ax.tricontourf(triang, vy, cmap='jet')
273 plt.colorbar(vyx)
274 ax.set_title('Velocidade Vertical (Vy)')
275 plt.show()
276
277 plt.figure(4)
278 triang = mtri.Triangulation(X, Y, IEN)
279 ax = plt.axes()
280 ax.set_aspect('equal')
281 ax.triplot(triang, 'k.-', linewidth=0.5, alpha=0.1)
282 wzx = ax.tricontourf(triang, w_z, cmap='jet')

```

```

283 plt.colorbar(wzx)
284 ax.set_title('Vorticidade (wz)')
285 plt.show()

```

A.3 Script dos casos de estudo

O código completo utilizado para os estudos apresentados no capítulo 5 é escrito descritivamente a seguir. Caso prefira, o leitor pode acessar o arquivo .py do código clicando no ícone de clipe, a seguir.



```

1  '''
2  #####
3  #                               IMPORTS
4  #####
5  '''
6  import meshio
7  import numpy as np
8  import matplotlib.pyplot as plt
9  import matplotlib.tri as mtri
10 import time
11
12 start = time.time() # Inicia a contagem de tempo de processamento
13
14 '''
15 #####
16 #                               PARAMETROS DE INPUT DO SIMULADOR
17 #####
18 '''
19 rho = 1.0           # densidade
20 mu = 0.1           # viscosidade dinamica
21 nu = mu/rho        # viscosidade cinematica
22 cp = 1.0           # capacidade termica a pressao constante

```

```

23 k = 1.0                # condutividade termica
24 alpha = k / (rho * cp) # difusividade termica do material
25
26 ### Condiçoes de contorno do dominio ###
27 c_inf = 0.0           # cc inferior
28 c_sup = 1.0           # cc superior
29 c_centro = 0.5        # cc centro
30
31 ### Iteracoes ###
32 n_inter = 500         # numero de iteracoes
33 dt = 0.001            # intervalo de tempo
34 t = 0.0               # tempo inicial
35
36 '''
37 #####
38 #                       LEITOR DE MALHAS e IEN
39 #####
40 '''
41 msh = meshio.read('../MP-10x3-46509n-93170e.msh') # <- Inserir nome do
      arquivo .msh da malha desejada
42 X = msh.points[:, 0]
43 Y = msh.points[:, 1]
44 IEN = msh.cells[1].data
45 IENbound = msh.cells[0].data
46 IENboundTypeElem = list(msh.cell_data['gmsh:physical'][0] - 1)
47 boundNames = list(msh.field_data.keys())
48 IENboundElem = [boundNames[elem] for elem in IENboundTypeElem]
49 npoints = len(X) # obtendo o numero de pontos
50 ne = IEN.shape[0] # obtendo o numero de elementos
51
52 ### Cria lista de nos do contorno ###
53 cc = IENbound.reshape(IENbound.size) # lista de nos do contorno
54 ccName = [[] for i in range(len(cc))] # lista com os nomes das cc's
55 for elem in range(len(IENbound)):

```

```

56     ccName[IENbound[elem][0]] = IENboundElem[elem]
57     ccName[IENbound[elem][1]] = IENboundElem[elem]
58
59     ### Plot malha e pontos de contorno ###
60     # plt.triplot(X,Y,IEN,'k-')           # plot da malha
61     # plt.plot(X[cc],Y[cc],'b.')         # plot dos pontos do contorno
62     # ax = plt.axes()
63     # ax.set_aspect('equal')
64     # plt.show()
65
66     '''
67     #####
68     #             CRIANDO MATRIZES - K, M, vx, vy, psi, w
69     #####
70     '''
71     K = np.zeros((npoints, npoints), dtype='float') # Matriz K
72     M = np.zeros((npoints, npoints), dtype='float') # Matriz M
73     Gx = np.zeros((npoints, npoints), dtype='float') # Matriz Gx
74     Gy = np.zeros((npoints, npoints), dtype='float') # Matriz Gy
75     vx = np.zeros((npoints), dtype='float') # vetor vx
76     vy = np.zeros((npoints), dtype='float') # vetor vy
77     psi = np.zeros((npoints), dtype='float') # funcao corrente
78     w_z = np.zeros((npoints), dtype = 'float') # funcao vorticidade
79
80     '''
81     #####
82     #             JACOBIANO E AREA DO TRIANGULO
83     #####
84     '''
85     for e in range(0, ne):
86         v = IEN[e]
87
88         ### Jacobiano - triangulo ###
89         jacobian = X[v[2]] * (Y[v[0]] - Y[v[1]]) \

```

