

UNIVERSIDADE FEDERAL DE JUIZ DE FORA  
INSTITUTO DE CIÊNCIAS EXATAS  
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

***R<sub>e</sub>FUM2D: Uma Ferramenta de Volumes Finitos para a Simulação das Equações de Navier-Stokes em C++***

**Ricardo Ervilha Silva**

JUIZ DE FORA  
JANEIRO, 2026

# **$R_e$ FUM2D: Uma Ferramenta de Volumes Finitos para a Simulação das Equações de Navier-Stokes em C++**

**RICARDO ERVILHA SILVA**

UNIVERSIDADE FEDERAL DE JUIZ DE FORA  
INSTITUTO DE CIÊNCIAS EXATAS  
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO  
GRADUAÇÃO em CIÊNCIA DA COMPUTAÇÃO

Orientador: José Jerônimo Camata

JUIZ DE FORA  
JANEIRO, 2026

*R<sub>e</sub>FUM2D: UMA FERRAMENTA DE VOLUMES FINITOS  
PARA A SIMULAÇÃO DAS EQUAÇÕES DE NAVIER-STOKES  
EM C++*

Ricardo Ervilha Silva

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS  
EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTE-  
GRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE  
BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.

Aprovada por:

José Jerônimo Camata  
Professor e Doutor em Engenharia Civil/Computação em Alto Desempenho  
(COPPE-UFRJ)

Joventino de Oliveira Campos  
Professor e Doutor em Modelagem Computacional (PPGMC-UFJF)

Iury Higor Aguiar da Igreja  
Professor e Doutor em Modelagem Computacional (LNCC)

JUIZ DE FORA  
22 DE JANEIRO, 2026

*Dedico este trabalho aos familiares, amigos e professores que me ajudaram a discretizar cada momento difícil em passos menores e mais estáveis. Afinal, sem estabilidade numérica, nada avança... .*

## Resumo

Este trabalho apresenta o desenvolvimento do  $R_e$ FUM2D, um solucionador bidimensional de Dinâmica dos Fluidos Computacional, implementado em C++ e baseado no Método dos Volumes Finitos aplicado a malhas não estruturadas. O principal objetivo é a simulação de escoamentos governados pelas equações de Navier–Stokes. São descritas as estratégias de discretização das equações governantes e o funcionamento do algoritmo SIMPLE, amplamente utilizado em CFD. Adicionalmente, o trabalho aborda os principais aspectos computacionais do ReFUM2D, incluindo a organização do código, o pré-processamento das malhas, o cálculo de grandezas geométricas e a exportação dos resultados. A validação do solver é realizada por meio de benchmarks clássicos da literatura, cujos resultados apresentam boa concordância com dados de referência, evidenciando a robustez e a acurácia do código como ferramenta didática e de pesquisa em CFD.

**Palavras-chave:** Dinâmica de Fluidos Computacional, Métodos Numéricos, Método dos Volumes Finitos, Equações Diferenciais Parciais, Navier-Stokes.

## Abstract

This work presents the development of *R<sub>e</sub>FUM2D*, a two-dimensional Computational Fluid Dynamics solver implemented in C++ and based on the Finite Volume Method applied to unstructured meshes. The main objective is the simulation of flows governed by the Navier–Stokes equations. The strategies adopted for the discretization of the governing equations and the implementation of the SIMPLE algorithm, widely used in CFD, are described. In addition, the work addresses the main computational aspects of ReFUM2D, including code organization, mesh pre-processing, computation of geometric quantities, and result export. The solver validation is performed through classical benchmarks from the literature, whose results show good agreement with reference data, demonstrating the robustness and accuracy of the code as a didactic and research tool in CFD.

**Keywords:** Computational Fluid Dynamics, Numerical Methods, Finite Volume Method, Partial Differential Equations, Navier–Stokes.

## Agradecimentos

Primeiramente, agradeço à minha família, em especial aos meus pais, Joana e Ronaldo, por todo o apoio, paciência e incentivo ao longo de toda a minha trajetória acadêmica. Sou imensamente grato pelo esforço, dedicação e por nunca medirem esforços para que eu tivesse acesso à melhor educação possível.

Agradeço também às pessoas extraordinárias com quem tive o privilégio de conviver e aprender ao longo dos cinco anos de graduação. Obrigado pelas amizades construídas, pelas trocas de conhecimento e pelos momentos de descontração que tornaram essa caminhada mais leve e prazerosa.

Expresso ainda minha gratidão a todos os professores com quem tive contato durante o curso e ao Departamento de Ciência da Computação, pelos ensinamentos, orientações e contribuições fundamentais para a minha formação acadêmica e profissional.

Por fim, agradeço ao professor José Camata, pelo apoio, dedicação e orientação ao longo do desenvolvimento deste trabalho, bem como à Universidade Federal de Juiz de Fora, pela infraestrutura e oportunidades oferecidas durante minha formação.

*“O conhecido é finito, o desconhecido, infinito; intelectualmente permanecemos em uma ilhota dentro de um oceano ilimitado de inexplicabilidade. Nossa objetivo em todas as gerações é reivindicar por um pouco mais de terra.”*

*T. H. Huxley, 1887.*

# Conteúdo

<b>Lista de Figuras</b>	<b>8</b>
<b>Lista de Tabelas</b>	<b>10</b>
<b>Lista de Abreviações</b>	<b>11</b>
<b>1 INTRODUÇÃO</b>	<b>12</b>
<b>2 DESCRIÇÃO DO PROBLEMA FÍSICO</b>	<b>14</b>
2.1 Introdução . . . . .	14
2.2 Escoamentos . . . . .	14
2.3 Equações Governantes . . . . .	15
2.3.1 Conservação de Massa . . . . .	16
2.3.2 Conservação de Momento . . . . .	18
2.3.3 Conclusão . . . . .	22
<b>3 METODOLOGIA NUMÉRICA</b>	<b>24</b>
3.1 Método dos Volumes Finitos . . . . .	24
3.2 Malhas não estruturadas . . . . .	26
3.3 Volumes de controle em malhas não estruturadas . . . . .	27
3.4 Introdução ao SIMPLE . . . . .	28
3.5 Discretização da Equação de Continuidade . . . . .	29
3.6 Discretização das Equações de Momento . . . . .	31
3.6.1 Discretização do termo difusivo . . . . .	32
3.6.2 Discretização do termo advectivo . . . . .	41
3.6.3 Discretização da Pressão . . . . .	46
3.6.4 Forma final da equação de momento . . . . .	47
3.7 Tratamento do acoplamento pressão-velocidade . . . . .	49
3.7.1 Interpolação de Rhie-Chow . . . . .	50
3.8 Derivação da equação para pressão . . . . .	53
3.9 Coeficientes de relaxação . . . . .	55
3.10 Tratamento das condições de contorno . . . . .	56
<b>4 DESCRIÇÃO DO CÓDIGO <i>R<sub>e</sub>FUM2D</i></b>	<b>58</b>
4.1 Processamento das informações do problema . . . . .	59
4.2 Pré-processamento . . . . .	61
4.2.1 Cálculo do volume das células . . . . .	62
4.2.2 Cálculo das áreas das faces e informações sobre as normais . . . . .	63
4.3 Processamento . . . . .	64
4.4 Exportação da solução . . . . .	67
<b>5 RESULTADOS</b>	<b>68</b>
5.1 Fluxo de Kovasznay . . . . .	68
5.2 Fluxo na cavidade acionado pela tampa . . . . .	70
5.3 Problema do escoamento sobre um degrau . . . . .	73
5.4 Problema do fluxo ao redor de um cilindro . . . . .	77

<b>6 CONCLUSÕES E TRABALHOS FUTUROS</b>	<b>80</b>
<b>Bibliografia</b>	<b>82</b>

# Lista de Figuras

2.1	Elemento de fluido utilizado para derivar as equações. . . . .	15
2.2	Elemento de fluido com a orientação e direção dos fluxos de massa. . . . .	17
2.3	Elemento de fluido com as forças na direção $x$ exibidas. Imagem retirada de (VERSTEEG; MALALASEKERA, 2007). . . . .	19
2.4	(a) Tensão cisalhante, responsável por deformar o corpo e alterar seu formato e (b) tensão normal a qual pode comprimir ou expandir o volume do elemento. Imagem retirada de (ANDERSON, 1995). . . . .	20
3.1	Volume de controle utilizado para balanços de conservação. Imagem retirada de (MALISKA, 2004). . . . .	24
3.2	Exemplo de malha não estruturada para simular o escoamento de fluidos ao redor de um aerofólio. Imagem retirada de (HASSELBACHER, 1999). . .	27
3.3	Volumes de controle em malhas não estruturadas 2D: (a) centrado na célula; (b) centrado no nó. Imagem retirada de (VERSTEEG; MALALASEKERA, 2007). . . . .	27
3.4	Stencil de uma malha não estruturada com elementos triangulares. Na imagem, o volume de referência está com rótulo $O$ , enquanto os numerados de 1 até 3 são seus vizinhos. Os vetores $\vec{n}$ são as normais de cada face da célula $O$ . . . . .	29
3.5	Escolha de vetores $\vec{n}_1$ e $\vec{n}_2$ para decompor o gradiente difusivo. Na imagem também está retratado o vetor normal a face $\vec{n}_f$ . $d_{ON}$ representa a distância entre a célula $O$ e seu vizinho $N$ . . . . .	34
3.6	Volume de controle $O$ , e seus vizinhos rotulados de 1 até 4. . . . .	36
3.7	Exemplo de uma malha em (a) e exemplo de um volume de controle disponível nessa malha em (b), retratando seus vizinhos e ângulos formados. .	38
3.8	Gráficos da solução da equação 3.36. A figura (a) representa a solução com $\theta = 0$ , enquanto a solução da figura (b) representa a solução com $\theta = 15$ . Ambas as figuras foram obtidas com a malha de 8192 volumes de controle. .	40
3.9	Comparação da convergência variando o número de correções. . . . .	40
3.10	Comparação de convergência entre UD e LUD. . . . .	45
3.11	Figura exibindo o domínio, retratando também a velocidade $\vec{u} = (1, 1)$ . O segmento em vermelho é utilizado para comparar as soluções. Imagem retirada de: <a href="https://www.openfoam.com/documentation/guides/latest/doc/guide-schemes-divergence-example.html">https://www.openfoam.com/documentation/guides/latest/doc/guide-schemes-divergence-example.html</a> . . . . .	46
3.12	Comparação entre UD e LUD na diagonal, em que a função é $\phi(x, 1 - x)$ . Imagem obtida com refinamento de 16384 volumes de controle em uma malha estruturada composta por quadriláteros. . . . .	46
3.13	Ilustração do <i>checkerboading</i> , onde existe oscilação na pressão. Imagem retirada de (VERSTEEG; MALALASEKERA, 2007). . . . .	49
4.1	Visão geral dos componentes. . . . .	58
4.2	Diagrama de classes do $R_e$ FUM2D. . . . .	59
4.3	Exemplo de um arquivo contendo as informações de entrada do problema. Por praticidade, quebrou-se a visualização em duas partes, sendo a primeira metade na esquerda. . . . .	60

4.4	Exemplo de malha não estruturada bidimensional. Os vértices da malha são destacados em azul, os centroides dos volumes de controle em vermelho e as faces em roxo. As regiões de contorno do domínio ( <i>Top</i> , <i>Bottom</i> , <i>Left</i> , <i>Right</i> ) estão explicitamente identificadas. Imagem criada com base em (MAZUMDER, 2015). . . . .	62
4.5	Fluxograma ilustrando visualmente as etapas de processamento. . . . .	67
5.1	Histórico de convergência do método SIMPLE: (a) convergência da velocidade e (b) convergência da pressão. . . . .	69
5.2	Domínio, malha e informações de contorno do problema da cavidade. . . .	70
5.3	Problema da cavidade $Re = 100$ : (a) magnitude das velocidades e (b) linhas de corrente. . . . .	71
5.4	Problema da cavidade $Re = 1000$ : (a) magnitude das velocidades e (b) linhas de corrente. . . . .	71
5.5	Comparação com (GHIA; GHIA; SHIN, 1982) em $Re = 100$ : (a) perfil vertical e (b) perfil horizontal. . . . .	72
5.6	Comparação com (GHIA; GHIA; SHIN, 1982) em $Re = 1000$ : (a) perfil vertical e (b) perfil horizontal. . . . .	72
5.7	Comparação da convergência para ambos os Reynolds, comparando o valor da norma do máximo e a quantidade de iterações para o problema da cavidade: (a) $R_e = 100$ (b) $R_e = 1000$ . . . . .	73
5.8	Domínio, malha e informações de contorno do problema do degrau. . . .	73
5.9	Comparação da convergência para $R_e = 100$ , comparando o valor da norma do máximo e a quantidade de iterações para o problema do degrau. . . .	75
5.10	Magnitude da velocidade e linhas de corrente do problema do degrau para $R_e = 10$ . . . . .	75
5.11	Magnitude da velocidade e linhas de corrente do problema do degrau para $R_e = 50$ . . . . .	75
5.12	Magnitude da velocidade e linhas de corrente do problema do degrau para $R_e = 100$ . . . . .	76
5.13	Comparação da qualidade da solução comparando com (ARMALY et al., 1983) em $R_e = 100$ : (a) $x/h = 3.06$ (b) $x/h = 6.12$ . . . . .	76
5.14	Domínio, malha e informações de contorno do problema do cilindro. . . .	77
5.15	Comparação da convergência para $R_e = 20$ , comparando o valor da norma do máximo e a quantidade de iterações para o problema do cilindro. . . .	78
5.16	Magnitude da velocidade e linhas de corrente do <i>flow over a cylinder</i> para $R_e = 20$ . . . . .	78

## **Lista de Tabelas**

- 5.1 Comparação entre os limites inferior e máximo de diversos trabalhos reportados em (SCHÄFER et al., 1996) e a solução obtida neste trabalho. . . . . 79

## Lista de Abreviações

P&D	Pesquisa e Desenvolvimento
CFD	<i>Computational Fluid Dynamics</i>
MVF	Método dos Volumes Finitos
EDP	Equação Diferencial Parcial
GPU	<i>Graphics Processing Unit</i>
UD	<i>Upwind Differencing</i>
LUD	<i>Linear Upwind Differencing</i>
BiCGSTAB	<i>Biconjugate gradient stabilized method</i>
VTK	<i>Visualization Toolkit</i>
SIMPLE	<i>Semi-Implicit Method for Pressure Linked Equations</i>
PWIM	<i>Pressure-Weighted Momentum Interpolation</i>
YAML	<i>YAML Ain't Markup Language</i>

# 1 INTRODUÇÃO

Com o vertiginoso avanço dos computadores de alta velocidade e dispositivos de armazenamento, o desenvolvimento de algoritmos voltados à resolução dos mais diversos tipos de problemas tem se tornado um campo de grande interesse para centros P&D ao redor do mundo (MALISKA, 2004). Esse progresso tem impulsionado, em especial, a área da Dinâmica dos Fluidos Computacional (CFD), que se dedica à análise de sistemas envolvendo escoamento de fluidos, transferência de calor e outros fenômenos correlatos que podem ser modelados e simulados por meio de recursos computacionais (VERSTEEG; MALALASEKERA, 2007).

Na área de simulação numérica, os métodos tradicionais para resolução das Equações Diferenciais Parciais (EDPs) são: método de Diferenças Finitas (MDF), método de Elementos Finitos (MEF) e método de Volumes Finitos (MVF). Historicamente, o MDF foi sempre utilizado em mecânica de fluidos, enquanto o MEF foi voltado para a área de problemas estruturais (MALISKA, 2004). Nesse sentido, até a década de 1970, o MDF destacava-se pela sua experiência consolidada na simulação de escoamentos, embora apresentasse limitações na representação de geometrias complexas, enquanto o MEF mostrava-se mais adequado para tratar tais geometrias, mas ainda carecia de ferramentas eficazes para lidar com termos não lineares e o acoplamento de equações característicos da mecânica dos fluidos. Tais problemas levaram ao aprimoramento do MVF, no qual as relações são obtidas por meio de balanços de volumes elementares (MOUKALLED; MANGANI; DARWISH, 2015).

O resultado desse balanço é conseguir expressar de forma exata a conservação das propriedades relevantes para cada célula. Essa relação entre o método e o princípio físico subjacente forma um dos principais atrativos para o MVF, tornando-o mais simples e robusto para lidar com geometrias não-estruturadas, o que permite o cálculo dos fluxos ao redor de regiões complexas sem precisar gastar muito tempo gerando malhas (DING; SUN, 2013). Como apontado por (MALISKA, 2004), no contexto de pacotes comerciais, o MVF também é o mais utilizado e com ampla penetração industrial, visto que é muito

importante satisfazer os princípios de conservação ao nível discreto, não existindo problemas de gerações/sumidouros no interior do domínio de cálculo. A depuração também fica mais fácil, visto que, como há um balanço para todos os volumes de controle, para qualquer tamanho de malha, todos os princípios de conservação devem ser satisfeitos, mesmo nas malhas mais grosseiras (CHÉNIER; EYMARD; TOUAZI, 2006). Nos outros dois métodos, a solução pode ser conferida somente em malhas bem refinadas, já que não existe o conceito de balanços em volumes elementares.

Neste contexto, o presente trabalho apresenta o software *R<sub>e</sub>FUM2D* (*R*esolution of *F*inite-*v*olume *d*iscretizations on *U*nstructured *M*eshes), um conjunto de códigos desenvolvido em C++ para a resolução de problemas de CFD bidimensionais por meio do método dos volumes finitos em malhas não estruturadas. O objetivo principal do *R<sub>e</sub>FUM2D* é a simulação de escoamentos governados pelas equações de Navier–Stokes incompressíveis e estacionárias, buscando reproduzir de forma robusta e confiável resultados e aplicações em problemas de interesse. Diferentemente de plataformas de grande porte amplamente utilizadas, como OpenFOAM (WELLER et al., 1998) e Ansys (ANSYS, 2016), o desenvolvimento de um solucionador próprio oferece maior liberdade para adaptação, depuração e inovações, permitindo controle direto sobre todas as etapas do algoritmo. Essa abordagem evita as limitações impostas por abstrações internas e pela complexidade inerente a softwares generalistas. Assim, o *R<sub>e</sub>FUM2D* tem como proposta não ser apenas uma ferramenta de simulação, mas também uma base didática e de pesquisa, voltada ao entendimento aprofundado dos métodos de CFD e à experimentação de novas abordagens.

Este trabalho está organizado da seguinte forma. Na Seção 2, apresenta-se a descrição do problema físico em estudo, com a dedução das equações de Navier–Stokes a partir das leis de conservação de massa e momento. A Seção 3 descreve a metodologia numérica adotada, discutindo os principais aspectos matemáticos envolvidos na resolução de problemas de CFD através do MVF. Na Seção 4, são apresentados a organização e os aspectos computacionais do *R<sub>e</sub>FUM2D*, incluindo a descrição de seus principais componentes. Em seguida, a Seção 5 discute os resultados obtidos a partir de *benchmarks* da literatura, com o objetivo de validar a acurácia e a robustez do *solver*. Por fim, a Seção 6 reúne as considerações finais e as perspectivas para trabalhos futuros.

## 2 DESCRIÇÃO DO PROBLEMA FÍSICO

### 2.1 Introdução

Este capítulo apresenta os fundamentos matemáticos do modelo de dinâmica de fluidos, baseados nos princípios de conservação de massa e momento, que sustentam a dedução das equações governantes e a análise das principais propriedades do problema.

### 2.2 Escoamentos

Fluidos, no qual se denotam líquidos ou gases, são substâncias que não mudam permanentemente sob a ação de forças. Ao contrário dos sólidos, que quando submetidos a forças conseguem resistir a deformação, fluidos não conseguem resistir a tais tensões e são colocados em movimento. Em função disso, enquanto sólidos possuem um formato bem definido, fluidos não tem um formato definitivo e se ajustam conforme o ambiente que os cerca (MOUKALLED; MANGANI; DARWISH, 2015).

Um primeiro ponto importante quando se analisa escoamentos de fluidos, é que o foco está no ponto de vista macroscópico, analisando propriedades tais como velocidade, pressão e temperatura. Isso pode ser pensado como uma média sobre uma quantidade considerável de moléculas. Desse ponto de vista, um ponto no fluido seria o menor elemento possível do fluido no qual as propriedades macroscópicas não são influenciadas pelas moléculas individuais que o compõem (VERSTEEG; MALALASEKERA, 2007).

Outra hipótese importante é os fluidos serem assumidos um meio “continuum”, de tal modo que as propriedades físicas são definidas para todos os pontos no espaço. Através de tal hipótese, fluidos podem ser categorizados como Newtonianos ou Não-Newtonianos. Fluidos Newtonianos são caracterizados por uma relação linear entre tensões de cisalhamento e taxas de cisalhamento, com a viscosidade molecular  $\mu$ , a qual mede a habilidade do fluido resistir à deformação quando submetido a forças externas, sendo o coeficiente angular da relação. Em contrapartida, fluidos Não-Newtonianos possuem um comporta-

mento não linear mais complexo de ser modelado.