```

90         + X[v[0]] * (Y[v[1]] - Y[v[2]]) \
91         + X[v[1]] * (-Y[v[0]] + Y[v[2]])
92
93     ### area do triangulo ###
94     area = jacobian / 2.0
95
96     '''
97     #####
98     #           MATRIZES B (RIGIDEZ - LAPLACIANO) e B TRANSPOSTA
99     #####
100    '''
101    b1 = Y[v[1]] - Y[v[2]]
102    b2 = Y[v[2]] - Y[v[0]]
103    b3 = Y[v[0]] - Y[v[1]]
104
105    c1 = X[v[2]] - X[v[1]]
106    c2 = X[v[0]] - X[v[2]]
107    c3 = X[v[1]] - X[v[0]]
108
109    B = (1.0 / (2.0 * area)) * np.array([[b1, b2, b3],
110                                         [c1, c2, c3]])
111
112    ### transposta de B ###
113    BT = B.transpose()
114
115    '''
116    #####
117    #           MATRIZES k_elemento e m_elemento
118    #####
119    '''
120    kelem = area * np.dot(BT, B)
121
122    melem = (area / 12.0) * np.array([[2.0, 1.0, 1.0],
123                                     [1.0, 2.0, 1.0],

```

```

124         [1.0, 1.0, 2.0]))
125
126     '''
127     #####
128     #           MATRIZES gx_elemento e gy_elemento
129     #####
130     '''
131     gxelem = (1.0 / 6.0) * np.array([[b1, b2, b3],
132                                     [b1, b2, b3],
133                                     [b1, b2, b3]])
134
135     gyelem = (1.0 / 6.0) * np.array([[c1, c2, c3],
136                                     [c1, c2, c3],
137                                     [c1, c2, c3]])
138
139     '''
140     #####
141     #           Matriz A
142     #####
143     '''
144     ## i e j local sao para percorrer dentro da "mini" matriz(k_elem)
145     ## o global eh para percorrer dentro da matriz maior (K)
146     for ilocal in range(0, 3): # de zero ao numero de elementos
147         iglobal = IEN[e, ilocal]
148         for jlocal in range(0, 3):
149             jglobal = IEN[e, jlocal]
150             K[iglobal, jglobal] += alpha * kelem[ilocal, jlocal]
151             M[iglobal, jglobal] += melem[ilocal, jlocal]
152             Gx[iglobal, jglobal] += gxelem[ilocal, jlocal]
153             Gy[iglobal, jglobal] += gyelem[ilocal, jlocal]
154
155     Aw = (M/dt) + (nu * K.copy()) # Matriz A da vorticidade
156     Ac = -K.copy()                # Matriz A da vorticidade
157     Minv = np.linalg.inv(M)

```

```

158
159 for j in cc:
160     Aw[j, :] = 0.0 # Zerando as linhas j      # A = Aw
161     Aw[j, j] = 1.0 # Igualando a 1 diagonal
162     Ac[j, :] = 0.0 # Zerando as linhas j
163     Ac[j, j] = 1.0 # Igualando a 1 diagonal
164
165 '''
166 #####
167 #                               VETOR b
168 #####
169 '''
170 b = np.zeros((npoints), dtype='float')
171 bc = np.zeros((npoints), dtype='float')
172
173 ### CC velocidade ###
174 ## 'inferior', 'superior', 'esquerda', 'direita' e 'obstaculo' sao os
    nomes dos limites do dominio que foram atribuidos no GMSH na
    construcao da malha
175 for i in cc:
176     if ccName[i] == 'inferior':
177         vx[i] = 0.0
178         vy[i] = 0.0
179     if ccName[i] == 'superior':
180         vx[i] = 0.0
181         vy[i] = 0.0
182     if ccName[i] == 'esquerda':
183         vx[i] = 1.0
184         vy[i] = 0.0
185     ## Este if esta comentado devido a ser condicao de Neuman
186     # if ccName[i] == 'direita':
187     #     vx[i] = 1.0
188     #     vy[i] = 0.0
189     if ccName[i] == 'obstaculo':

```

```

190     vx[i] = 0.0
191     vy[i] = 0.0
192
193 for n in range(1,n_inter):           # looping no tempo
194     vxDiag = np.diag(vx)
195     vyDiag = np.diag(vy)
196     vxGx = vxDiag @ Gx
197     vyGy = vyDiag @ Gy
198
199     wp = Minv @ (Gx @ vy - Gy @vx) # wz na parede
200
201     b = ((M/dt) @ w_z) - (vxGx @ w_z) - (vyGy @ w_z)
202     ### Ajustes das CC no vetor b (vorticidade w_z) ###
203     for i in cc:
204         b[i] = wp[i]
205
206     Ainv = np.linalg.inv(Aw)
207     w_z = Ainv @ b
208
209     bc = - M @ w_z
210     ### Ajustes das CC no vetor bc (corrente) ###
211     for i in cc:
212         if ccName[i] == 'inferior':
213             bc[i] = c_inf
214         if ccName[i] == 'superior':
215             bc[i] = c_sup
216         if ccName[i] == 'esquerda':
217             bc[i] = Y[i]
218         if ccName[i] == 'direita':
219             bc[i] = Y[i]
220
221     '''
222     ### SEMPRE VERIFICAR ESSA CONDICAO DE CONTORNO ###
223     '''

```