Os fluidos também podem ser classificados como unidimensional ou multidimensional (2 ou mais dimensões), fase única ou multifásica, estacionário ou não estacionário, real (viscoso) ou ideal (invíscido), compressível ou incompressível, turbulento ou laminar, rotacional ou irrotacional, entre outros, os quais serão melhor discutidos conforme necessidade posterior. A proposta dessas classificações é simplificar o processo de análise e descrição do fenômeno físico (MALISKA, 2004).

## 2.3 Equações Governantes

Será considerado para as demonstrações das equações governantes o caso tridimensional. Dessa forma, um pequeno elemento de fluido com lados  $\Delta x$ ,  $\Delta y$  e  $\Delta z$  está representado na Figura 2.1, onde as seis faces nomeadas com  $W$ ,  $E$ ,  $S$ ,  $N$ ,  $B$  e  $T$  significam respectivamente:

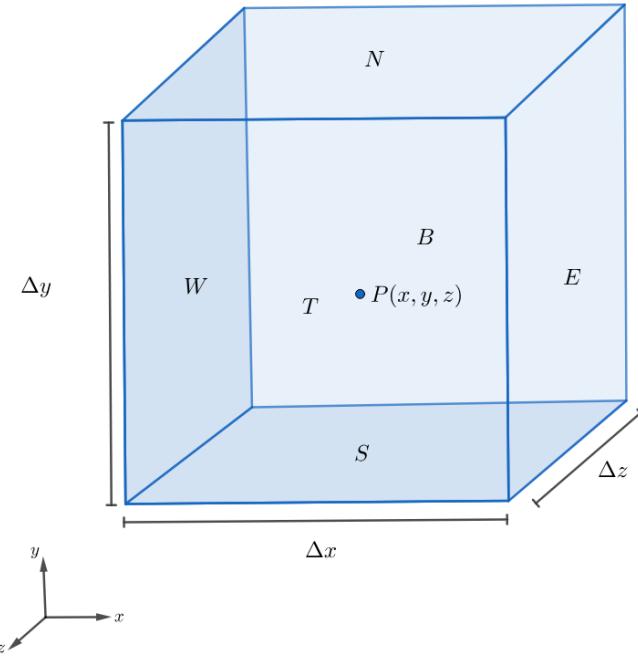


Figura 2.1: Elemento de fluido utilizado para derivar as equações.

mente: oeste (*west*), leste (*east*), sul (*south*), norte (*north*), fundo (*bottom*) e topo (*top*). O centro do elemento é o ponto  $P$ . Todas as propriedades são funções de espaço e tempo, como  $\rho(x, y, z, t)$ ,  $p(x, y, z, t)$  e  $\vec{u}(x, y, z, t)$  para densidade, pressão e vetor velocidade. Por questões de simplicidade, se estiver no texto apenas  $p$ ,  $\rho$  ou  $\vec{u}$ , considera-se que o valor

já é relacionado ao ponto  $(x, y, z)$  no tempo  $t$ . Por fim, o elemento em consideração é pequeno o suficiente para considerar que as propriedades nas faces são aproximados de forma precisa pelos primeiros dois termos da expansão de Taylor.

A seguir encontram-se as demonstrações para as equações de Navier-Stokes, as quais regem o movimento de fluidos. Tais equações têm como base fundamental as leis físicas de conservação de massa e momento. As deduções seguem a abordagem proposta por (VERSTEEG; MALALASEKERA, 2007), a qual tenta, a partir de pressupostos intuitivos, deduzir cada uma das equações.

### 2.3.1 Conservação de Massa

De acordo com Lavoisier, o princípio de conservação de massa estabelece que, na ausência de fontes ou sumidouros de massa, o volume deverá conservar sua massa. Em outras palavras, a taxa de variação da massa no elemento de fluido é igual à taxa líquida do que entra e sai no elemento de fluido. A taxa de variação de massa no elemento de fluido pode ser expressa por:

$$\frac{\partial \rho}{\partial t} \Delta x \Delta y \Delta z. \quad (2.1)$$

Agora, precisa-se avaliar a taxa de variação do fluxo de massa ao longo das faces do elemento, a qual será dada pelo produto da densidade, área e velocidade normal as faces. Baseado na Figura 2.1, tal taxa pode ser expressa como:

$$\begin{aligned} & \left( \rho u - \frac{\partial(\rho u)}{\partial x} \frac{1}{2} \Delta x \right) \Delta y \Delta z - \left( \rho u + \frac{\partial(\rho u)}{\partial x} \frac{1}{2} \Delta x \right) \Delta y \Delta z \\ & + \left( \rho v - \frac{\partial(\rho v)}{\partial y} \frac{1}{2} \Delta y \right) \Delta x \Delta z - \left( \rho v + \frac{\partial(\rho v)}{\partial y} \frac{1}{2} \Delta y \right) \Delta x \Delta z \\ & + \left( \rho w - \frac{\partial(\rho w)}{\partial z} \frac{1}{2} \Delta z \right) \Delta x \Delta y - \left( \rho w + \frac{\partial(\rho w)}{\partial z} \frac{1}{2} \Delta z \right) \Delta x \Delta y. \end{aligned} \quad (2.2)$$

Ao qual os fluidos que tem direção entrando no elemento produzem um aumento de massa e tem sinal positivo e os que saem possuem um sinal negativo. A Figura 2.2 ilustra com maiores detalhes o significado da equação 2.2.

Portanto, a taxa de variação de massa no elemento é agora igual à taxa de variação do

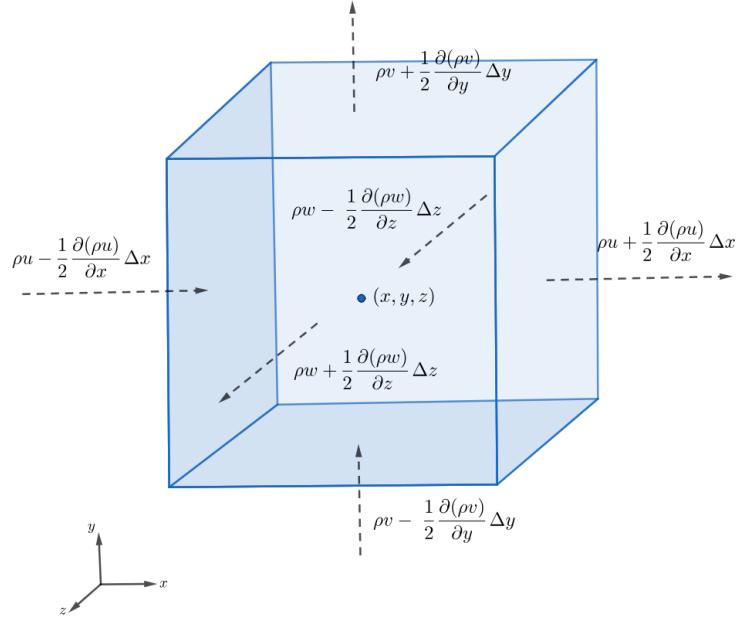


Figura 2.2: Elemento de fluido com a orientação e direção dos fluxos de massa.

fluxo de massa ao longo de cada uma das suas faces. Assim, tem-se que:

$$\begin{aligned} & \left( \rho u - \frac{\partial(\rho u)}{\partial x} \frac{1}{2} \Delta x \right) \Delta y \Delta z - \left( \rho u + \frac{\partial(\rho u)}{\partial x} \frac{1}{2} \Delta x \right) \Delta y \Delta z \\ & + \left( \rho v - \frac{\partial(\rho v)}{\partial y} \frac{1}{2} \Delta y \right) \Delta x \Delta z - \left( \rho v + \frac{\partial(\rho v)}{\partial y} \frac{1}{2} \Delta y \right) \Delta x \Delta z \\ & + \left( \rho w - \frac{\partial(\rho w)}{\partial z} \frac{1}{2} \Delta z \right) \Delta x \Delta y - \left( \rho w + \frac{\partial(\rho w)}{\partial z} \frac{1}{2} \Delta z \right) \Delta x \Delta y = \frac{\partial \rho}{\partial t} \Delta x \Delta y \Delta z. \end{aligned} \quad (2.3)$$

Simplificando alguns termos e agrupando outros, chega-se em:

$$-\frac{\partial(\rho u)}{\partial x} \Delta x \Delta y \Delta z - \frac{\partial(\rho v)}{\partial y} \Delta x \Delta y \Delta z - \frac{\partial(\rho w)}{\partial z} \Delta x \Delta y \Delta z = \frac{\partial \rho}{\partial t} \Delta x \Delta y \Delta z. \quad (2.4)$$

Dividindo ambos os lados da equação acima por  $\Delta x \Delta y \Delta z$  (volume do paralelepípedo) e passando os termos do lado esquerdo para o lado direito:

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} + \frac{\partial(\rho w)}{\partial z} = 0. \quad (2.5)$$

Ao qual em formato mais compacto e usual pode ser denotada como:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{u}) = 0, \quad (2.6)$$

em que  $\vec{u} = (u, v, w)$ . A equação 2.6 é a versão não estacionária (por conter um termo

dependente do tempo) da equação de conservação de massa, também conhecida como **equação da continuidade** em um fluido compressível.

Por fim, para um fluido **incompressível**, a densidade  $\rho$  não depende da posição ou tempo, sendo, portanto, uma constante. Sendo assim, a equação 2.6 se torna:

$$\nabla \cdot \vec{u} = 0. \quad (2.7)$$

A equação acima diz: para fluidos incompressíveis a taxa de variação do fluxo em qualquer elemento de fluido é zero, ou seja: “quantidade que sai” = “quantidade que entra”.

### 2.3.2 Conservação de Momento

O princípio do balanço do momento linear (ou 2<sup>a</sup> Lei de Newton) estabelece que:

“A taxa de variação do momento linear de um determinado conjunto de partículas é igual à soma vetorial de todas as forças externas atuando sobre ele.”

As taxas de aumento do momento em  $x$ ,  $y$  e  $z$  por unidade de área de uma partícula de fluido são dadas por:

$$\rho \frac{Du}{Dt}, \rho \frac{Dv}{Dt}, \rho \frac{Dw}{Dt} \quad (2.8)$$

onde  $\frac{D\phi}{Dt}$  de um campo  $\phi(t, \vec{x}(t))$ , o qual pode ser escalar ou vetorial, e que represente velocidade, densidade ou temperatura; representa a **derivada material**. A derivada material pode ser obtida aplicando a regra da cadeia para considerar as mudanças induzidas por todas as variáveis independentes ao longo de um caminho:

$$\begin{aligned} \frac{D\phi}{Dt} &= \frac{\partial\phi}{\partial t} \frac{dt}{dt} + \frac{\partial\phi}{\partial x} \frac{dx}{dt} + \frac{\partial\phi}{\partial y} \frac{dy}{dt} \\ &= \frac{\partial\phi}{\partial t} + u \frac{\partial\phi}{\partial x} + v \frac{\partial\phi}{\partial y} \\ &= \frac{\partial\phi}{\partial t} + \vec{u} \cdot \nabla \phi. \end{aligned} \quad (2.9)$$

Em que a equação 2.9 representa a taxa de variação de  $\phi$  quando se acompanha uma partícula de fluido em movimento. Diferente da derivada parcial, que olha a variação

em um ponto fixo do espaço, a derivada material calcula a variação na perspectiva do referencial que viaja com o fluido.

Em uma partícula de fluido, dois tipos de forças podem atuar: forças de superfície (pressão, tensões cisalhantes e tensões normais) as quais agem diretamente na superfície do elemento de fluido e forças de corpo (força elétrica e força gravitacional) as quais atuam diretamente na massa volumétrica do elemento de fluido (“atuam a distância”). Considerando a direção  $x$ , dispõem-se em um elemento de fluido as forças retratadas na Figura 2.3, que servirá como alicerce para derivar a equação de momento em  $x$ .

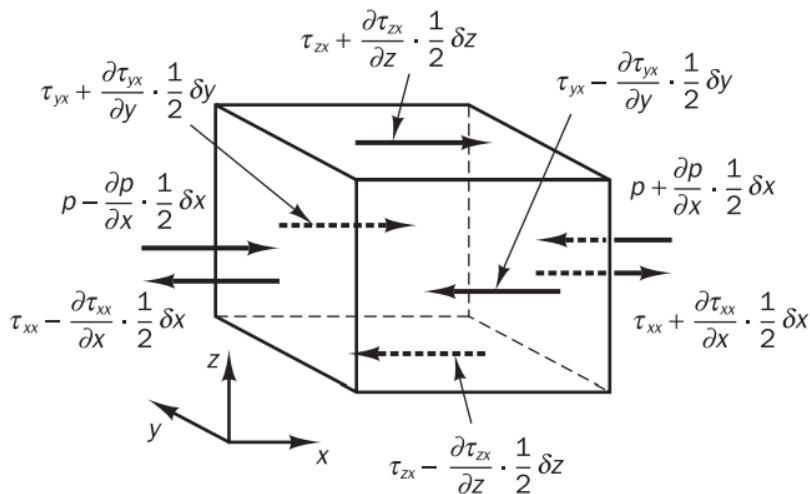


Figura 2.3: Elemento de fluido com as forças na direção  $x$  exibidas. Imagem retirada de (VERSTEEG; MALALASEKERA, 2007).

As tensões cisalhantes e tensões normais atuam na superfície do fluido puxando ou empurrando a superfície por meio da fricção. Exemplos das duas são encontrados na Figura 2.4. Em fluidos viscosos, a tensão normal é muito menor que a tensão cisalhante e muitas vezes é negligenciada. Tensões normais se tornam importantes quando o gradiente das velocidades normais são grandes, como experienciado em ondas de choque (ANDERSON, 1995).

Denotando a força de corpo que age no elemento por  $\vec{f}$  e sendo  $f_x$  sua componente horizontal, a força de corpo que atua no elemento em  $x$  é dada por:

$$\rho(\Delta x \Delta y \Delta z) f_x. \quad (2.10)$$

Na figura 2.3, a convenção é que forças de tensão são denotadas por  $\tau$ , em que

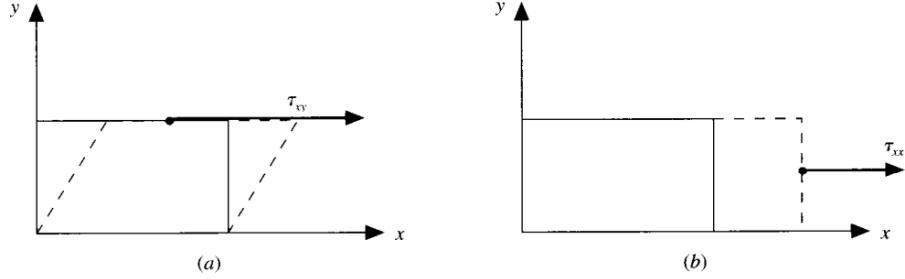


Figura 2.4: (a) Tensão cisalhante, responsável por deformar o corpo e alterar seu formato e (b) tensão normal a qual pode comprimir ou expandir o volume do elemento. Imagem retirada de (ANDERSON, 1995).

$\tau_{ij}$  é a tensão na direção  $j$  exercida em um plano perpendicular a direção  $i$ . Com isso, pode-se fazer um somatório das forças que atuam em cada um dos pares de faces. No caso das faces  $E$  e  $W$  tem-se que:

$$\begin{aligned} & \left[ \left( p - \frac{\partial p}{\partial x} \frac{\Delta x}{2} \right) - \left( \tau_{xx} - \frac{\partial \tau_{xx}}{\partial x} \frac{\Delta x}{2} \right) \right] \Delta y \Delta z + \left[ - \left( p + \frac{\partial p}{\partial x} \frac{\Delta x}{2} \right) + \left( \tau_{xx} + \frac{\partial \tau_{xx}}{\partial x} \frac{\Delta x}{2} \right) \right] \Delta y \Delta z \\ &= \left( -\frac{\partial p}{\partial x} + \frac{\partial \tau_{xx}}{\partial x} \right) \Delta x \Delta y \Delta z. \end{aligned} \quad (2.11)$$

Para as faces  $N$  e  $S$ :

$$-\left( \tau_{yx} - \frac{\partial \tau_{yx}}{\partial y} \frac{\Delta y}{2} \right) \Delta x \Delta z + \left( \tau_{yx} + \frac{\partial \tau_{yx}}{\partial y} \frac{\Delta y}{2} \right) \Delta x \Delta z = \frac{\partial \tau_{yx}}{\partial y} \Delta x \Delta y \Delta z. \quad (2.12)$$

E por fim para  $T$  e  $B$ :

$$-\left( \tau_{zx} - \frac{\partial \tau_{zx}}{\partial z} \frac{\Delta z}{2} \right) \Delta x \Delta y + \left( \tau_{zx} + \frac{\partial \tau_{zx}}{\partial z} \frac{\Delta z}{2} \right) \Delta x \Delta y = \frac{\partial \tau_{zx}}{\partial z} \Delta x \Delta y \Delta z. \quad (2.13)$$

Ficando que a força total na direção  $x$  ( $F_x$ ), considerando a soma das equações 2.10 a 2.13, é igual a:

$$F_x = \left[ -\frac{\partial p}{\partial x} + \frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z} \right] \Delta x \Delta y \Delta z + \rho f_x \Delta x \Delta y \Delta z. \quad (2.14)$$

Denotando a aceleração do fluido em  $x$  como  $a_x$ , a mesma é igual à taxa de variação de  $u$  ao longo do tempo. Uma vez que acompanha-se o movimento de um fluido,

essa taxa é dada justamente pela sua derivada material. Logo:

$$a_x = \frac{Du}{Dt}. \quad (2.15)$$

Como, pela Segunda Lei de Newton,  $F_x = ma_x$ , pode-se igualar 2.14 com 2.15 para chegar em:

$$\begin{aligned} & \left[ -\frac{\partial p}{\partial x} + \frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z} \right] \Delta x \Delta y \Delta z + \rho f_x \Delta x \Delta y \Delta z = \rho \Delta x \Delta y \Delta z \frac{Du}{Dt} \\ & \Rightarrow \rho \frac{Du}{Dt} = -\frac{\partial p}{\partial x} + \frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z} + \rho f_x. \end{aligned} \quad (2.16)$$

Onde está implícito que  $m = \rho \Delta x \Delta y \Delta z$ . Assim, a equação obtida em 2.16 é a equação de momento em  $x$  para um fluido viscoso. Similarmente, equações para  $y$  e  $z$  podem ser obtidas:

$$\begin{aligned} \rho \frac{Dv}{Dt} &= -\frac{\partial p}{\partial y} + \frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \tau_{yy}}{\partial y} + \frac{\partial \tau_{zy}}{\partial z} + \rho f_y, \\ \rho \frac{Dw}{Dt} &= -\frac{\partial p}{\partial z} + \frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{yz}}{\partial y} + \frac{\partial \tau_{zz}}{\partial z} + \rho f_z. \end{aligned} \quad (2.17)$$

As equações de (2.16) e (2.17), juntamente à equação (2.6), são chamadas de **Equações de Navier-Stokes**, descobertas de forma independente na metade do século 19. No entanto, as equações de momento ainda contêm as componentes de tensão  $\tau_{ij}$ , desconhecidas. Para torná-las mais úteis na descrição do comportamento dos fluidos, é necessário introduzir um modelo para representar essas tensões. Na maioria dos fluidos de interesse, tais tensões podem ser expressas como funções da taxa de deformação. Para não alongar-se muito mais na discussão, e introduzir eventuais complexidades desnecessárias, optou-se por omitir o passo-a-passo para lidar com o conjunto de tensões.

Sendo assim, pode-se rearranjar os termos viscosos (maiores detalhes em (VERSTEEG; MALALASEKERA, 2007)) tomando como hipóteses o fluido ser incompressível e newtoniano, para tornar a formulação mais útil para o método de volumes finitos:

$$\rho \frac{Du}{Dt} = -\frac{\partial p}{\partial x} + \nabla \cdot (\mu \nabla u) + \rho f_x. \quad (2.18)$$

$$\rho \frac{Dv}{Dt} = -\frac{\partial p}{\partial y} + \nabla \cdot (\mu \nabla v) + \rho f_y. \quad (2.19)$$

$$\rho \frac{Dw}{Dt} = -\frac{\partial p}{\partial z} + \nabla \cdot (\mu \nabla w) + \rho f_z. \quad (2.20)$$

### 2.3.3 Conclusão

Limitando-se às equações (2.18) até (2.20), em conjunto com a equação da continuidade incompressível (2.7), em duas dimensões apenas e também em regime estacionário, pode-se apresentar o conjunto de equações de interesse deste trabalho:

$$\begin{cases} \nabla \cdot (\rho u \vec{u}) = -\frac{\partial p}{\partial x} + \nabla \cdot (\mu \nabla u) + S_x, \\ \nabla \cdot (\rho v \vec{u}) = -\frac{\partial p}{\partial y} + \nabla \cdot (\mu \nabla v) + S_y, \\ \nabla \cdot \vec{u} = 0. \end{cases} \quad (2.21)$$

Em que foram explicitados os termos do operador de derivada material, e as forças de corpo foram tratadas apenas como um termo fonte  $S$ . No próximo capítulo, serão abordados as estratégias adotadas para discretizar esse sistema de equações. Um último adendo, é a possibilidade de adimensionalizar as equações, deixando-as em função somente de um número adimensional, denotado por Número de Reynolds ( $R_e$ ), definido por:

$$R_e = \frac{\rho u L}{\mu}. \quad (2.22)$$

Em que  $u$  [ $m/s$ ] é a velocidade característica do escoamento, tipicamente semelhante à velocidade média, e  $L$  [m] é o comprimento característico, que representa uma dimensão de referência que depende da geometria do problema. Ademais,  $\rho$  [ $Kg/m^3$ ] é a densidade do fluido e  $\mu$  [ $Pa \cdot s$ ] é a viscosidade dinâmica. O número de Reynolds representa fisicamente a razão entre as forças iniciais e as forças viscosas que atuam no fluido, e é muito útil, ao permitir comparar sistemas que são fundamentalmente os mesmos, variando somente suas escalas. Outra utilidade deste é poder determinar se o escoamento é caracterizado como laminar ou turbulento. Por questões de simplicidade, neste trabalho o foco é apenas em escoamentos laminares, no qual os valores de Reynolds costumam ser menores e as partículas do fluido possuem comportamento mais ordenado e previsível.

Para o tratamento de turbulência, a modelagem matemática costuma ser rigorosamente mais complexa, além do gasto computacional ser mais acentuado, motivos impeditivos para considerá-la neste momento.

## 3 METODOLOGIA NUMÉRICA

Neste capítulo será apresentado a metodologia numérica utilizada para a resolução das equações de Navier-Stokes (2.21), focando nos seus aspectos matemáticos. Inicialmente, introduz-se o método de volumes finitos para aproximação numérica, e em seguidas são discutidas as etapas de discretização que servirão como base do solucionador desenvolvido.

### 3.1 Método dos Volumes Finitos

De início, todo método que, para obter as equações aproximadas, satisfaz a conservação de uma propriedade ao nível de volumes elementares é classificado como um **método de volumes finitos**. Para se obter as equações desse método, existem duas abordagens principais. Na primeira, realizam-se balanços da propriedade em questão em cada volume elementar. Na segunda, integram-se, no espaço e no tempo, as equações diferenciais escritas na **forma conservativa** sobre o volume elementar (MALISKA, 2004). A forma conservativa (também conhecida como forma divergente) é aquela em que, na equação diferencial, os fluxos aparecem dentro do operador divergente. Nisso, após a primeira integração, surgem explicitamente os fluxos nas fronteiras do volume elementar, o que a torna equivalente a um balanço de propriedade.

Para ilustrar os procedimentos apresentados, será utilizado o exemplo de (MALISKA, 2004), mostrado na Figura 3.1. O interesse é obter uma equação que represente

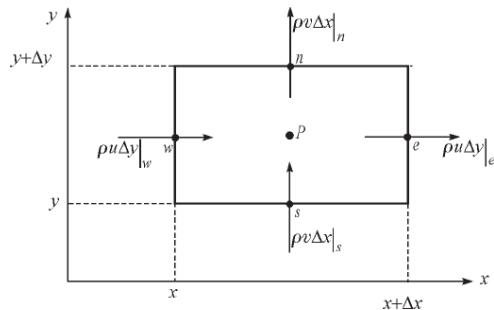


Figura 3.1: Volume de controle utilizado para balanços de conservação. Imagem retirada de (MALISKA, 2004).

a conservação de massa. Fazendo um balanço nesse elemento, obtém-se que:

$$\rho u \Delta y|_e - \rho u \Delta y|_w + \rho v \Delta x|_n - \rho v \Delta x|_s = 0, \quad (3.1)$$

onde as letras  $e$ ,  $w$ ,  $n$  e  $s$  representam as direções leste, oeste, norte e sul. Dividindo a equação pelo volume do elemento  $\Delta x \Delta y$  (em que 2D o volume referido seria a área do retângulo), encontra-se:

$$\frac{(\rho u|_e - \rho u|_w)}{\Delta x} + \frac{(\rho v|_n - \rho v|_s)}{\Delta y} = 0. \quad (3.2)$$

No limite quando  $\Delta x \rightarrow 0$  e  $\Delta y \rightarrow 0$ , obtém-se exatamente a equação da continuidade (2.6) na forma estacionária:

$$\frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} = 0. \quad (3.3)$$

A equação acima está em forma conservativa, visto que as variáveis estão dentro dos operadores de derivada. Pode-se então, utilizando a segunda etapa, obter de outra forma a equação de conservação de massa através da integração da equação acima no volume de controle elementar. Assim, isso resultaria em:

$$\int_w^e \int_s^n \left[ \frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} \right] dy dx = 0. \quad (3.4)$$

A partir desse ponto, utiliza-se inicialmente o fato de que a integral de uma soma é igual à soma das integrais. Em seguida, aplica-se o Teorema de Fubini para inverter a ordem de integração na primeira parcela, permitindo integrar primeiro em  $e$  e  $w$ . Dessa forma, obtém-se o seguinte resultado:

$$\int_s^n [\rho u|_e - \rho u|_w] dy + \int_w^e [\rho v|_n - \rho v|_s] dx = 0. \quad (3.5)$$

Como o volume de controle analisado é infinitesimal, pode-se considerar que o fluxo de massa avaliado no meio das faces representa a média de variação do mesmo ao longo de todos os pontos. Outra forma de visualizar isso, é que o valor médio de uma propriedade  $\phi$  em um volume é dado por:

$$\bar{\phi} = \frac{1}{V} \int_V \phi dV, \quad (3.6)$$

onde  $V$  denota o volume de controle sobre o qual a média é computada. Sendo assim, consegue-se:

$$\rho u \Delta y|_e - \rho u \Delta y|_w + \rho v \Delta x|_n - \rho v \Delta x|_s = 0. \quad (3.7)$$

Observe agora que a equação acima é idêntica à Equação (3.1). Desse modo, realizar o balanço ou a integração na forma conservativa são processos equivalentes. No entanto, de forma prática, há uma preferência por integrar as equações em forma conservativa, visto que os balanços não são sempre triviais de se deduzir. Conforme relatado por (MALISKA, 2004), para o balanço da quantidade de movimento, é necessário identificar e fazer o somatório de todas as tensões que atuam no volume elementar, sendo tarefa difícil para volumes irregulares.

## 3.2 Malhas não estruturadas

O foco neste trabalho é gerar simulações para malhas não estruturadas. Malhas não estruturadas (Figura 3.2) são a forma mais geral de arranjo para geometrias mais complexas. Oferecem grande flexibilidade tanto na escolha dos tipos de elementos quanto na forma de distribuí-los espacialmente, permitindo refinar regiões de maior interesse. Assim, volumes de controle podem ter qualquer formato e pode-se misturá-los, formando malhas híbridas (VERSTEEG; MALALASEKERA, 2007). Essa flexibilidade, no entanto, vem ao custo de uma complexidade adicional. Em um sistema não estruturado, não há forma direta de conectar várias entidades juntas baseadas somente em seus índices. Assim, a conectividade tem de ser definida explicitamente, não havendo forma de adivinhar o índice de nós, faces ou elementos vizinhos a partir de um índice de referência como pode ser feito em malhas estruturadas (MOUKALLED; MANGANI; DARWISH, 2015).

A parte mais atrativa das malhas não estruturadas é poder calcular o fluxo ao redor de geometrias variadas, sem gastar muito tempo em geração de malha e mapeamento entre sistemas de coordenadas, gerando diversos termos adicionais na discretização que acabariam por complicar a modelagem numérica.

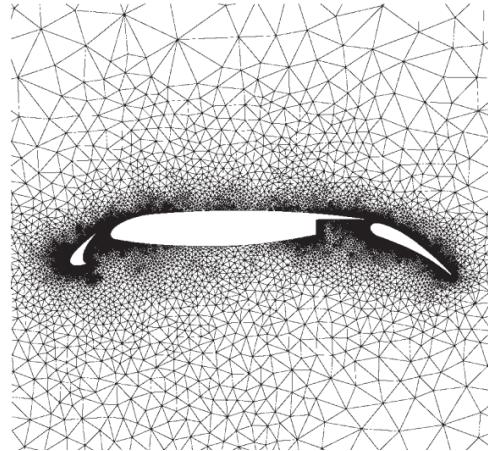


Figura 3.2: Exemplo de malha não estruturada para simular o escoamento de fluidos ao redor de um aerofólio. Imagem retirada de (HASELBACHER, 1999).

### 3.3 Volumes de controle em malhas não estruturadas

Uma grande questão dentro do MVF é decidir como serão gerados os volumes de controle a partir da malha. Há duas formas principais de se definir os volumes de controle em malhas não estruturadas: volumes de controle centrados na célula (*cell-center*) e volumes de controle centrados nos nós/vértices (*vertex-center*). Tais variantes estão ilustradas na Figura 3.3.

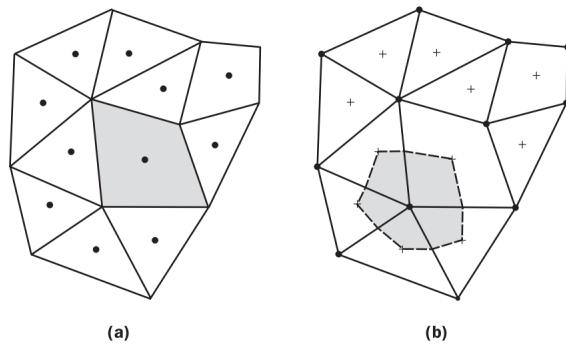


Figura 3.3: Volumes de controle em malhas não estruturadas 2D: (a) centrado na célula; (b) centrado no nó. Imagem retirada de (VERSTEEG; MALALASEKERA, 2007).

Na abordagem centrada na célula, as variáveis de interesse são colocadas no centroide do volume de controle, previamente definido pelo gerador de malhas, como ilustrado na Figura 3.3(a). No método centrado no nó, por sua vez, as variáveis são associadas aos vértices do arranjo. Nesse caso, é necessário realizar um processo de tesselação, em que os centróides dos elementos adjacentes são conectados de modo a formar subvolumes em torno de cada nó, definindo assim os volumes de controle onde

serão aplicadas as discretizações, conforme mostrado na Figura 3.3(b).

A abordagem *cell-center* destaca-se por seu entendimento mais simples e por ser amplamente documentada na literatura. Além disso, pode reduzir o consumo de memória, uma vez que as variáveis primárias são armazenadas por volume de controle (isto é, por célula, no centróide), evitando o armazenamento nodal; a economia efetiva depende da malha e do modelo considerado, mas o custo de armazenamento das incógnitas escala diretamente com o número de volumes de controle. De forma geral, ambas as abordagens são utilizadas em métodos de volumes finitos; no entanto, em *softwares* de CFD amplamente utilizados, como **OpenFOAM** e **Ansys Fluent**, adota-se a abordagem *cell-center*, razão pela qual ela também foi escolhida neste trabalho.

## 3.4 Introdução ao SIMPLE

Considere novamente as Equações apresentadas em (2.21). Uma primeira característica importante é que elas são não-lineares, pois os termos nas equações de momento  $\vec{u} \cdot \nabla u$  e  $\vec{u} \cdot \nabla v$ , chamados de termos advectivos, acarretarão termos que envolvem produtos de variáveis do problema. Outra questão é a indisponibilidade de uma equação explícita para o cálculo da pressão. Enquanto o campo de velocidades pode ser computado a partir das equações de momento, o campo de pressão não pode ser computado diretamente usando a equação de continuidade, o que seria necessário para uma solução do problema (MOUKALLED; MANGANI; DARWISH, 2015). Consequentemente, uma equação para a pressão é necessária e precisa ser derivada a partir das equações disponíveis. Essa abordagem é a essência do método **SIMPLE** (*Semi Implicit Method for Pressure Correction Linked Equations*) de (PATANKAR; SPALDING, 1972), algoritmo adotado neste trabalho. A solução envolve discretizar e resolver as equações de momento para obter uma aproximação para o vetor velocidade, e reformular a equação discreta de continuidade para obter uma equação que envolva a pressão, de tal forma que surgirá um processo iterativo a ser seguido.

O algoritmo segue um tipo de procedimento iterativo do tipo Método de Picard, no qual a equação de momento é resolvida usando o campo de pressão obtido na iteração anterior. O campo resultante de velocidade satisfaz a conservação de momento, mas não

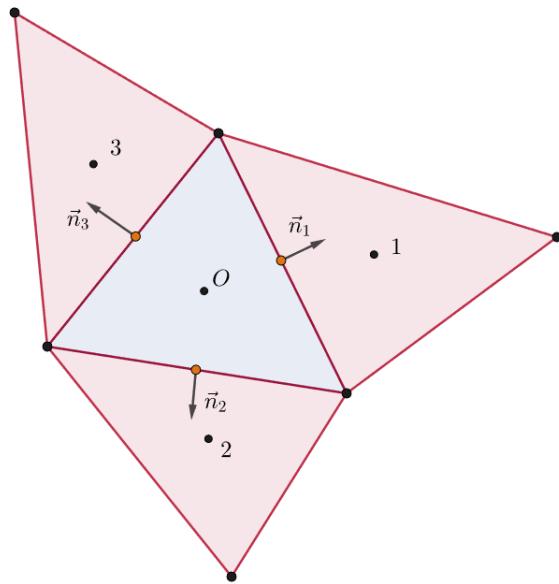


Figura 3.4: Stencil de uma malha não estruturada com elementos triangulares. Na imagem, o volume de referência está com rótulo  $O$ , enquanto os numerados de 1 até 3 são seus vizinhos. Os vetores  $\vec{n}$  são as normais de cada face da célula  $O$ .

necessariamente a conservação de massa. Esse campo de velocidade então é usado para construir uma equação relacionada à pressão, cuja sua solução possibilitará corrigir ambos os campos para satisfazer dessa vez a equação de conservação de massa (VERSTEEG; MALALASEKERA, 2007). Uma nova iteração começa então, e essa sequência é repetida até que o campo de velocidade e pressão satisfaçam a tolerância desejada das equações de conservação de massa e momento. De forma preliminar, para iniciar o algoritmo, é necessário discretizar as equações de momento e de continuidade. Essas discretizações serão feitas considerando o volume de controle não estruturado na Figura 3.4 quando necessário.

## 3.5 Discretização da Equação de Continuidade

A forma discreta da equação de continuidade é obtida integrando a Equação (2.7) sobre o volume  $O$ , resultando em:

$$\int_{V_c} (\nabla \cdot \vec{u}) dV = 0, \quad (3.8)$$

onde  $V_c$  representa o volume de controle. Agora, invoca-se o Teorema de Gauss, também conhecido como **Teorema da Divergência**. O teorema diz que, dado um volume  $V$  no espaço tridimensional, com superfície de contorno denotada por  $S$  e  $\vec{n}$  sendo o campo de vetores normais unitários apontando para fora de  $S$ , se  $\vec{v}$  for um campo vetorial definido em  $V$ , então:

$$\int_V (\nabla \cdot \vec{v}) dV = \oint_S (\vec{v} \cdot \vec{n}) dS. \quad (3.9)$$

O teorema da divergência implica que o fluxo líquido de um campo vetorial através de uma superfície fechada é igual ao volume total de todas as fontes e sorvedouros (isto é, a integral de volume de sua divergência) sobre a região dentro da superfície. Sendo assim, pode-se aplicar o mesmo na equação (3.8):

$$\oint_S (\vec{u} \cdot \vec{n}) dS = 0. \quad (3.10)$$

Uma vez que a célula em questão é delimitada por um conjunto discreto de faces planas (Figura 3.4), a integral de superfície pode ser reescrita como um somatório sobre as faces:

$$\sum_{f=1}^{N_{f,O}} \oint_S (\vec{u} \cdot \vec{n}) dS = 0, \quad (3.11)$$

onde  $f$  indica o índice que varia sobre cada face, até o número de faces total da célula  $O$  dado por  $N_{f,O}$ . Usando que a média dos valores em uma face é aproximadamente igual ao valor avaliado no centro da face multiplicado pela respectiva área da face (Equação (3.6)):

$$\sum_{f=1}^{N_{f,O}} (\vec{u}_f \cdot \vec{n}_f) A_f = 0. \quad (3.12)$$

Em que os subscritos  $f$  indicam os respectivos valores avaliados no centro de cada face do volume de controle. Uma coisa interessante e válida a ser realizada é reintroduzir a densidade  $\rho$  na equação discreta, multiplicando ambos os lados pelo valor constante  $\rho_f$ :

$$\sum_{f=1}^{N_{f,O}} \rho_f (\vec{u}_f \cdot \vec{n}_f) A_f = \sum_{f=1}^{N_{f,O}} \dot{m}_f = 0. \quad (3.13)$$

Com isso, através da discretização surge um significado físico. A notação  $\dot{m}_f$  representa o fluxo de massa no centro da face  $f$ , e indica o quanto de massa do fluido entra e sai em cada uma das faces da célula.

## 3.6 Discretização das Equações de Momento

Para a equação de momento, será aqui descrito seu processo de discretização apenas na componente  $x$ , sendo a componente  $y$  obtida no mesmo processo. Sendo assim, a primeira etapa é integrar a equação de momento em  $x$  sobre o volume de controle  $O$ :

$$\int_{V_c} (\nabla \cdot (\rho u \vec{u})) dV = - \int_{V_c} \left( \frac{\partial p}{\partial x} \right) dV + \int_{V_c} (\nabla \cdot (\mu \nabla u)) dV + \int_{V_c} S_x dV. \quad (3.14)$$

Novamente, fazendo uso do Teorema da Divergência (3.9), pode-se trocar as integrais de volume dos termos advectivo e difusivo por integrais de superfície:

$$\int_S \rho u (\vec{u} \cdot \vec{n}) dS = - \int_{V_c} \left( \frac{\partial p}{\partial x} \right) dV + \int_S \mu ((\nabla u) \cdot \vec{n}) dS + \int_{V_c} S_x dV. \quad (3.15)$$

Representando novamente as integrais de superfície como somatórios ao longo das faces do volume de controle, e tomando a média dos valores como o valor no centro da face, tem-se que:

$$\sum_{f=1}^{N_{f,O}} \rho_f u_f (\vec{u} \cdot \vec{n})_f A_f = - \int_{V_c} \left( \frac{\partial p}{\partial x} \right) dV + \sum_{f=1}^{N_{f,O}} \mu_f ((\nabla u) \cdot \vec{n})_f A_f + \int_{V_c} S_x dV. \quad (3.16)$$

Neste momento, já é possível discretizar a integral do termo fonte. Essa integral pode ser feita de duas formas (MAZUMDER, 2015): a primeira opção é realizar a integração exata. Contudo, isso só é viável caso a função seja conhecida, tendo uma relação explícita, e o volume de controle tenha formato bem definido, critérios esses que na maior parte dos casos não são satisfeitos. A segunda opção é novamente considerar o valor médio de tal forma que:

$$\int_{V_c} S_x dV = \bar{S}_x V = S_{x,O} V_O. \quad (3.17)$$

No qual a ideia principal é avaliar o termo fonte pegando a função em que os valores são fornecidos nos centroides dos volumes de controle, e multiplicar pelos respectivos volumes das células. Assim, dando continuação:

$$\sum_{f=1}^{N_{f,O}} \rho_f u_f (\vec{u} \cdot \vec{n})_f A_f = - \int_{V_c} \left( \frac{\partial p}{\partial x} \right) dV + \sum_{f=1}^{N_{f,O}} \mu_f (\nabla u \cdot \vec{n})_f A_f + S_{x,O} V_O, \quad (3.18)$$

ao avaliar a equação (3.18), surgem alguns entraves, uma vez que os valores de  $u_f$  não são conhecidos, apenas os valores nos centroides. Dessa forma, não é possível tratar diretamente o gradiente de pressão, nem tampouco o termo difusivo. Assim, nas seções subsequentes, cada uma dessas parcelas será analisada de maneira independente, visto que existem aspectos relevantes a serem discutidos individualmente para cada termo.