```

224     if ccName[i] == 'obstaculo':
225         if Y[i] > 0.07 and Y[i] < 0.23: # range de altura do obstaculo
226             bc[i] = 0.15                # valor do centro do obstaculo
227         if Y[i] > 0.43 and Y[i] < 0.58: # range de altura do obstaculo
228             bc[i] = 0.5                # valor do centro do obstaculo
229         if Y[i] > 0.77 and Y[i] < 0.93: # range de altura do obstaculo
230             bc[i] = 0.85                # valor do centro do obstaculo
231
232     Ac_inv = np.linalg.inv(Ac)
233     psi = Ac_inv @ bc
234
235     vx = Minv @ (Gy @ psi)
236     vy = Minv @ (-Gx @ psi)
237
238     ### Ajustes das CC nos vetores vx e vy ###
239     ## 'inferior', 'superior', 'esquerda', 'direita' e 'obstaculo' sao os
240     ## nomes dos limites do dominio que foram atribuidos no GMSH na
241     ## construcao da malha
242
243     for i in cc:
244         if ccName[i] == 'inferior':
245             vx[i] = 0.0
246             vy[i] = 0.0
247         if ccName[i] == 'superior':
248             vx[i] = 0.0
249             vy[i] = 0.0
250         if ccName[i] == 'esquerda':
251             vx[i] = 1.0
252             vy[i] = 0.0
253         ## Este if esta comentado devido a ser condicao de Neuman
254         # if ccName[i] == 'direita':
255         #     vx[i] = 1.0
256         #     vy[i] = 0.0
257         if ccName[i] == 'obstaculo':
258             vx[i] = 1.0

```

```

256         vy[i] = 0.0
257
258     '''
259     #####
260     #           SALVANDO A SOLUCAO EM .vtk
261     #####
262     '''
263     point_data = {'vx': vx}
264     data_vy = {'vy': vy}
265     data_psi = {'psi': psi}
266     data_wzx = {'wzx': w_z}
267     point_data.update(data_vy)
268     point_data.update(data_psi)
269     point_data.update(data_wzx)
270     meshio.write_points_cells('E:\TCC - Solucao\Meio Poroso\MP\MP10x3-' +
        str(n) + '.vtk', msh.points, msh.cells, point_data=point_data, )
        ## <- Apontar o caminho da pasta onde deseja salvar os resultados,
        e o nome dos arquivos gerados
271
272     '''
273     #####
274     #           GRAFICO MATPLOTLIB - SOLUCAO FINAL
275     #####
276     '''
277     ### Corrente ###
278     plt.figure(1)
279     triang = mtri.Triangulation(X, Y, IEN)
280     ax = plt.axes()
281     ax.set_aspect('equal')
282     ax.triplot(triang, 'k.-', linewidth=0.5, alpha=0.1)
283     psix = ax.tricontourf(triang, psi, cmap='jet')
284     plt.colorbar(psix)
285     ax.set_title('Corrente (psi)')
286     plt.show()

```

```

287
288 ### V_x ###
289 plt.figure(2)
290 triang = mtri.Triangulation(X, Y, IEN)
291 ax = plt.axes()
292 ax.set_aspect('equal')
293 ax.triplot(triang, 'k.-', linewidth=0.5, alpha=0.1)
294 vxx = ax.tricontourf(triang, vx, cmap='jet')
295 plt.colorbar(vxx)
296 ax.set_title('Velocidade Horizontal (Vx)')
297 plt.show()
298
299 ### V_y ###
300 plt.figure(3)
301 triang = mtri.Triangulation(X, Y, IEN)
302 ax = plt.axes()
303 ax.set_aspect('equal')
304 ax.triplot(triang, 'k.-', linewidth=0.5, alpha=0.1)
305 vyx = ax.tricontourf(triang, vy, cmap='jet')
306 plt.colorbar(vyx)
307 ax.set_title('Velocidade Vertical (Vy)')
308 plt.show()
309
310 ### Vorticidade ###
311 plt.figure(4)
312 triang = mtri.Triangulation(X, Y, IEN)
313 ax = plt.axes()
314 ax.set_aspect('equal')
315 ax.triplot(triang, 'k.-', linewidth=0.5, alpha=0.1)
316 wzx = ax.tricontourf(triang, w_z, cmap='jet')
317 plt.colorbar(wzx)
318 ax.set_title('Vorticidade (w_z)')
319 plt.show()
320

```

```
321 '''
322 #####
323 #                 TEMPO DE EXECUCAO
324 #####
325 '''
326 print("O tempo de execucao do codigo foi: ")
327 end = time.time()
328 print(end - start, '[s]')
```