### 3.6.1 Discretização do termo difusivo

Relembrando a parcela de difusiva:

$$\sum_{f=1}^{N_{f,O}} \mu_f (\nabla u \cdot \vec{n})_f A_f. \quad (3.19)$$

Nessa expressão,  $A_f$  é a área da face, quantidade inferida a partir da malha. Já o valor de  $\mu_f$  ainda não está diretamente disponível, pois a viscosidade  $\mu$  é definida nos centroides das células. Para contornar esse ponto, é necessário interpolar  $\mu$  na face usando os valores disponíveis nas células vizinhas.

Em muitos problemas, são necessários tratamentos especiais para essa interpolação. Entretanto, o procedimento mais comum (VERSTEEG; MALALASEKERA, 2007) é utilizar uma interpolação ponderada pelas distâncias. Para uma propriedade genérica  $\phi$ , essa interpolação pode ser escrita como:

$$\phi_f = \frac{\phi_1/d_1 + \phi_2/d_2}{1/d_1 + 1/d_2}, \quad (3.20)$$

em que as distâncias  $d_1$  e  $d_2$  são as distâncias entre o centro das células até o meio da face. O erro associado a esse tipo de interpolação é de segunda ordem, quando  $d_1 = d_2$ , e situa-se entre primeira e segunda ordem caso as distâncias sejam diferentes.

Uma forma equivalente e útil para implementação é reescrever a Equação 3.20 como:

$$\phi_f = \left( \frac{1/d_1}{1/d_1 + 1/d_2} \right) \phi_1 + \left( 1 - \frac{1/d_1}{1/d_1 + 1/d_2} \right) \phi_2, \quad (3.21)$$

definindo o peso:

$$w_f = \frac{1/d_1}{1/d_1 + 1/d_2}, \quad (3.22)$$

obtém-se a forma compacta:

$$\phi_f = w_f \phi_1 + (1 - w_f) \phi_2. \quad (3.23)$$

A função  $w_f$  pode ser pré-computada uma vez que depende somente de informações geométricas locais. Assim, para determinar o valor de  $\mu_f$ , basta-se aplicar a Equação (3.23). O principal desafio restante está no tratamento do termo  $(\nabla u \cdot \vec{n})_f$ , isto é, na avaliação do gradiente de  $u$  projetado na direção normal à face.

### Sobre-relaxação do gradiente difusivo

No caso do termo  $(\nabla u \cdot \vec{n})$ , há um grande problema, visto que as malhas não estruturadas geralmente são não ortogonais, ou seja, a direção que une os centroides não é colinear a direção normal. Assim, enquanto em malhas estruturadas o gradiente poderia ser encontrado de forma bem simples e direta; no caso de malhas não estruturadas é necessário realizar manipulações com o intuito de tratar essa diferença. A primeira coisa que pode ser feita então é dividir o fluxo normal em duas componentes:

$$(\nabla u)_f \cdot \vec{n}_f = (\nabla u)_f \cdot \vec{E}_f + (\nabla u)_f \cdot \vec{T}, \quad (3.24)$$

onde o primeiro termo representa uma contribuição que pode ser obtida via aproximação com série de Taylor devido à direção ser igual a que une os centros das células, enquanto o segundo termo é chamado de **difusão cruzada** ou difusão não ortogonal, e aparece

devido a não ortogonalidade da malha a ser utilizada. Diferentes opções para decompor a normal existem na literatura, sendo que a escolha delas afeta convergência e estabilidade. A abordagem escolhida nesse trabalho é chamada de sobre-relaxada (*over-relaxed*), e a figura 3.5 ilustra a sua decomposição.

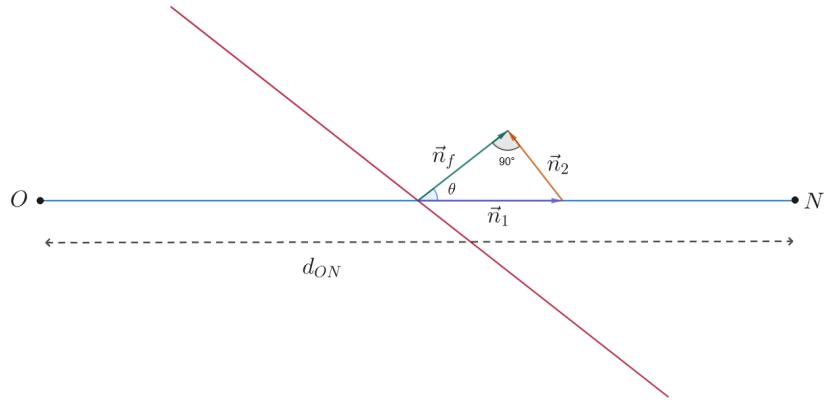


Figura 3.5: Escolha de vetores  $\vec{n}_1$  e  $\vec{n}_2$  para decompor o gradiente difusivo. Na imagem também está retratado o vetor normal a face  $\vec{n}_f$ .  $d_{ON}$  representa a distância entre a célula  $O$  e seu vizinho  $N$ .

Na Figura 3.5, dividiu-se a normal em dois vetores, de forma que um deles é ortogonal a normal da face. Assim, a importância do termo envolvendo  $u_O$  e  $u_N$  é forçada a aumentar conforme a não ortogonalidade da malha também aumentar, de tal forma que isso evite problemas de *overflow*. Matematicamente, o cômputo de  $\vec{n}_1$  é dado primeiro calculando o cosseno do ângulo  $\theta$ :

$$\cos \theta = \frac{|\vec{n}_f|}{|\vec{n}_1|} \Leftrightarrow |\vec{n}_1| = \frac{|\vec{n}_f|}{\cos \theta}, \quad (3.25)$$

a partir da definição de produto escalar, pode-se trocar  $\cos \theta$  por:

$$|\vec{n}_1| = |\vec{n}_f| \left( \frac{|\vec{n}_f| |\vec{d}_{ON}|}{\vec{n}_f \cdot \vec{d}_{ON}} \right), \quad (3.26)$$

mas isso corresponde ao valor da magnitude de  $\vec{n}_1$ , enquanto o interesse é no vetor. Desse modo, basta multiplicar a magnitude pela direção do vetor  $\vec{d}_{ON}$  (vetor que liga os centroides):

$$\vec{n}_1 = |\vec{n}_1| \left( \frac{\vec{d}_{ON}}{|\vec{d}_{ON}|} \right), \quad (3.27)$$

que simplificando resulta em:

$$\vec{n}_1 = \vec{d}_{ON} \left( \frac{|\vec{n}_f|^2}{\vec{n}_f \cdot \vec{d}_{ON}} \right). \quad (3.28)$$

Para achar o vetor  $\vec{n}_2$ , basta usar o vetor normal e o vetor  $\vec{n}_1$ :  $\vec{n}_2 = \vec{n}_1 - \vec{n}_f$ . Assim, conforme  $\theta$  aumenta, a magnitude de  $\vec{n}_1$  e  $\vec{n}_2$  aumentam conjuntamente, evitando problemas numéricos. Conforme discutido e trabalhado por (JASAK, 1996), o mesmo conseguiu executar diversos experimentos e constatar a efetividade e estabilidade da abordagem sobre-relaxada, fundamentando a utilização neste trabalho pela mesma. Para calcular o gradiente na direção que está alinhada aos centroides, pode-se performar uma expansão em série de Taylor para  $O$  e  $N$  sobre  $f$ :

$$\begin{aligned} u_N &= u_f + \frac{\partial u}{\partial x} \Big|_f (x_N - x_f) + \frac{\partial u}{\partial y} \Big|_f (y_N - y_f) \\ &\quad + \frac{1}{2} \frac{\partial^2 u}{\partial x^2} \Big|_f (x_N - x_f)^2 + \frac{1}{2} \frac{\partial^2 u}{\partial y^2} \Big|_f (y_N - y_f)^2 + \frac{1}{2} \frac{\partial^2 u}{\partial x \partial y} \Big|_f (x_N - x_f)(y_N - y_f) + \dots \end{aligned} \quad (3.29)$$

$$\begin{aligned} u_O &= u_f + \frac{\partial u}{\partial x} \Big|_f (x_O - x_f) + \frac{\partial u}{\partial y} \Big|_f (y_O - y_f) \\ &\quad + \frac{1}{2} \frac{\partial^2 u}{\partial x^2} \Big|_f (x_O - x_f)^2 + \frac{1}{2} \frac{\partial^2 u}{\partial y^2} \Big|_f (y_O - y_f)^2 + \frac{1}{2} \frac{\partial^2 u}{\partial x \partial y} \Big|_f (x_O - x_f)(y_O - y_f) + \dots \end{aligned} \quad (3.30)$$

subtraindo a equação (3.29) da equação (3.30), obtém-se:

$$\begin{aligned} u_N - u_O &= \frac{\partial u}{\partial x} \Big|_f (x_N - x_O) + \frac{\partial u}{\partial y} \Big|_f (y_N - y_O) \\ &\quad + \frac{1}{2} \frac{\partial^2 u}{\partial x^2} \Big|_f [(x_N - x_f)^2 - (x_O - x_f)^2] + \frac{1}{2} \frac{\partial^2 u}{\partial y^2} \Big|_f [(y_N - y_f)^2 - (y_O - y_f)^2] \\ &\quad + \frac{1}{2} \frac{\partial^2 u}{\partial x \partial y} \Big|_f [(x_N - x_f)(y_N - y_f) - (x_O - x_f)(y_O - y_f)] + \dots \end{aligned} \quad (3.31)$$

truncando a série no segundo termo, e sabendo que  $\vec{n}_1 = \left( \frac{x_N - x_O}{d_{ON}}, \frac{y_N - y_O}{d_{ON}} \right)$ :

$$(\nabla u)_f \cdot \vec{n}_1 \approx \frac{\phi_N - \phi_O}{d_{ON}}. \quad (3.32)$$

Sendo que essa aproximação possui erro de truncamento da ordem de  $\mathcal{O}(h^2)$ . Já

para o caso da difusão cruzada, não se consegue ainda expressar o gradiente em termos dos valores das células diretamente. O termo do gradiente será tratado na forma de **correção deferida**, computando o seu valor usando o campo de valores da iteração atual e adicionando-o como um termo fonte na equação algébrica. O gradiente é computado usando os próprios centroides na malha, e os valores nas interfaces podem então ser interpolados.

### Reconstrução do gradiente usando mínimos quadrados

Para realizar o cômputo da difusão cruzada, é necessário então estimar o valor do gradiente, a fim de incorporá-lo no sistema algébrico das equações discretas. O método utilizado nesse trabalho é chamado de Mínimos Quadrados (*least-squares*), visto que ele é mais flexível quanto a questão da ordem de acurácia e o *stencil* utilizado (MOUKALLED; MANGANI; DARWISH, 2015). Considere o volume de controle da Figura 3.6.

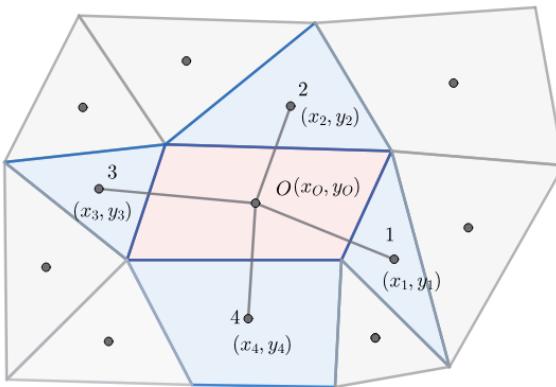


Figura 3.6: Volume de controle  $O$ , e seus vizinhos rotulados de 1 até 4.

Baseado em qualquer um dos vizinhos  $N$  de  $O$ , é plausível aproximar o valor de uma propriedade  $\phi$  em  $N$  usando uma aproximação por série de Taylor:

$$\phi_N = \phi_O + \left. \frac{\partial \phi}{\partial x} \right|_O \Delta x_{ON} + \left. \frac{\partial \phi}{\partial y} \right|_O \Delta y_{ON}, \quad (3.33)$$

em que  $\Delta x_{ON}$  e  $\Delta y_{ON}$  representam a variação da distância dos centroides de  $O$  e  $N$  nas direções  $x$  e  $y$ . É possível realizar tal aproximação para cada um dos quatro vizinhos,

formando o conjunto de equações:

$$\left\{ \begin{array}{l} \phi_1 - \phi_O = \frac{\partial \phi}{\partial x} \Big|_O \Delta x_1 + \frac{\partial \phi}{\partial y} \Big|_O \Delta y_1 \\ \phi_2 - \phi_O = \frac{\partial \phi}{\partial x} \Big|_O \Delta x_2 + \frac{\partial \phi}{\partial y} \Big|_O \Delta y_2 \\ \phi_3 - \phi_O = \frac{\partial \phi}{\partial x} \Big|_O \Delta x_3 + \frac{\partial \phi}{\partial y} \Big|_O \Delta y_3 \\ \phi_4 - \phi_O = \frac{\partial \phi}{\partial x} \Big|_O \Delta x_4 + \frac{\partial \phi}{\partial y} \Big|_O \Delta y_4 \end{array} \right.$$

que pode ser representado matricialmente como:

$$\begin{bmatrix} \Delta x_1 & \Delta y_1 \\ \Delta x_2 & \Delta y_2 \\ \Delta x_3 & \Delta y_3 \\ \Delta x_4 & \Delta y_4 \end{bmatrix} \begin{bmatrix} \frac{\partial \phi}{\partial x} \Big|_O \\ \frac{\partial \phi}{\partial y} \Big|_O \end{bmatrix} = \begin{bmatrix} \phi_1 - \phi_O \\ \phi_2 - \phi_O \\ \phi_3 - \phi_O \\ \phi_4 - \phi_O \end{bmatrix}$$

Isso resulta em um sistema sobredeterminado de equações lineares, o qual pode ser escrito, de forma compacta, como  $\mathbf{A}(\nabla\phi)_O \approx \mathbf{b}$ , em que a matriz  $\mathbf{A}$  contém as informações geométricas (vetores que ligam o ponto  $O$  aos seus vizinhos) e o vetor  $\mathbf{b}$  reúne as diferenças de propriedade  $\Delta\phi$  entre  $O$  e seus vizinhos.

Como o sistema é sobredeterminado, sua solução é obtida no sentido de mínimos quadrados, resolvendo-se as chamadas *equações normais*, cuja solução formal pode ser escrita como:

$$(\nabla\phi)_O \approx (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}, \quad (3.34)$$

assumindo que  $\mathbf{A}^T \mathbf{A}$  é inversível.

Uma vez conhecido o gradiente no centro da célula  $O$ , os gradientes nas faces podem ser obtidos por interpolação usando a equação (3.23). Unindo a ideia de decomposição do vetor normal à face com a reconstrução via mínimos quadrados, pode-se obter então:

$$\sum_{f=1}^{N_{f,O}} \mu_f (\nabla u \cdot \vec{n})_f A_f = \sum_{f=1}^{N_{f,O}} \mu_f \frac{u_N - u_O}{d_{ON}} A_f + \sum_{f=1}^{N_{f,O}} \mu_f (\nabla u^* \cdot \vec{n}_2) A_f, \quad (3.35)$$

em que  $\vec{n}_2$  foi calculado a partir da equação (3.28), e  $\nabla u^*$  representa o gradiente calculado com base nos valores de  $u$  de uma dada iteração. Esse procedimento será dado de forma explícita, passando essa componente para o vetor  $\mathbf{b}$  do sistema de equações. Uma última simplicidade é expressar o produto  $\frac{\mu_f A_f}{d_{ON}}$  com a notação  $D_f$ , tendo o significado físico de representar um fluxo difusivo na célula.

### Exemplo numérico: Equação de Poisson

Durante as etapas de desenvolvimento, conseguiu-se testar a eficácia dessa abordagem de tratamento do gradiente e escolheu-se exemplificar também sua utilização em um problema de Poisson, com solução manufaturada  $u(x, y) = 100x(1 - x)y(1 - y)$ . Os dados do problema são:

$$\begin{cases} \nabla^2 u = -200 x(1 - x) - 200 y(1 - y), & \text{em } \Gamma, \\ u(x, y) = 0, & \text{em } \partial\Gamma, \end{cases} \quad (3.36)$$

em que  $\Gamma = [0, 1] \times [0, 1]$  representa um domínio quadrado de lado unitário.

A malha utilizada, bem como as malhas empregadas nos testes posteriores, foi gerada com o **Gmsh**, ferramenta de geração de malhas que será descrita com mais detalhes no próximo capítulo. Neste caso, utilizou-se a opção de geração de malhas *transfinite*, resultando em uma malha estruturada na qual os volumes de controle são triângulos retângulos, obtidos pela divisão de cada quadrilátero ao longo de uma de suas diagonais, como ilustrado na Figura 3.7(a).

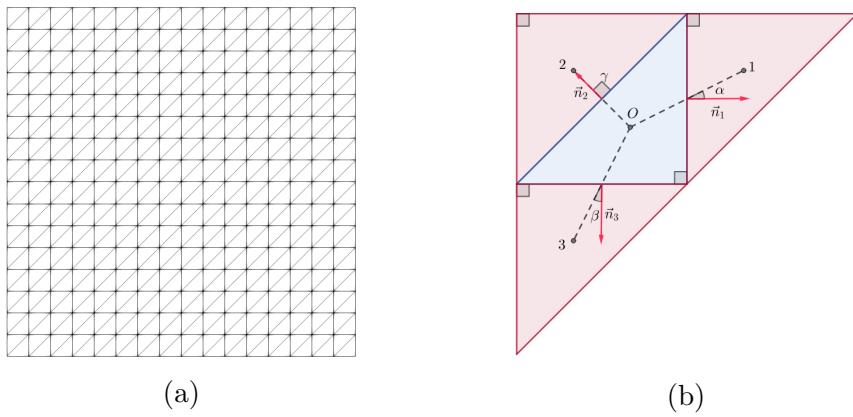


Figura 3.7: Exemplo de uma malha em (a) e exemplo de um volume de controle disponível nessa malha em (b), retratando seus vizinhos e ângulos formados.

Na Figura 3.7(b) estão destacados os centros dos volumes de controle, e as linhas tracejadas indicam a direção que liga os seus centroides. Ao traçar as normais às faces, destacadas em vermelho, observa-se que, para o triângulo cujo centro é o ponto 2, a normal e a linha que une os centros são colineares. O mesmo não ocorre nos outros dois triângulos, para os quais surgem ângulos distintos, aqui denotados por  $\alpha$  e  $\beta$ . Com base em relações de trigonometria e geometria plana, obtém-se aproximadamente  $\alpha \approx 26,3^\circ$  e  $\beta \approx 25,9^\circ$ .

Visando avaliar o comportamento do erro sob refinamento de malha, foram considerados os tamanhos  $N = 32, 128, 512, 2048$  e  $8192$  volumes de controle. O erro entre a solução numérica e a solução manufaturada é medido pela norma  $L_2$  (SILVA, 2023):

$$\|u - \bar{u}\|_{L_2} = \left( \int_{\Gamma} (u - \bar{u})^2 d\Gamma \right)^{\frac{1}{2}}, \quad (3.37)$$

onde  $\bar{u}$  representa a solução exata. Também, para ter um comparativo, variou-se o número de correções explícitas de não ortogonalidade (denotado a quantidade por  $\theta$ ), tomando como valores  $\theta = 5, 10, 15$ ; para verificar o impacto na solução, comparando-a com uma solução em que não foram utilizadas correções ( $\theta = 0$ ). O parâmetro  $h$ , que indica o refinamento de malha, é definido como (VASCONCELLOS, 1999):

$$h = \sqrt{\frac{\sum_{i=1}^N A_i}{N}}. \quad (3.38)$$

Onde  $N$  é o número de volumes de controle e  $A_i$  é o volume da célula  $i$ . Como o domínio é um quadrado de lados unitários, pode-se simplificar a conta acima para apenas  $h = \frac{1}{\sqrt{N}}$ .

Na Figura 3.8, estão exibidos os gráficos das soluções, comparando qualitativamente os resultados obtidos. Como esperado, na Figura (b) conseguiu-se capturar o formato da solução, tendo a característica semelhante a um pulso circular. Enquanto isso, na Figura (a) é notório que em virtude de ignorar a difusão cruzada, o pulso fica achatado em uma dada direção, sendo mais similar a uma elipse. Além disso, na Figura 3.9 estão exibidos os padrões de convergência para cada caso. Quando é negligenciado totalmente a difusão cruzada, o erro praticamente não varia, tendo um comportamento bem constante ao longo dos refinamentos. Em contraste, é possível notar a melhoria na

ordem de convergência já para 5 correções, em que a ordem sai de 0.07 para 1.74, bem mais próximo da convergência esperada de 2. Para o caso de  $\theta = 10$  e  $\theta = 15$  não há tanta diferença, indicando que provavelmente para 10 correções o processo já convergiu. O motivo de adotar um critério para número de correções ao invés de tolerância é estar de acordo com os *softwares* de CFD, que normalmente disponibilizam o controle também com número de correções.

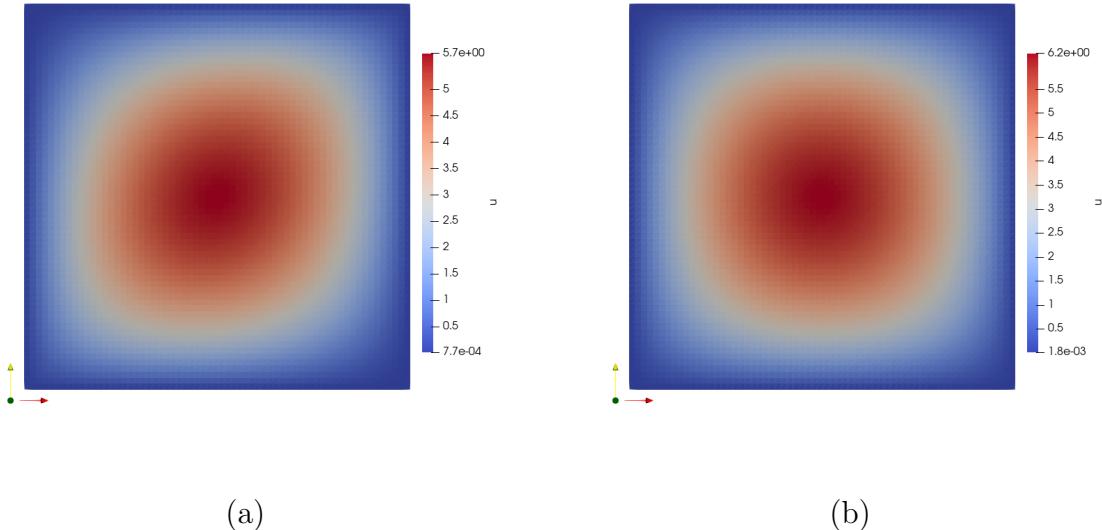


Figura 3.8: Gráficos da solução da equação 3.36. A figura (a) representa a solução com  $\theta = 0$ , enquanto a solução da figura (b) representa a solução com  $\theta = 15$ . Ambas as figuras foram obtidas com a malha de 8192 volumes de controle.

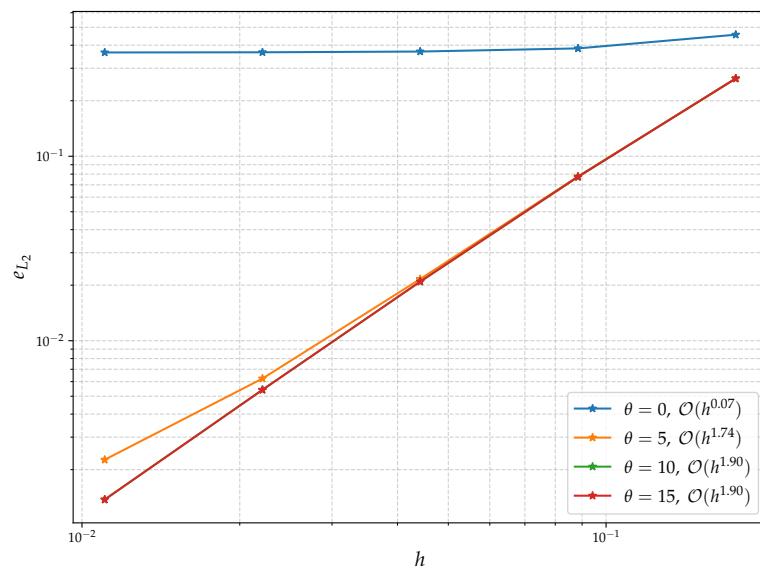


Figura 3.9: Comparaçāo da convergência variando o número de correções.

### 3.6.2 Discretização do termo advectivo

Dando prosseguimento, após o tratamento do termo difusivo, passa-se à análise do termo advectivo. Segundo a discretização apresentada na equação (3.18), o termo de advecção pode ser escrito na forma:

$$\sum_{f=1}^{N_{f,O}} \rho_f u_f (\vec{u} \cdot \vec{n})_f A_f, \quad (3.39)$$

onde nessa expressão, a densidade nas faces,  $\rho_f$ , pode novamente ser obtida por interpolação linear, enquanto a área de cada face,  $A_f$ , é calculada a partir da geometria da malha. De forma análoga ao tratamento da equação de continuidade discreta, é conveniente agrupar alguns termos e reescrever o somatório da seguinte maneira:

$$\sum_{f=1}^{N_{f,O}} \dot{m}_f u_f, \quad (3.40)$$

em que  $\dot{m}_f = \rho_f (\vec{u} \cdot \vec{n})_f A_f$  representa o conhecido fluxo de massa através da face  $f$ . Resta, portanto, determinar o valor de  $u_f$ , isto é, o valor da componente de velocidade  $u$  nas faces do volume de controle associado ao ponto  $O$ . No caso do fluxo de massa  $\dot{m}_f$ , seu tratamento detalhado será apresentado na próxima subseção.

Anteriormente, a interpolação linear foi utilizada para obter valores nas faces a partir de médias ponderadas pela distância aos centros de célula. A princípio, seria natural empregar a mesma estratégia para o cálculo de  $u_f$ , assumindo, por exemplo,  $u_f = w_f u_O + (1 - w_f) u_N$ , em que  $u_O$  e  $u_N$  são os valores de  $u$  nos centros das células adjacentes à face  $f$ . No entanto, conforme discutido em (VERSTEEG; MALALASEKERA, 2007), esse esquema apresenta limitações importantes.

Do ponto de vista teórico, a solução obtida pelos métodos numéricos pode tornar-se indistinguível da solução exata, independentemente do esquema de discretização adotado, desde que a malha seja suficientemente refinada (VERSTEEG; MALALASEKERA, 2007). Na prática, porém, a capacidade computacional disponível impõe o uso de um número finito e restrito de células. Em consequência, a solução numérica pode deixar de satisfazer algumas propriedades físicas, entre as quais se destacam: **conservatividade**, **limitabilidade** e **transpositividade**.

A conservatividade exige que, para garantir a conservação de  $\phi$  em todo o domínio, o fluxo que deixa um volume de controle seja exatamente igual ao fluxo que entra no volume adjacente através da mesma face. A limitabilidade requer que o esquema satisfaça o critério de dominância diagonal, assegurando compatibilidade com os métodos numéricos empregados na solução dos sistemas lineares. Esse critério é fundamental ao garantir que, na ausência de termos fonte, o valor de  $\phi$  permaneça limitado pelos valores de contorno, além de impor que todos os coeficientes da equação discreta possuam o mesmo sinal, preferencialmente positivo. Por fim, a transpositividade está relacionada à capacidade do esquema numérico de representar corretamente o transporte de uma grandeza no sentido do escoamento, respeitando a física do processo de advecção.

Baseado nesses critérios, embora a interpolação linear seja de segunda ordem, também se verifica que esse esquema satisfaz apenas o requisito de conservatividade, faltando nos outros critérios a depender do número de Péclet do escoamento (VERSTEEG; MALALASEKERA, 2007)). Para contornar essas dificuldades, adotou-se neste trabalho esquemas do tipo *upwind*, que incorporam explicitamente a direção do escoamento na definição dos valores nas faces.

## Métodos Upwind

Um dos principais problemas do esquema de interpolação linear é o fato de não considerar a direção do escoamento. Em um fluxo fortemente convectivo da esquerda para a direita, por exemplo, o tratamento torna-se inadequado, pois os valores de  $\phi$  localizados a montante (à esquerda) deveriam exercer maior influência sobre o valor final.

Os esquemas do tipo *upwind* incorporam explicitamente essa característica física, definindo o valor na face em função da direção do fluxo de massa. Se o fluxo de massa (ou a direção do vento) é da célula  $O$  para a célula  $N$ , o valor na face é majoritariamente influenciado por  $O$  e, por consequência, atribui-se à face o valor de  $\phi_O$ . No caso contrário, isto é, quando o fluxo se dá de  $N$  em direção a  $O$ , o valor de  $\phi_N$  passa a ser o dominante na face.

Mais formalmente, assumindo que o vetor normal  $\vec{n}_f$  aponta da célula  $O$  para a célula  $N$ , o esquema *upwind differencing* (UD), de primeira ordem, pode ser escrito como:

$$\phi_f = \begin{cases} \phi_O, & \text{se } \dot{m}_f > 0; \\ \phi_N, & \text{se } \dot{m}_f < 0. \end{cases} \quad (3.41)$$

Para fins de implementação computacional, é conveniente recorrer à função  $\max(v_0, v_1)$ , que retorna o maior valor entre  $v_0$  e  $v_1$ . Dessa forma, pode-se escrever:  $\phi_f = \phi_O \max(\dot{m}_f, 0) - \phi_N \max(-\dot{m}_f, 0)$ .

Do ponto de vista de suas características numéricas, o esquema *upwind* é consistente no cálculo dos fluxos, implicando sua conservatividade. Além disso, trata-se de um esquema limitado, com todos os coeficientes positivos, tornando a matriz resultante diagonalmente dominante. Outra característica importante é que o método incorpora explicitamente a direção do escoamento em sua formulação, o que o torna mais realista do ponto de vista físico.

Por outro lado, a principal limitação do esquema *upwind* é sua baixa acurácia. Isso tende a introduzir difusão numérica (também chamada de falsa difusão), produzindo um amortecimento artificial da solução em regiões com gradientes acentuados (MAZUMDER, 2015; ANDERSON, 1995). Ainda assim, devido à sua simplicidade e robustez, o esquema *upwind* é amplamente utilizado em CFD, oferecendo forte estabilidade numérica. Em simulações que requerem maior precisão, como em LES (*Large Eddy Simulation*), ou em problemas com malhas de baixa qualidade e altos números de Reynolds, é comum empregar o *upwind* para obter uma solução inicial estável, que depois pode ser refinada com esquemas de ordem mais elevada.

Para mitigar o problema da falsa difusão e aumentar a acurácia do método, uma estratégia natural consiste em aumentar a ordem do esquema *upwind* pela inclusão de mais termos na expansão em série de Taylor (WARMING; BEAM, 1976), resultando no método *linear upwind differencing* (LUD). Em malhas não estruturadas, pode-se expandir a série em torno do ponto  $O$ , obtendo

$$\phi(x, y) = \phi_O + (\nabla \phi)_O \cdot \Delta \vec{r} + \mathcal{O}(|\Delta \vec{r}|^2), \quad (3.42)$$

onde  $\Delta \vec{r}$  é o vetor que une  $O$  ao centro da face. Assim, o valor da propriedade  $\phi$  pode ser

avaliado apenas com:

$$\phi_f = \begin{cases} \phi_P + (\nabla\phi)_P \cdot \Delta\vec{r}, & \text{se } \dot{m}_f > 0; \\ \phi_N + (\nabla\phi)_N \cdot \Delta\vec{r}, & \text{se } \dot{m}_f < 0. \end{cases} \quad (3.43)$$

E a adição de um termo que envolve um gradiente traz maior precisão para a solução do método, deixando uma variação linear entre o valor de  $\phi$  nos centroides e centro das faces. Apesar do ganho em acurácia, isso acarreta na perda de estabilidade. De maneira análoga ao caso da difusão cruzada, o gradiente também pode ser avaliado por meio da correção deferida conjuntamente aos mínimos quadrados.

### Exemplo Numérico: Convecção Pura

Para explicitar a diferença e agregar maior profundidade ao tratamento da convecção nesse trabalho, testou-se também o impacto dos esquemas *upwind* e *linear upwind*. O primeiro problema serviu apenas para validar a questão das ordens de convergência, e a solução exata é dada por:  $\phi(x, y) = \sin(\pi x) \sin(\pi y)$ . As condições do problema são dadas então:

$$\begin{cases} \vec{u} \cdot \nabla\phi = \pi \cos(\pi x) \sin(\pi y) + \pi \sin(\pi x) \cos(\pi y) & \text{em } \Gamma \\ \phi(x, y) = 0 & \text{em } \partial\Gamma. \end{cases} \quad (3.44)$$

Em que o valor da velocidade  $\vec{u}$  é prescrito no domínio  $\Gamma = [0, 1] \times [0, 1]$  e vale  $u = v = 1$ . Para obter a ordem de convergência dos métodos, realizou-se um refinamento de malha. A malha considerada dessa vez é uma malha cartesiana e estruturada composta por quadrados, análoga às malhas estruturadas tradicionais, para facilitar o cálculo da ordem de convergência. Os valores de tamanho de malha foram  $N = 16, 64, 256, 1024, 4096, 16384$ . Em relação às correções e aos cálculos do gradiente que o *linear upwind* exige, foram utilizadas apenas 10 correções. Os resultados estão disponíveis na Figura 3.10. Nela, é possível notar que o método UD apresenta uma ordem de convergência próxima à linear, enquanto o LUD está próxima à segunda ordem, estando ambos conforme o esperado.

Para avaliar características mais qualitativas, escolheu-se outro problema puramente convectivo, sendo que dessa vez o mesmo é bem conhecido (VERSTEEG; MALASEKERA, 2007). A formulação do problema é:

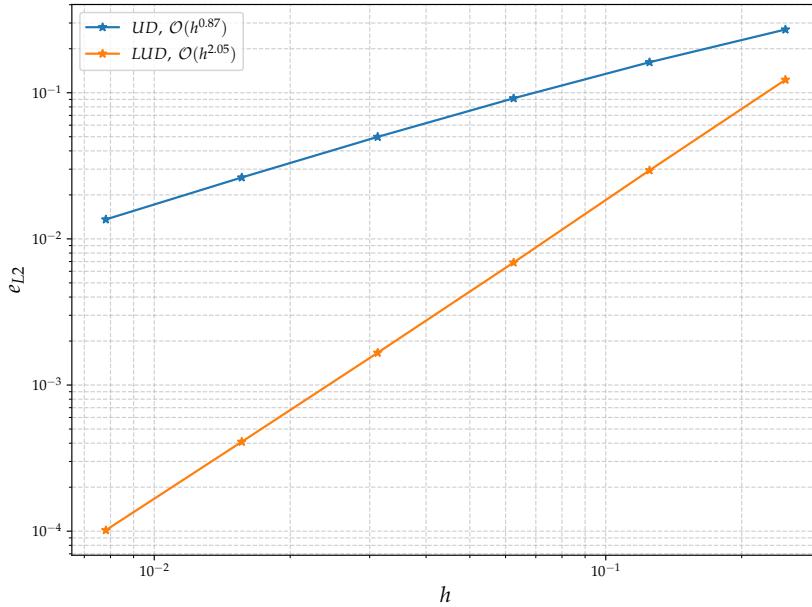


Figura 3.10: Comparaçao de convergencia entre UD e LUD.

$$\begin{cases} \vec{u} \cdot \nabla \phi = 0, \text{ em } \Gamma, \\ \phi(x=0, y) = \phi(x, y=1) = 1, \\ \phi(x=1, y) = \phi(x, y=0) = 0. \end{cases} \quad (3.45)$$

No qual novamente o campo de velocidade é  $u = v = 1$ . Para facilitar a visualizaçao, a Figura 3.11 apresenta graficamente o problema. A soluçao correta desse problema é bem documentada, e conforme a velocidade do fluido é na direção diagonal, todos os valores acima da diagonal serão  $\phi(x, y) = 1$ , enquanto os valores abaixo da diagonal serão  $\phi(x, y) = 0$ . Para validar a qualidade dos metodos, pode-se avaliar o corte da soluçao obtida na linha vermelha. Caso a soluçao seja exata, a mesma deverá ter o formato de um degrau.

Considerando o recorte apresentado na Figura 3.11, obteve-se o grafico apresentado na Figura 3.12. Nessa figura, apresentam-se a soluçao exata, caracterizada pelo degrau, bem como as soluções obtidas pelos esquemas UD e LUD. Observa-se que o esquema UD introduz um amortecimento significativo na soluçao, efeito diretamente associado à difusão numérica inerente a esse metodo (ANDERSON, 1995). Em contraste, o esquema LUD acompanha melhor a tendência do degrau, porm apresenta leves oscilações, evidenciadas por ondulações nas proximidades da descontinuidade.

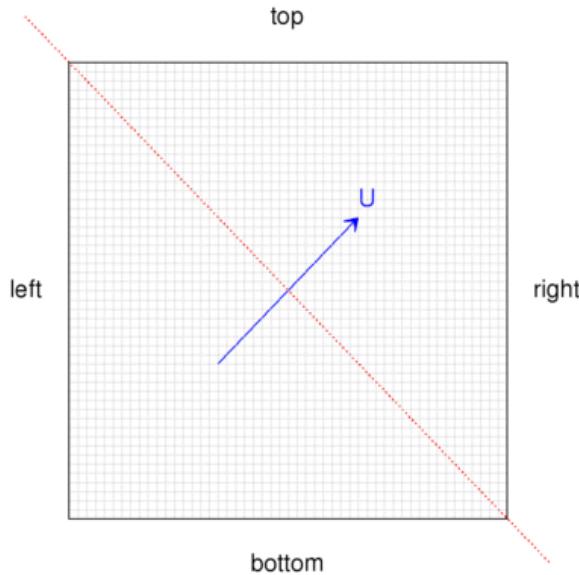


Figura 3.11: Figura exibindo o domínio, retratando também a velocidade  $\vec{u} = (1, 1)$ . O segmento em vermelho é utilizado para comparar as soluções. Imagem retirada de: <https://www.openfoam.com/documentation/guides/latest/doc/guide-schemes-divergence-example.html>.

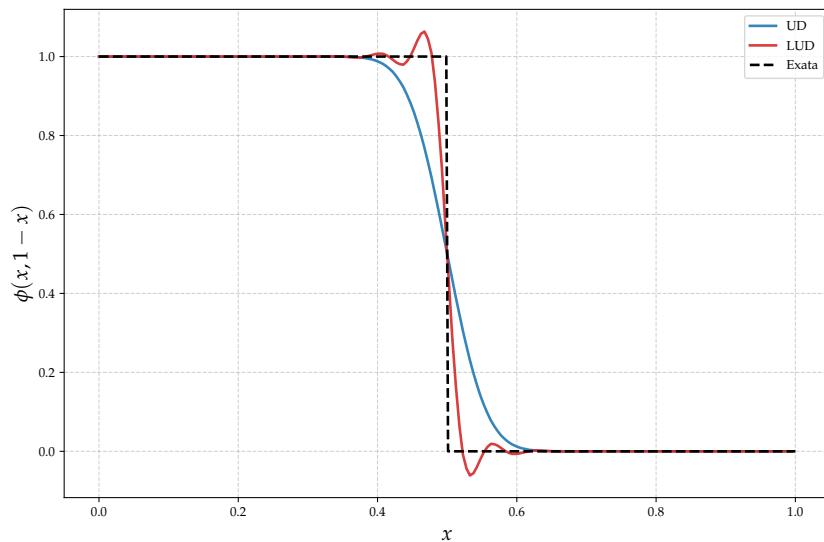


Figura 3.12: Comparaçāo entre UD e LUD na diagonal, em que a função é  $\phi(x, 1 - x)$ . Imagem obtida com refinamento de 16384 volumes de controle em uma malha estruturada composta por quadriláteros.

### 3.6.3 Discretização da Pressão

Na Equação (3.18), surge a necessidade de discretizar o termo que envolve a pressão. Para isso, considera-se o campo vetorial  $\vec{q}$  definido na direção  $x$  por  $\vec{q} = p \hat{i}$  (análogo para  $y$ ), em que  $p$  é o campo escalar de pressão. A divergência desse campo é dada por

$$\nabla \cdot \vec{q} = \left( \frac{\partial}{\partial x} \hat{i} + \frac{\partial}{\partial y} \hat{j} \right) \cdot (p\hat{i}) = \frac{\partial p}{\partial x}. \quad (3.46)$$

Dessa forma, integrando no volume de controle, tem-se:

$$\int_{V_c} \frac{\partial p}{\partial x} dV = \int_{V_c} \nabla \cdot \vec{q} dV. \quad (3.47)$$

Pelo Teorema da Divergência (Equação 3.9), obtém-se:

$$\int_{V_c} \frac{\partial p}{\partial x} dV = \int_{V_c} \nabla \cdot \vec{q} dV = \oint_{\partial V_c} (\vec{q} \cdot \vec{n}) dS. \quad (3.48)$$

Por analogia ao procedimento adotado para os demais termos, a integral de superfície pode ser aproximada por um somatório sobre as faces do volume de controle associado ao ponto  $O$ , de modo que

$$\int_{V_c} \frac{\partial p}{\partial x} dV \approx \sum_{f=1}^{N_{f,O}} (\vec{q} \cdot \vec{n})_f A_f = \sum_{f=1}^{N_{f,O}} ((p\hat{i}) \cdot \vec{n}_f) A_f = \sum_{f=1}^{N_{f,O}} p_f n_{x,f} A_f, \quad (3.49)$$

em que  $n_{x,f} = \vec{n}_f \cdot \hat{i}$  representa a componente  $x$  do vetor normal à face  $f$ .

Assim, a integral do gradiente de pressão na direção  $x$  é escrita como um somatório dos valores de pressão nas faces, projetados na direção normal correspondente. Nesse caso, a única quantidade ainda desconhecida é  $p_f$ , que pode ser obtida, por exemplo, por interpolação linear entre os valores de pressão nos centros das células adjacentes à face.

### 3.6.4 Forma final da equação de momento

Nas seções anteriores deste capítulo discutiram-se os tratamentos numéricos associados a cada termo da equação de momento. Nesta subseção, reúnem-se esses resultados com o objetivo de apresentar a forma final da equação de momento discretizada.

Considerando-se as discretizações do termo convectivo usando o esquema *linear upwind*, do termo difusivo usando difusão cruzada, além dos termos de pressão e de fonte,

a equação discretizada em  $x$  é sintetizada como:

$$\begin{aligned} \sum_{f=1}^{N_{f,O}} [(u_O + (\nabla u)_O \cdot \Delta \vec{r}) \max(0, \dot{m}_f) - (u_N + (\nabla u)_N \cdot \Delta \vec{r}) \max(0, -\dot{m}_f)] = \\ - \sum_{f=1}^{N_{f,O}} p_f n_{x,f} A_f + \sum_{f=1}^{N_{f,O}} \left[ D_f (u_N - u_O) + \sum_{f=1}^{N_{f,O}} \mu_f A_f (\nabla u \cdot \vec{n}_2) \right] + S_{x,O} V_O. \end{aligned} \quad (3.50)$$

Pode-se ainda reunir todos os termos que dependem de  $u_O$  ou  $u_N$  no lado esquerdo, deixando no lado direito apenas os termos que envolvem gradientes (tratados de forma explícita) e contribuições conhecidas. Assim, obtém-se:

$$\begin{aligned} \sum_{f=1}^{N_{f,O}} (u_O \max(0, \dot{m}_f) - u_N \max(0, -\dot{m}_f)) - \sum_{f=1}^{N_{f,O}} D_f (u_N - u_O) = - \sum_{f=1}^{N_{f,O}} p_f n_{x,f} A_f \\ + \sum_{f=1}^{N_{f,O}} ((\nabla u)_O^* \cdot \Delta \vec{r}) \max(0, \dot{m}_f) - ((\nabla u)_N^* \cdot \Delta \vec{r}) \max(0, -\dot{m}_f) + \sum_{f=1}^{N_{f,O}} \mu_f A_f (\nabla u^* \cdot \vec{n}_2)_f + S_{x,O} V_O, \end{aligned} \quad (3.51)$$

em que a notação  $(\nabla u)^*$  indica que os gradientes são avaliados explicitamente, por meio de valores provenientes de iterações anteriores e reconstruídos via método de mínimos quadrados. Para simplificar a escrita, é conveniente agrupar os termos associados à célula central  $O$  e às células vizinhas  $N$ , tratando todo o lado direito como um termo fonte efetivo  $Q_x$ :

$$\sum_{f=1}^{N_{f,O}} u_O [\max(0, \dot{m}_f) + D_f] - \sum_{f=1}^{N_{f,O}} u_N [\max(0, -\dot{m}_f) + D_f] = Q_x. \quad (3.52)$$

tomando agora as seguintes notações,

$$\begin{aligned} a_O &= \sum_{f=1}^{N_{f,O}} [\max(0, \dot{m}_f) + D_f], \\ a_N &= [\max(0, -\dot{m}_f) + D_f]. \end{aligned} \quad (3.53)$$

A equação de balanço para a componente  $u$  pode ser escrita como

$$a_O u_O + \sum_{f=1}^{N_{f,O}} a_N u_N = Q_x. \quad (3.54)$$

a qual representa a forma simplificada da equação de momento na direção  $x$ , já na forma

algébrica típica dos métodos de volumes finitos. A formulação análoga pode ser obtida para a componente  $v$  da velocidade.

### 3.7 Tratamento do acoplamento pressão-velocidade

Nesta seção, será considerado o procedimento para lidar com o acoplamento entre pressão e velocidade, ponto relevante do SIMPLE. A primeira característica a ser analisada é o armazenamento das variáveis  $\vec{u}$  e  $p$ . A forma mais natural seria armazená-los no centroide das células. Contudo, se o campo de pressão for altamente não uniforme, o mesmo poderia ser tratado como um campo uniforme na equação discretizada. Considere o exemplo da Figura 3.13, onde, no problema, o campo de pressão obtido é altamente irregular, tendo a configuração de um tabuleiro de xadrez (*checkerboarding*).

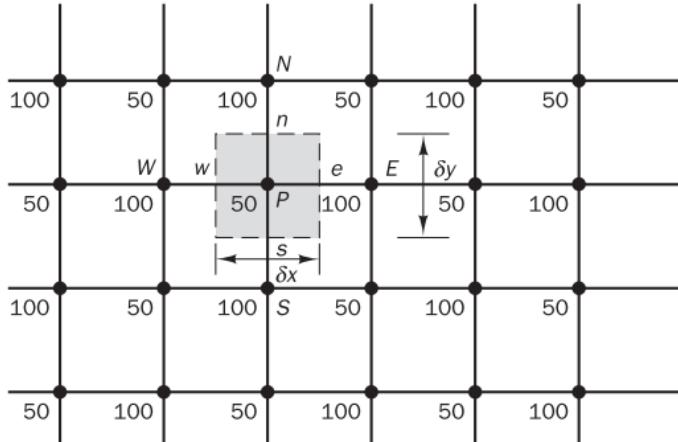


Figura 3.13: Ilustração do *checkerboading*, onde existe oscilação na pressão. Imagem retirada de (VERSTEEG; MALALASEKERA, 2007).

Conforme discutido por (VERSTEEG; MALALASEKERA, 2007), o gradiente poderia ser obtido como:

$$\frac{\partial p}{\partial x} = \frac{p_E - p_W}{2\delta x}, \frac{\partial p}{\partial y} = \frac{p_N - p_S}{2\delta y}, \quad (3.55)$$

em que nenhuma das expressões depende da pressão no nó de referência  $P$ . Substituindo os valores da Figura 3.13, prediz-se que o gradiente de pressão em todos os pontos é zero. Mas, na realidade, isso não é verdade, por haver oscilações da pressão em ambas as direções. Esse comportamento, é obviamente não físico, e deve ser tratado no esquema

numérico.

Um remédio para esse problema é o uso de *grids* escalonados (*staggered grids*) (HARLOW; WELCH, 1965), cuja ideia é avaliar variáveis, tais como pressão ou temperatura, nos centroides, mas deslocar o valor das velocidades para os centros das faces, formando um arranjo desencontrado de variáveis. Contudo, o uso desse tipo de arranjo possui desvantagens críticas. Há alto custo em termos de memória, por ser necessário armazenar o desencontro de velocidades por meio de várias representações das malhas em memória, a fim de facilitar o cálculo e a indexação. Além disso, o fator mais preponderante nesse caso é o uso das malhas não estruturadas, visto que não há a noção de indexação necessária para percorrer e encontrar os deslocamentos.

Desse modo, manter todas as variáveis armazenadas no mesmo lugar torna-se uma necessidade prática na abordagem com malhas não estruturadas. Para isso, utilizou-se a abordagem mais conhecida na literatura, chamada de *Pressure Weighted Interpolation Momentum* (PWIM), popularmente conhecida como Interpolação de Rhie-Chow (RHIE; CHOW, 1983).

### 3.7.1 Interpolação de Rhie-Chow

O grande problema na abordagem colocalizada é o uso de interpolação linear para o cálculo das velocidades nas faces. Essa interpolação resulta em desacoplamento entre pressão e velocidade, o que gera o problema oscilatório. Para contornar isso, em seu *paper* original, Rhie e Chow resolveram esse problema adicionando um novo termo à interpolação, que atua de forma a dissipar o comportamento oscilatório (MOUKALLED; MANGANI; DARWISH, 2015).

A ideia principal da interpolação com PWIM é imitar o *grid* escalonado, porém em uma fórmula fechada. A dedução começa usando a forma algébrica da equação de momento em  $x$ , mas com a pequena diferença em relação à equação (3.54), em que se explicita o termo de pressão:

$$a_O \hat{u}_O + \sum_{f=1}^{N_{f,O}} a_N \hat{u}_N = -\frac{\partial p}{\partial x} \Big|_O V_O, \quad (3.56)$$

no qual  $\hat{u}$  representa a velocidade predita após a resolução da equação de momento, utilizando os valores anteriores de  $u$ ,  $v$  e  $p$ . Na etapa de inicialização do método, tais campos geralmente assumem valores nulos. A discretização do termo de pressão é realizada em um formato distinto, no qual se considera o valor médio da pressão multiplicado pelo volume da célula. Dessa forma, pode-se isolar o valor de  $\hat{u}_O$  de modo que:

$$\hat{u}_O = -\frac{1}{a_O|_O} \sum_{f=1}^{N_{f,O}} a_N|_O \hat{u}_N|_O - \left. \frac{1}{a_O|_O} \frac{\partial p}{\partial x} \right|_O V_O, \quad (3.57)$$

no qual a notação  $|_\square$  indica que o dado coeficiente pertence à célula  $\square$ . Pode-se replicar a equação anterior para uma célula vizinha com índice 1:

$$\hat{u}_1 = -\frac{1}{a_O|_1} \sum_{f=1}^{N_{f,1}} a_N|_1 \hat{u}_N|_1 - \left. \frac{1}{a_O|_1} \frac{\partial p}{\partial x} \right|_1 V_1. \quad (3.58)$$

Nesse momento, é necessário considerar uma equação de velocidade para a face  $f$  entre  $O$  e 1, supondo que haja uma célula  $f$  intermediária. Isso originará:

$$\hat{u}_f = -\frac{1}{a_O|_f} \sum_{f'=1}^{N_{f,O|_f}} a_N|_f \hat{u}_N|_f - \left. \frac{1}{a_O|_f} \frac{\partial p}{\partial x} \right|_f V_f. \quad (3.59)$$

Alguns valores da equação (3.59) precisarão ser obtidos por interpolação linear. Logo:

$$\frac{V_f}{a_O|_f} = w_f \frac{V_O}{a_O|_O} + (1 - w_f) \frac{V_1}{a_O|_1}. \quad (3.60)$$

Além disso, a parte envolvendo o somatório para os vizinhos irá ser transformada em:

$$\frac{1}{a_O|_f} \sum_{f'=1}^{N_{f,O|_f}} a_N|_f \hat{u}_N|_f = w_f \left( \frac{1}{a_O|_O} \sum_{f=1}^{N_{f,O}} a_N|_O \hat{u}_N|_O \right) + (1 - w_f) \left( \frac{1}{a_O|_1} \sum_{f=1}^{N_{f,1}} a_N|_1 \hat{u}_N|_1 \right), \quad (3.61)$$

Substituindo as equações (3.57) e (3.58) em (3.61):

$$-\frac{1}{a_O|_f} \sum_{f'=1}^{N_{f,O|_f}} a_N|_f \hat{u}_N|_f = w_f \left( \hat{u}_O + \left. \frac{1}{a_O|_O} \frac{\partial p}{\partial x} \right|_O V_O \right) + (1 - w_f) \left( \hat{u}_1 + \left. \frac{1}{a_O|_1} \frac{\partial p}{\partial x} \right|_1 V_1 \right), \quad (3.62)$$

Por fim, pode-se substituir (3.60) e (3.62) em (3.59):

$$\begin{aligned}\hat{u}_f &= w_f \left[ \hat{u}_O + \frac{1}{a_O|_O} \frac{\partial p}{\partial x} \Big|_O V_O \right] \\ &\quad + (1 - w_f) \left[ \hat{u}_1 + \frac{1}{a_O|_1} \frac{\partial p}{\partial x} \Big|_1 V_1 \right] - \left[ w_f \frac{V_O}{a_O|O} + (1 - w_f) \frac{V_1}{a_O|_1} \right] \frac{\partial p}{\partial x} \Big|_f,\end{aligned}\tag{3.63}$$

para encontrar finalmente a interpolação para a velocidade  $\hat{u}_f$ . De forma similar, a expressão para  $\hat{v}_f$  é:

$$\begin{aligned}\hat{v}_f &= w_f \left[ \hat{v}_O + \frac{1}{a_O|_O} \frac{\partial p}{\partial y} \Big|_O V_O \right] \\ &\quad + (1 - w_f) \left[ \hat{v}_1 + \frac{1}{a_O|_1} \frac{\partial p}{\partial y} \Big|_1 V_1 \right] - \left[ w_f \frac{V_O}{a_O|O} + (1 - w_f) \frac{V_1}{a_O|_1} \right] \frac{\partial p}{\partial y} \Big|_f.\end{aligned}\tag{3.64}$$

Um fato em relação ao SIMPLE é que não são necessários apenas as velocidades  $\hat{u}_f$  e  $\hat{v}_f$ , mas também os valores do fluxo de massa  $\dot{m}_f$ , utilizados então para o cálculo do termo advectivo, conforme mencionado anteriormente. Para calculá-lo, precisa-se então calcular o produto escalar  $\vec{u}_f \cdot \vec{n}_f$ . Isso resultará em:

$$\begin{aligned}\vec{u}_f \cdot \vec{n}_f &= \hat{u}_f n_{x,f} + \hat{v}_f n_{y,f} \\ &= \left( w_f \left[ \hat{u}_O + \frac{1}{a_O|_O} \frac{\partial p}{\partial x} \Big|_O V_O \right] \right. \\ &\quad \left. + (1 - w_f) \left[ \hat{u}_1 + \frac{1}{a_O|_1} \frac{\partial p}{\partial x} \Big|_1 V_1 \right] - \left[ w_f \frac{V_O}{a_O|O} + (1 - w_f) \frac{V_1}{a_O|_1} \right] \frac{\partial p}{\partial x} \Big|_f \right) n_{x,f} \\ &\quad + \left( w_f \left[ \hat{v}_O + \frac{1}{a_O|_O} \frac{\partial p}{\partial y} \Big|_O V_O \right] \right. \\ &\quad \left. + (1 - w_f) \left[ \hat{v}_1 + \frac{1}{a_O|_1} \frac{\partial p}{\partial y} \Big|_1 V_1 \right] - \left[ w_f \frac{V_O}{a_O|O} + (1 - w_f) \frac{V_1}{a_O|_1} \right] \frac{\partial p}{\partial y} \Big|_f \right) n_{y,f}\end{aligned}\tag{3.65}$$

Que finalmente, em formato vetorial:

$$\begin{aligned}\vec{u}_f \cdot \vec{n}_f &= w_f \vec{u}_O \cdot \vec{n}_f + (1 - w_f) \vec{u}_1 \cdot \vec{n}_f \\ &\quad + w_f \frac{V_O}{a_O|_O} (\nabla p)_O \cdot \vec{n}_f + (1 - w_f) \frac{V_1}{a_O|_1} (\nabla p)_1 \cdot \vec{n}_f - \left[ w_f \frac{V_O}{a_O|O} + (1 - w_f) \frac{V_1}{a_O|_1} \right] (\nabla p)_f \cdot \vec{n}_f\end{aligned}\tag{3.66}$$

representa o cômputo dos valores do fluxo de massa nas faces adotado neste trabalho. A equação (3.66) consiste basicamente em uma interpolação linear, mas adiciona termos que

envolvem a pressão, o que traz estabilidade. Na equação (3.66), os termos que envolvem o gradiente de pressão nas células  $O$  e  $1$  podem ser calculados como

$$V_O(\nabla p)_O \cdot \vec{n}_f = n_{x,f} \sum_{ff(O)} p_{ff} A_{ff} n_{x,ff} + n_{y,f} \sum_{ff(O)} p_{ff} A_{ff} n_{y,ff}, \quad (3.67)$$

em que utilizou-se a notação  $ff$  para indexar o *loop* nas faces e não existir confusão com o  $f$  da face interpolada. E o termo que envolve o gradiente de pressão, mas na face  $f$ , pode ser determinado como

$$(\nabla p)_f \cdot \vec{n}_f = \frac{p_1^{(k)} - p_O}{d_{O1}} + \zeta, \quad (3.68)$$

onde existiria um termo  $\zeta$  a ser contabilizado devido a não ortogonalidade da malha. Entretanto, o mesmo é negligenciável, não impactando na convergência do algoritmo (MAZUMDER, 2015).

### 3.8 Derivação da equação para pressão

Conforme descrito no início do capítulo, é necessário derivar uma equação específica para o cálculo do campo de pressão. A partir dos valores obtidos na solução das equações de momento e da interpolação de Rhie-Chow, o campo de velocidades passa a satisfazer, em boa aproximação, a equação de momento, mas ainda não satisfaz exatamente a equação de continuidade.

Dessa forma, torna-se necessário corrigir os campos de pressão, velocidade e fluxo de massa, de modo a impor simultaneamente o atendimento à continuidade. Denotando-se as correções por meio de um sobrescrito “ $\square$ ”, as relações entre o valor exato, o valor aproximado “ $\hat{\square}$ ” e a correspondente correção podem ser escritas da seguinte forma:

$$\begin{aligned} \vec{v} &= \hat{\vec{v}} + \vec{v}', \\ p &= \hat{p} + p', \\ \dot{m} &= \hat{\dot{m}} + \dot{m}'. \end{aligned} \quad (3.69)$$

parte-se, então, da Equação (3.57). De forma análoga, os valores exatos também satisfa-

zem essa equação e podem ser nela substituídos. Ao se subtrair a equação aproximada da equação exata, obtém-se, para a célula  $O$ , a seguinte expressão para a correção da velocidade:

$$u'_O = -\frac{1}{a_O|_O} \sum_{f=1}^{N_{f,O}} a_N|_O u'_N - \frac{1}{a_O|_O} \left. \frac{\partial p'}{\partial x} \right|_O V_O. \quad (3.70)$$

Nesse ponto, o algoritmo SIMPLE introduz uma simplificação importante na primeira parcela da correção de velocidade: por simplicidade, o termo que envolve as correções  $u'_N$  é negligenciado, assumindo-se que, ao longo das iterações, essa contribuição tende a zero e que o termo associado ao gradiente de pressão passa a dominar a expressão.

Utilizando uma discretização análoga àquela empregada na seção de tratamento do termo de pressão (3.49), pode-se escrever para  $u$  e  $v$ :

$$u'_O = -\frac{1}{a_O|_O} \sum_{f(O)} p'_f A_f n_{x,f}, \quad (3.71)$$

$$v'_O = -\frac{1}{a_O|_O} \sum_{f(O)} p'_f A_f n_{y,f}, \quad (3.72)$$

em que  $p'_f$  é a correção de pressão interpolada na face  $f$ , e a soma é realizada sobre as faces do volume de controle associado à célula  $O$ . De forma análoga, pode-se derivar a expressão da velocidade nas faces. O detalhe algébrico é omitido aqui para brevidade, mas, após algumas manipulações e o uso da Equação (3.66), chega-se a uma relação do tipo:

$$u'_f = - \left( w_f \frac{V_O}{a_O|_O} + (1 - w_f) \frac{V_1}{a_O|_1} \right) \left. \frac{\partial p'}{\partial x} \right|_f, \quad (3.73)$$

$$v'_f = - \left( w_f \frac{V_O}{a_O|_O} + (1 - w_f) \frac{V_1}{a_O|_1} \right) \left. \frac{\partial p'}{\partial y} \right|_f, \quad (3.74)$$

em que  $w_f$  é dado por (3.23),  $V_O$  e  $V_1$  são os volumes das células adjacentes à face, e  $a_O|_O$  e  $a_O|_1$  são os coeficientes diagonais correspondentes na discretização das equações de momento.

A equação de continuidade discretizada é exatamente satisfeita quando se considera o campo de velocidades exato. Entretanto, no processo iterativo, dispõe-se apenas de um campo aproximado, que, nesse estágio, satisfaz a equação de momento, mas não necessariamente a continuidade. Manipulando-se a equação de continuidade, obtém-se a

seguinte relação para as correções de velocidade:

$$\sum_{f(O)} \rho_f (\vec{u}' \cdot \vec{n}_f) A_f = - \sum_{f(O)} \rho_f (\hat{\vec{u}} \cdot \vec{n}_f) A_f, \quad (3.75)$$

em que o somatório é tomado sobre as faces do volume de controle associado à célula  $O$ ,  $\hat{\vec{u}}$  representa o campo de velocidade aproximado (obtido na etapa anterior) e  $\vec{u}'$  é a correção de velocidade. O lado direito representa, portanto, o desequilíbrio de massa (resíduo de continuidade) calculado com as velocidades aproximadas.

Após substituições adicionais para  $\vec{u}'$  em função da correção de pressão, obtém-se a equação de correção de pressão na forma:

$$\sum_{f(O)} \rho_f \left[ w_f \frac{V_O}{a_O|_O} + (1 - w_f) \frac{V_1}{a_O|_1} \right] \left( \frac{p'_O - p'_N}{d_{O1}} \right) A_f = - \sum_{f(O)} \rho_f (\hat{\vec{u}} \cdot \vec{n}_f) A_f, \quad (3.76)$$

Com base nessa equação, obtém-se os valores de  $p'$  e o campo de pressão é atualizado; as correções de velocidade são calculadas e, em seguida, as velocidades são corrigidas. Ao final dessa etapa, o campo corrigido passa a satisfazer a continuidade, mas deixa de satisfazer exatamente as equações de momento, de modo que o processo é repetido iterativamente até um determinado critério de convergência.

## 3.9 Coeficientes de relaxação

Uma característica crítica do SIMPLE está na sua convergência. Embora a forma algébrica da equação do momento seja linear, seus coeficientes dependem dos campos de velocidade e pressão. Essa não linearidade é tratada por um processo iterativo durante o qual os coeficientes são calculados no início de cada iteração com base nos valores das variáveis dependentes obtidos na iteração anterior.

Essa mudança nos valores dos coeficientes resulta em grandes variações na velocidade e afeta a taxa de convergência, o que pode originar divergências. Para reduzir essas variações, pode-se aplicar sub-relaxações. Denotando esse coeficiente de sub-relaxação para a velocidade como  $\lambda_{uv}$ , a equação do momento pode ser escrita como (MOUKAL-

LED; MANGANI; DARWISH, 2015):

$$\frac{a_O}{\lambda_{uv}} u_O + \sum_{f(O)} a_N u_N = Q_x + \frac{1 - \lambda_{uv}}{\lambda_{uv}} a_O u_O^{(n)}, \quad (3.77)$$

em que  $u_O^{(n)}$  é o valor de  $u_O$  na iteração anterior do SIMPLE. Observa-se que, além de modificar o coeficiente diagonal (efetivamente ampliando  $a_O$ ), o termo da iteração anterior é incorporado ao lado direito, contribuindo para estabilizar a evolução do campo de velocidades.

De modo análogo, e em virtude das simplificações introduzidas pelo algoritmo SIMPLE na dedução da equação de correção de pressão, é conveniente aplicar sub-relaxação também na atualização da pressão. Não há necessidade de sub-relaxar diretamente a correção de velocidade, uma vez que é a pressão corrigida que garante, em última instância, a conservação de massa. Sendo assim, denotando o coeficiente de relaxação para a pressão por  $\lambda_p$ , a atualização de  $p$  é escrita como

$$p = \hat{p} + \lambda_p p', \quad (3.78)$$

em que  $\hat{p}$  representa o campo de pressão vigente e  $p'$  é a correção obtida a partir da equação de correção de pressão. Valores de  $\lambda_p$  inferiores a 1 tornam a atualização mais conservadora, melhorando a estabilidade do processo iterativo em detrimento de uma convergência potencialmente mais lenta.

## 3.10 Tratamento das condições de contorno

Um elemento de contorno é aquele que possui ao menos uma face localizada na fronteira do domínio, denominada *face de contorno*. O tratamento das condições de contorno nessas faces é fundamental para a formulação numérica.

Na interpolação de Rhie-Chow, os valores das faces de contorno são utilizados diretamente quando a condição de contorno prescreve um valor. Ou seja:  $\dot{m}_f = \square$ , onde  $\square$  representa o valor informado da condição de contorno. Neste trabalho, optou-se por tratar apenas as condições de contorno *Wall*, *Inlet* e *Outlet*, justamente pela conveniência

dessas condições nos problemas utilizados.

No caso de ser uma condição do tipo *Wall*, ela pode ser classificada como *slip* (deslizante) ou *no slip* (fixa). Os seus valores relacionados são parametrizados como:  $p_b = ?; \dot{m}_b = 0; \vec{u} = \vec{u}_{wall}$ , onde o símbolo de interrogação “?” significa que o valor é desconhecido. O fluxo de massa é zero, implicando não haver velocidade em direções ortogonais ao contorno. Além disso, a velocidade do fluido perto da parede é igual à velocidade da parede  $\vec{u}_{wall}$ . Em termos de implementação numérica, esses valores podem ser armazenados e substituídos sempre que necessário na avaliação dos fluxos nas faces de contorno, contribuindo como termos fonte nas equações discretizadas.

A pressão desconhecida  $p_b$  pode ser extrapolada a partir dos valores internos do domínio por meio de uma expansão em série de Taylor. Por simplicidade, truncou-se a série já no primeiro termo da expansão, o que implica que  $p_b = p_O$  (a pressão na face de contorno é a pressão da célula vizinha).

No caso da condição *inlet*, a mesma é parametrizada por:  $p_b = ?; \dot{m}_b = \square; \vec{u}_b = \square$ . Nesse caso, existe um fluxo de massa, e a velocidade possui componente ortogonal à condição de contorno, fazendo a massa entrar no domínio. Similarmente à condição *wall*, pode-se usar a mesma expressão para aproximar o valor de  $p_b$ . Na terceira e última condição, *Outlet*, ela é parametrizada por:  $p_b = \square; \dot{m}_b = ?; \vec{u}_b = ?$ . Neste caso, é informado agora o valor da pressão, mas o valor da velocidade não é especificado. Também é necessário extrapolar o valor do fluxo de massa no contorno para ser o valor da célula *O*.

Sobre a equação de correção de pressão, há apenas os últimos pormenores. Primeiro, no caso da condição de contorno *wall*, como o fluxo de massa é padronizado como 0, não há a necessidade de corrigir o valor no contorno durante a execução, ou seja,  $\dot{m}'_b = 0$ . De forma análoga, o mesmo se verifica para a condição *inlet*. No caso do valor de  $p'$ , o tratamento é consistente, e o valor é extrapolado pegando o valor da célula vizinha. Para a condição *Outlet*, como há o valor de  $p_b$ , a sua correção é então utilizada como  $p'_b = 0$ .

## 4 DESCRIÇÃO DO CÓDIGO $R_e$ FUM2D

$R_e$ FUM2D é um *software* escrito na linguagem C++ que permite a partir dos dados informados pelo usuário simular problemas de escoamentos de fluidos envolvendo as equações de Navier-Stokes utilizando a metodologia do MVF. O  $R_e$ FUM2D é dividido em quatro partes principais: processamento das informações pertinentes à simulação, pré-processamento da malha, processamento e exportação da solução; todos feitos de forma serial, ou seja, sem utilização de programação *multithreading* ou via GPUs. O  $R_e$ FUM2D foi construído utilizando a biblioteca de álgebra linear **Eigen** (GUENNEBAUD; JACOB, 2010), descrita melhor posteriormente. A estrutura do  $R_e$ FUM2D está exibida na Figura 4.1.

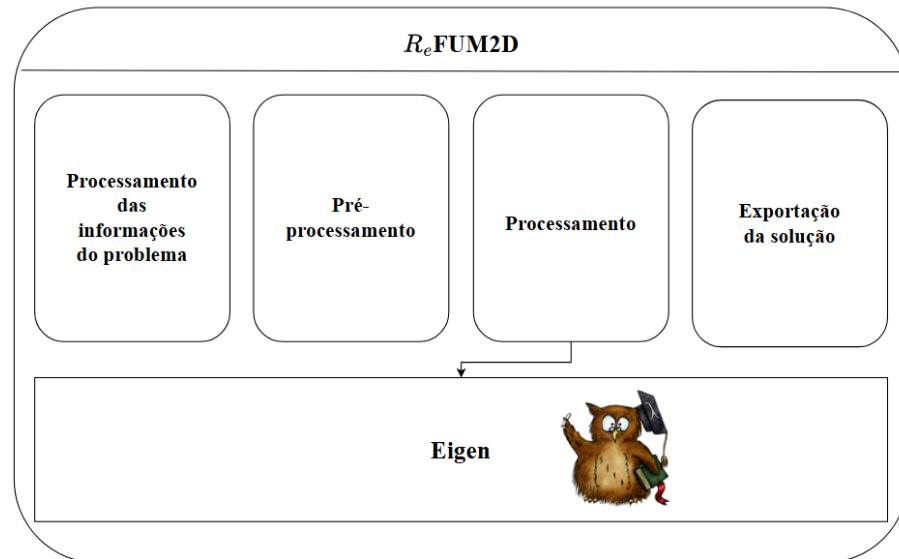
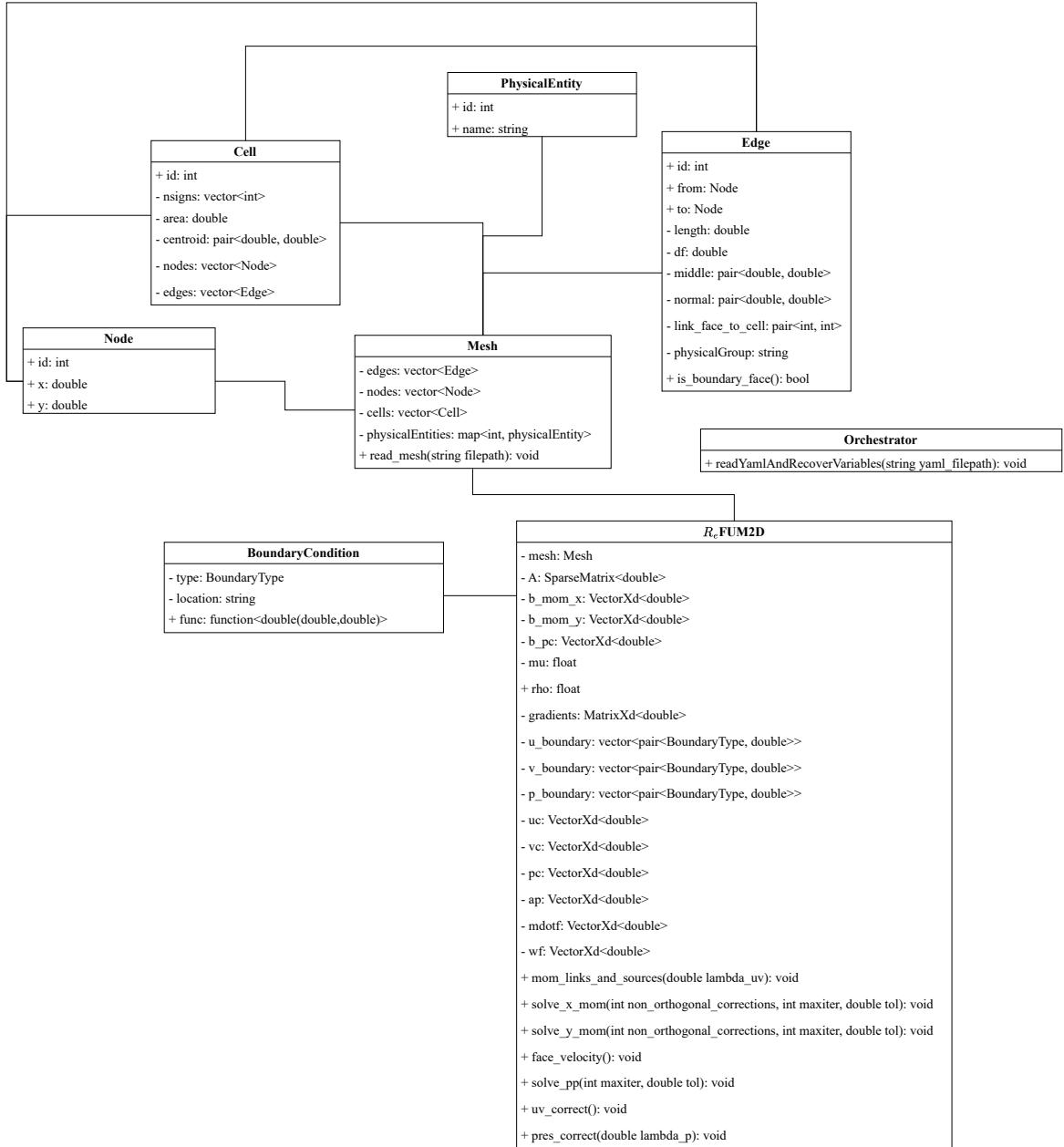


Figura 4.1: Visão geral dos componentes.

Além disso, como o *software* foi construído seguindo uma abordagem orientada a objetos, a Figura 4.2 retrata um diagrama de classes que exibe as classes utilizadas com seus atributos e métodos, além dos seus respectivos relacionamentos.

Figura 4.2: Diagrama de classes do *R<sub>e</sub>FUM2D*.

## 4.1 Processamento das informações do problema

Nesta primeira etapa, são feitas a leitura da malha informada pelo usuário, e também os dados da simulação. Como entrada, o *R<sub>e</sub>FUM2D* suporta malhas com extensão `.msh`, na versão 2.0 ASCII. Este formato é do software **Gmsh** (GEUZAINÉ; REMACLE, 2009), um código livre de geração e visualização de malhas bidimensionais e tridimensionais. Para a leitura dos dados do usuário, é necessário informar como argumento do programa o caminho até um arquivo **YAML** (*YAML Ain't Markup Language*), onde nele estarão descritas

as propriedades físicas do problema, detalhes de condições de contorno e informações do algoritmo SIMPLE. Um exemplo de arquivo está descrito na Figura 4.3:

```

1 problem:
2   name: Lid-Driven Cavity Flow
3   mu: 1e-2
4   rho: 1.0
5   reynolds: 100
6   boundaries:
7     u:
8       - { type: DIRICHLET, region: TOP, value: one }
9       - { type: DIRICHLET, region: BOTTOM, value: zero }
10      - { type: DIRICHLET, region: LEFT, value: zero }
11      - { type: DIRICHLET, region: RIGHT, value: zero }
12
13     v:
14       - { type: DIRICHLET, region: TOP, value: zero }
15       - { type: DIRICHLET, region: BOTTOM, value: zero }
16       - { type: DIRICHLET, region: LEFT, value: zero }
17       - { type: DIRICHLET, region: RIGHT, value: zero }
18
19     p:
20       - { type: NEUMANN, region: TOP, value: zero }
21       - { type: NEUMANN, region: BOTTOM, value: zero }
22       - { type: NEUMANN, region: LEFT, value: zero }
23       - { type: NEUMANN, region: RIGHT, value: zero }
24
25 mshfile: ./inputs/lid_driven_cavity_flow_8656.msh
26
27 simple:
28   momentum:
29     lambda_uv: 0.6
30     non_corrections: 6
31     iterations_bicgstab: 200
32     tolerance_bicgstab: 1e-6
33
34   pressure_correction:
35     lambda_p: 0.3
36     iterations_bicgstab: 100
37     tolerance_bicgstab: 1e-4
38
39   utol: 1e-8
40   vtol: 1e-8
41   ptol: 1e-8
42
43 save_iterations: yes
44 exportfolder: ./outputs/

```

Figura 4.3: Exemplo de um arquivo contendo as informações de entrada do problema. Por praticidade, quebrou-se a visualização em duas partes, sendo a primeira metade na esquerda.

em que são fornecidos o caminho até a malha, os parâmetros físicos: densidade para a parte convectiva e viscosidade para a parte difusiva, e as condições de contorno para cada uma das variáveis. Os tipos referem-se a como o valor será aplicado, sendo as opções “DIRICHLET” para um valor prescrito, ou “NEUMANN” para prescrever o valor do gradiente. As regiões são os respectivos locais onde a condição de contorno será aplicada, valores esses definidos dentro do `Gmsh`, e que precisam estar em acordo. O campo “*value*” corresponde ao valor da condição, e pode ser uma função, como:  $one(x, y) = 1$  ou  $zero(x, y) = 0$ . Tais valores, infelizmente, precisam ser definidos dentro do código e referenciados com o mesmo nome, visto que a definição direta no arquivo exigiria um tratamento de expressões simbólicas, processo que está além do escopo deste trabalho. Ainda há campos contendo as informações para execução do SIMPLE e detalhes para a exportação da solução. Dentro do código criado, o tratamento dessa etapa está contido na classe *Orchestrator*, que fica responsável por processar todos os dados do *YAML* para disponibilizá-los conforme requerido. A leitura da malha fica contida dentro da classe *Mesh*.

## 4.2 Pré-processamento

A malha é representada computacionalmente na classe `Mesh`. Esta classe é composta por três principais *arrays*: nós da malha (vértices dos volumes de controle), elementos da malha (células) e arestas (faces dos volumes de controle). Considerar-se-á a malha da Figura 4.4 para exemplificar alguns detalhes nesta etapa. As inspirações para realizar o pré-processamento estão detalhadas de melhor forma em (MAZUMDER, 2015). Como principais pontos, tem-se que:

- Cada célula possui uma numeração correspondente no *array* de células.
- Cada face possui uma numeração correspondente no *array* de faces.
- Cada vértice possui uma numeração correspondente no *array* de vértices.
- Cada face possui uma numeração local, que pode ir de 0 até  $|f| - 1$ , onde  $|f|$  denota o número de faces do volume de controle.
- Em relação a uma dada face, as células também podem possuir numeração local valendo 0 ou 1.

A relação espacial que se dá entre os valores globais e locais é denominada: conectividade. Uma conectividade que precisa ser determinada para o MVF é chamada de: *link\_face\_to\_cell*, onde sua ideia é recuperar o *id* das células que compartilham uma determinada face. Baseado na figura 4.4, as seguintes chamadas dessa função são verdadeiras: *link\_face\_to\_cell(20,0) = 8* e *link\_face\_to\_cell(20,1) = 12* ou *link\_face\_to\_cell(9,0) = 3* e *link\_face\_to\_cell(9,1) = 2*. Outra funcionalidade da mesma é permitir identificar uma face como face de contorno, visto que quando uma aresta possui somente uma célula compartilhando-a (situação do contorno), uma das numerações locais referencia a *flag* “-1”. A partir dessa relação, pode-se usá-la para construir o Algoritmo 1. Nele, em detrimento da conectividade entre face e células, pode-se resgatar as células adjacentes, e interpolar na face o valor da propriedade segundo a equação 3.23. os próximos processamentos estão relacionados à questões geométricas, e com isso dedicou-se subseções próprias aos mesmos.

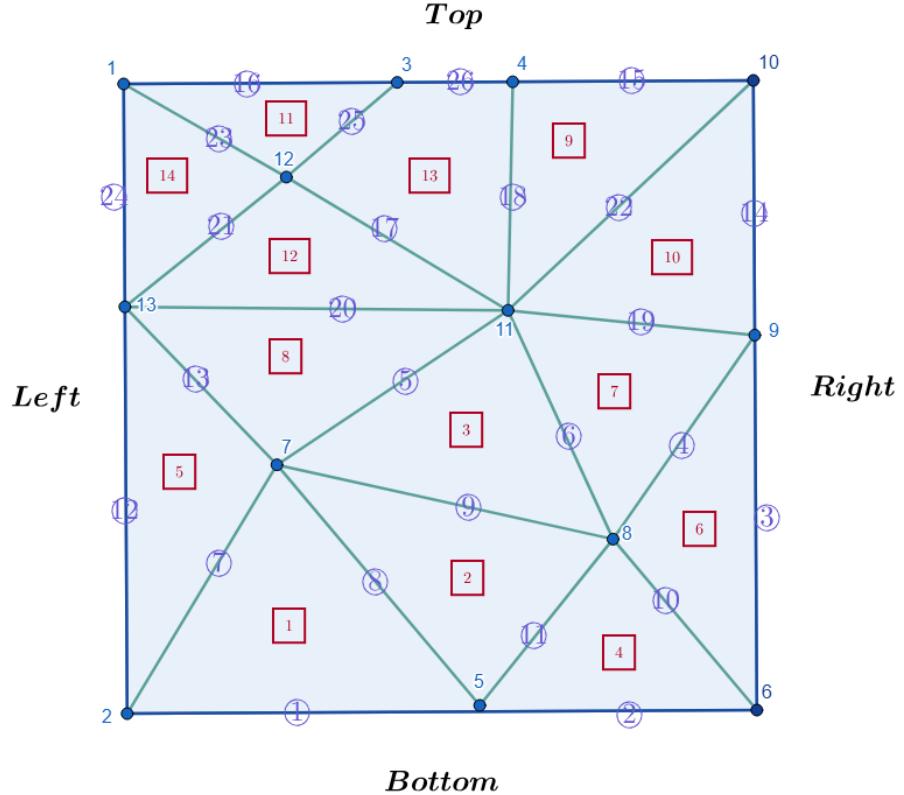


Figura 4.4: Exemplo de malha não estruturada bidimensional. Os vértices da malha são destacados em azul, os centroides dos volumes de controle em vermelho e as faces em roxo. As regiões de contorno do domínio (*Top*, *Bottom*, *Left*, *Right*) estão explicitamente identificadas. Imagem criada com base em (MAZUMDER, 2015).

---

**Algoritmo 1:** Uso da conectividade *link\_face\_to\_cell*


---

```

1 for if ← 1 to nfaces do
2   | ic1 ← link_face_to_cell(iface, 0);
3   | ic2 ← link_face_to_cell(iface, 1);
4   | vf = interpolate_values(ic1, ic2);
5 end for

```

---

### 4.2.1 Cálculo do volume das células

O volume das células é uma informação importante a ser encontrada, e não é informado pelo gerador de malhas. Sendo assim, considere um vetor  $\vec{q} = x\hat{i}$ . Tomando o divergente desse campo vetorial:

$$\nabla \cdot \vec{q} = \nabla \cdot (x\hat{i}) = \frac{\partial}{\partial x}(x)\hat{i} \cdot \hat{i} = 1. \quad (4.1)$$

uma vez que o divergente desse vetor  $\vec{q}$  é unitário, pode-se dizer então que:

$$V_O = \int_{V_O} dV = \int_{V_O} \nabla \cdot \vec{q} dV, \quad (4.2)$$

nesse formato, pode-se recorrer ao Teorema de Gauss 3.9 e transformar a integral em um somatório nas faces, de modo que:

$$V_O = \sum_{f=1}^{N_{f,O}} x_f n_{x,f} A_f. \quad (4.3)$$

De maneira análoga pode ser feito para  $\vec{q} = y\hat{j}$ , a fim de derivar uma expressão do volume usando a direção  $y$ , e depois extrair uma média das duas versões para aumentar a robustez do cálculo.

#### 4.2.2 Cálculo das áreas das faces e informações sobre as normais

Em 2D, calcular a área da face consiste apenas em calcular o comprimento de um segmento usando a distância euclidiana. Já no caso do vetor normal, o mesmo pode ser obtido usando algumas relações envolvendo a tangente à face. Os componentes tangentes podem ser encontrados como:

$$t_{x,f} = \frac{x_2 - x_1}{A_f}, \quad t_{y,f} = \frac{y_2 - y_1}{A_f} \quad (4.4)$$

em vista de obter-se a normal, pode-se usar relações de geometria analítica. A primeira delas é que a normal e tangente são perpendiculares. A segunda, é que o produto vetorial entre os dois vetores é um vetor ortogonal e unitário  $\hat{k}$  na direção  $z$ . Assim:

$$\begin{cases} \vec{n}_f \cdot \vec{t}_f = 0 \\ \vec{n}_f \times \vec{t}_f = \vec{k} \end{cases} \quad (4.5)$$

resolvendo o sistema obtém-se que:  $n_{x,f} = t_{y,f}$  e  $n_{y,f} = -t_{x,f}$ . Um ponto crítico é que, apesar de saber os valores das componentes da normal, algo igualmente importante é conhecer em qual sentido ela aponta. Para aplicar o teorema da divergência de Gauss, uma hipótese importante é que todas as normais apontam para fora do volume de controle em questão. Portanto, durante a execução do *software*, precisa-se guardar qual sentido

a normal aponta e corrigir sempre que necessário para realizar os balanços. Uma ideia disso está exemplificada no Algoritmo 2, no qual cada célula possuirá um vetor interno armazenando quais os sinais das normais para cada face, onde 1 indica que a normal já aponta para fora, e -1 não.

---

**Algoritmo 2:** Determinação dos sinais das normais

---

```

1 for  $ic \leftarrow 1$  to  $ncells$  do
2   for  $if \leftarrow 1$  to  $nfaces(ic)$  do
3      $ic1 = link\_face\_to\_cell(ic, 0);$ 
4     if ( $icell == ic1$ ) then
5       |  $nsign(ic, if) = 1;$ 
6     else
7       |  $nsign(ic, if) = -1;$ 
8     end if
9   end for
10 end for

```

---

Uma vez que a direção das normais está bem definida, pode-se usá-las para o cômputo do volume de uma célula baseado na equação 4.3, onde o Algoritmo 3 ilustra a sua execução aproveitando a estrutura  $nsign$  construída anteriormente.

---

**Algoritmo 3:** Cálculo do volume utilizando a informação de  $nsign$ 


---

```

1 for  $ic \leftarrow 1$  to  $ncells$  do
2    $vol(ic) = 0;$ 
3   for  $if \leftarrow 1$  to  $nfaces(ic)$  do
4     |  $vol(ic) = vol(ic) + nsigns(ic, ifc) * normal(ic).x * xf(if) * area(if);$ 
5   end for
6 end for

```

---

## 4.3 Processamento

Após a execução do pré-processamento dentro da classe *Mesh*, que irá guardar então informações das classes *Node* (vértices), *Cell* (células) e *Edge* (faces), além de informações de contorno manifestadas através da classe *PhysicalEntity*, a próxima etapa então é montar os sistemas de equações e resolvê-los utilizando algum método de resolução de sistemas. Toda a lógica do processamento encontra-se dentro da classe *R<sub>e</sub>FUMSolver*, que irá executar as etapas do algoritmo SIMPLE para então resolver o problema. Nessa

etapa, a biblioteca *Eigen* possui papel vital, possibilitando performar de forma mais eficiente os cálculos algébricos.

A biblioteca *Eigen* é uma biblioteca em C++ amplamente utilizada para operações de álgebra linear, oferecendo uma interface eficiente e flexível para o manuseio de vetores e matrizes densas e esparsas. Entre suas principais funcionalidades, destaca-se o suporte nativo a matrizes esparsas, o que a torna particularmente adequada para aplicações científicas e de engenharia que envolvem grandes sistemas lineares (GUENNEBAUD; JACOB, 2010).

Resgatando o que foi discutido na seção de Metodologia Numérica, o SIMPLE pode ser descrito de forma algorítmica através do pseudocódigo 4, que ilustra em alto nível a sequência de chamadas que o *R<sub>e</sub>FUM2D* utiliza na prática.

---

**Algoritmo 4:** Algoritmo SIMPLE

---

**Entrada:**  $utol, vtol, ptol, non\_corrections, iter\_bicgstab\_mom,$   
 $tol\_bicgstab\_mom, iter\_bicgstab\_pc, tol\_bicgstab\_pc, \lambda_{uv}, \lambda_p$ .

```

1 while  $uerr > utol \vee verr > vtol \vee perr > ptol$  do
2    $mom\_links\_and\_sources(\lambda_{uv});$ 
3    $solve\_x\_mom(non\_corrections, iter\_bicgstab\_mom, tol\_bicgstab\_mom);$ 
4    $solve\_y\_mom(non\_corrections, iter\_bicgstab\_mom, tol\_bicgstab\_mom);$ 
5    $face\_velocity();$ 
6    $solve\_pp(iter\_bicgstab\_pc, tol\_bicgstab\_pc);$ 
7    $uv\_correct();$ 
8    $pres\_correct(\lambda_p);$ 
9 end while
```

---

A primeira etapa do algoritmo consiste na inicialização das estruturas de dados, sendo a maior parte delas inicializada com zeros, caracterizando o chute inicial do método iterativo. Em seguida, procede-se à montagem dos sistemas das equações de momento nas direções  $x$  e  $y$  por meio da função *mom\_links\_and\_sources*, onde montam-se as matrizes  $A$  e  $b$  do sistema, sendo que a matriz  $A$  é montada com auxílio da estrutura de dados *triplets*, que serve como base para montar uma matriz esparsa dentro do *Eigen*. A resolução desses sistemas é realizada pelas rotinas *solve\_x\_mom* e *solve\_y\_mom*, as quais empregam o método iterativo BiCGSTAB pré-condicionado com Jacobi.

No interior dessas rotinas, são executados os procedimentos de correção da não ortogonalidade da malha, bem como o cálculo dos gradientes associados ao esquema de discretização LUD. Por razões de economia computacional, optou-se por realizar de forma

conjunta as correções explícitas, em vez de aplicar separadamente a correção de não ortogonalidade e, em seguida, tratar exclusivamente o esquema LUD, processo considerado mais adequado pelos autores. O método BiCGSTAB requer a definição do número máximo de iterações e da tolerância de convergência, parâmetros fornecidos pelo usuário. Com o objetivo de acelerar a convergência, utiliza-se o pré-condicionador de Jacobi.

Após a resolução das equações de momento, realiza-se a interpolação do fluxo de massa nas faces dos volumes de controle com base no esquema de Rhie–Chow, procedimento implementado na função *face\_velocity*. Nesse estágio, o campo de velocidades satisfaz as equações de momento, porém ainda não atende à equação de continuidade. Para corrigir essa inconsistência, a função *solve\_pp* monta e resolve uma equação de correção de pressão, cuja finalidade é forçar o campo de velocidades a satisfazer a conservação de massa.

Na sequência, os valores de velocidade e pressão são atualizados com base na solução da equação de pressão, por meio das rotinas *uv\_correct* e *p\_correct*. Essas atualizações incorporam fatores de sub-relaxação, previamente definidos, com o objetivo de melhorar a estabilidade numérica e favorecer a convergência do método.

O algoritmo é executado iterativamente até que todas as tolerâncias especificadas pelo usuário no arquivo *YAML* sejam satisfeitas. Para a avaliação do erro em cada iteração, mantém-se um vetor auxiliar contendo os valores da iteração anterior, permitindo o cálculo da norma do máximo e a verificação do critério de convergência adotado.

Por fim, para ilustrar visualmente as etapas do algoritmo, construiu-se o fluxograma disposto na figura 4.5. Como é possível perceber, o algoritmo é composto por diversas etapas, sendo custoso do ponto de vista computacional. Denotando a quantidade de células por  $|c|$  e quantidade de faces por célula como  $|f|$ , grande parte dos *loops* para preenchimento dos sistemas lineares são da ordem de  $O(|c| \cdot |f|)$ . Sendo assim, desprezando a resolução dos sistemas, somente para a montagem dos sistemas lineares o algoritmo possui complexidade  $O(k \cdot |c| \cdot |f|)$ , onde  $k$  é a quantidade de iterações no *loop* externo do SIMPLE, que costumam ser da ordem de milhares, evidenciando uma demanda natural do algoritmo para estratégias de paralelização.

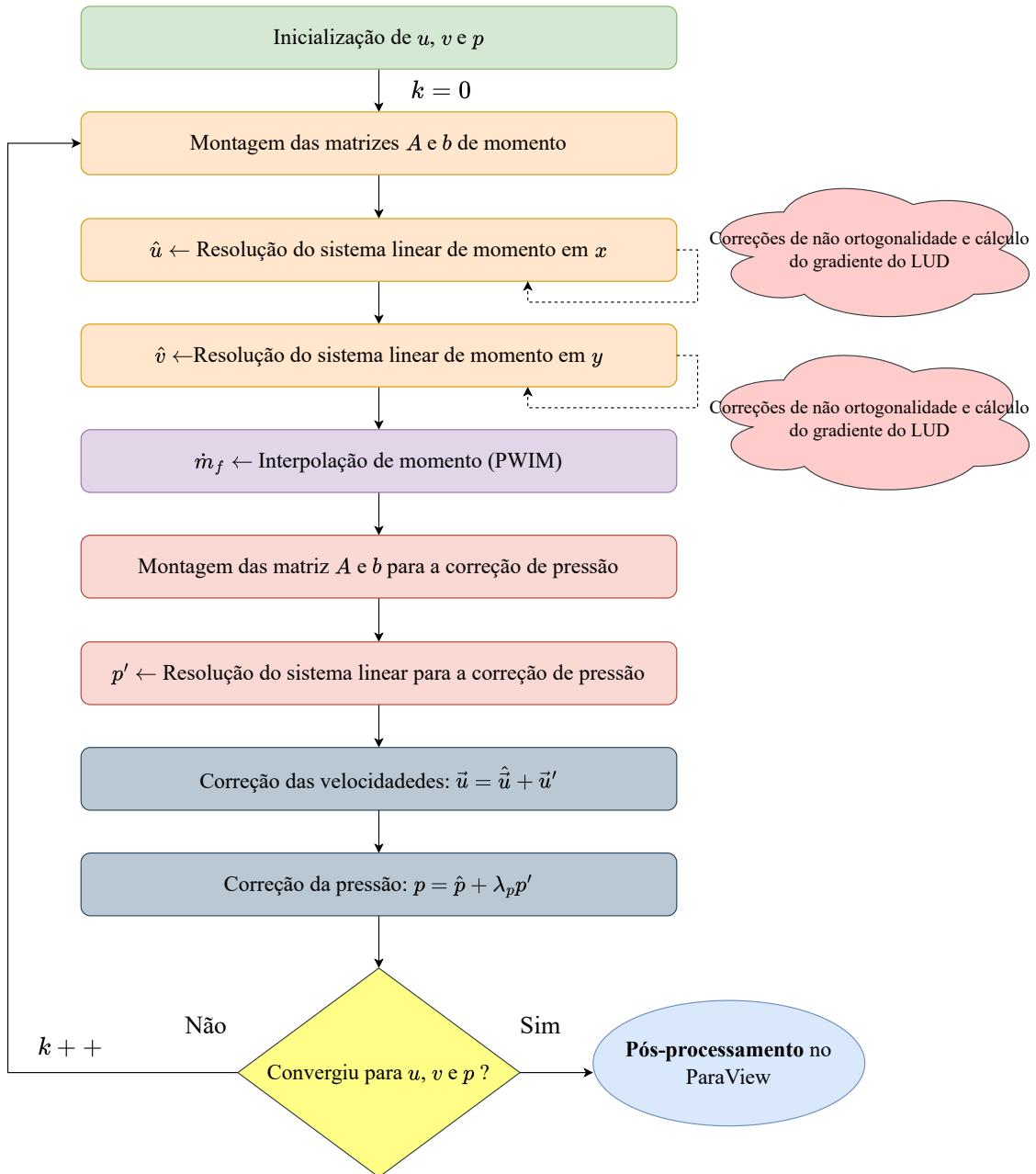


Figura 4.5: Fluxograma ilustrando visualmente as etapas de processamento.

## 4.4 Exportação da solução

O *R<sub>e</sub>FUM2D* faz a escrita da solução utilizando o formato VTK (*Visualization Toolkit*) (Kitware, 2026), utilizado para a escrita de resultados em *grids* não-estruturados e baseado na sintaxe XML, uma sintaxe voltada para formatos de conjunto de dados topologicamente irregulares. Uma grande vantagem do formato VTK é ter suporte nativo pelo *ParaView* (AHRENS; GEVECI; LAW, 2005), um código *open-source* para visualização científica, que foi utilizado neste trabalho para pós-processamento e visualização das soluções.

## 5 RESULTADOS

Os resultados são apresentados com foco na precisão e corretude das soluções geradas pelo método numérico desenvolvido, iniciando-se pela análise da ordem de convergência e, em seguida, pela validação do código por meio de problemas clássicos da área de CFD. No algoritmo SIMPLE, as tolerâncias para velocidade e pressão foram fixadas em  $1 \times 10^{-8}$ , valores amplamente adotados por estarem próximos da precisão de máquina simples, enquanto os coeficientes de sobre-relaxação foram ajustados de acordo com o problema considerado. Para as iterações explícitas, adotou-se  $\theta = 5$ , correspondendo a cinco iterações, e o processo de avaliação da convergência seguiu os mesmos critérios descritos nas etapas de difusão e convecção. O solucionador BiCGSTAB foi empregado na resolução das equações de momento, com tolerância de  $1 \times 10^{-6}$  e limite de 200 iterações, ao passo que a equação de pressão foi resolvida com tolerância de  $1 \times 10^{-4}$  e até 100 iterações.

Os testes foram realizados em um *laptop* equipado com um processador Intel<sup>®</sup> Core<sup>™</sup> i5-12450HX de 12<sup>a</sup> geração, 16 GB de memória RAM e 512 GB de armazenamento interno em SSD. O ambiente de execução adotado foi o sistema operacional Ubuntu 22.04 LTS, e a versão da linguagem C++ utilizada foi o padrão C++17, conforme disponibilizado pelo *toolchain* configurado via CMake.

### 5.1 Fluxo de Kovasznay

Nessa seção, testa-se o comportamento de convergência do esquema desenvolvido utilizando o problema do Fluxo de Kovasznay, um *benchmark* bem popular dentro da área de CFD (CHEN et al., 2021). Esse problema é de interesse, visto que possui solução analítica, a qual é descrita por (KOVASZNAY, 1948; PIETRO; ERN, 2011):

$$\begin{aligned}\vec{u}(x, y) &= \left(1 - e^{\lambda x} \cos(2\pi y), \frac{\lambda}{2\pi} e^{\lambda x} \sin(2\pi y)\right), \\ p(x, y) &= \frac{1}{2}(1 - e^{2\lambda x}).\end{aligned}\tag{5.1}$$

Onde o valor de  $\lambda$  está relacionado ao número de Reynolds:

$$\lambda = \frac{R_e}{2} - \sqrt{\frac{R_e^2}{4} + 4\pi^2}. \quad (5.2)$$

Para realizar a simulação tomou-se  $R_e = 20$ , resultando em um problema com  $\lambda \approx -1.81$ . O domínio da simulação é:  $\Omega := [-0.5, 0] \times [1.5, 2]$ . Ao contrário das simulações feitas anteriormente, dessa vez utilizou-se uma malha não estruturada composta apenas por elementos triangulares, gerada no `gmsh` com o algoritmo de triangulação de Delaunay. No caso dos valores de  $\lambda_{uv}$  e  $\lambda_p$ , optou-se por serem 0.6 e 0.3, respectivamente. A quantidade de volumes de controle na malha foram:  $N = 460, 942, 1992, 3962, 7826, 15644$ ; dobrando aproximadamente a quantidade. O fator de refinamento e cálculo da norma são calculados com as equações (3.38) e (3.37).

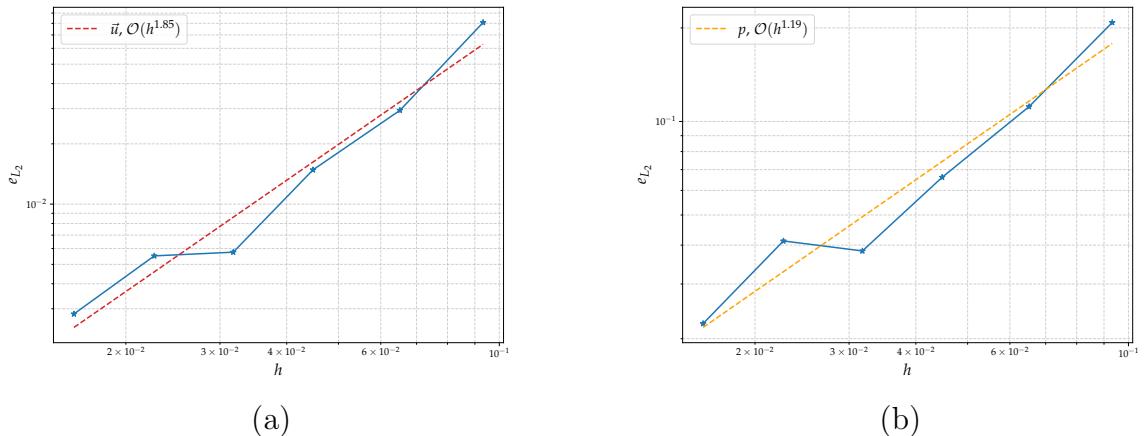


Figura 5.1: Histórico de convergência do método SIMPLE: (a) convergência da velocidade e (b) convergência da pressão.

Na Figura 5.1 são apresentados os históricos de convergência obtidos pelo método SIMPLE para os campos de velocidade e pressão. A convergência da velocidade foi avaliada a partir do módulo do vetor velocidade, definido como  $|\vec{u}| = \sqrt{u^2 + v^2}$ , de modo a representar de forma global o comportamento do campo ao longo do processo iterativo. Observa-se que, para a velocidade, a ordem de convergência obtida encontra-se próxima de 2, indicando um comportamento aproximadamente de segunda ordem, consistente com a discretização espacial empregada.

Para o campo de pressão, a ordem de convergência apresenta-se mais próxima de 1, sugerindo um comportamento de primeira ordem. Esse resultado é coerente com

o tratamento usual da pressão no algoritmo SIMPLE, no qual a correção de pressão e o acoplamento entre as variáveis tendem a limitar a ordem de convergência. Nota-se ainda que, em certo momento, ocorre um aumento do erro da pressão para uma malha mais refinada, comportamento no qual os autores não possuem uma boa hipótese de explicação.

## 5.2 Fluxo na cavidade acionado pela tampa

Nesse problema, o domínio é definido como:  $\Omega = [0, 1] \times [0, 1]$ . Detalhes sobre suas condições de contorno são dados na Figura 5.2, em que também está ilustrada a malha utilizada, composta por 8656 volumes de controle, e as condições de contorno do problema.

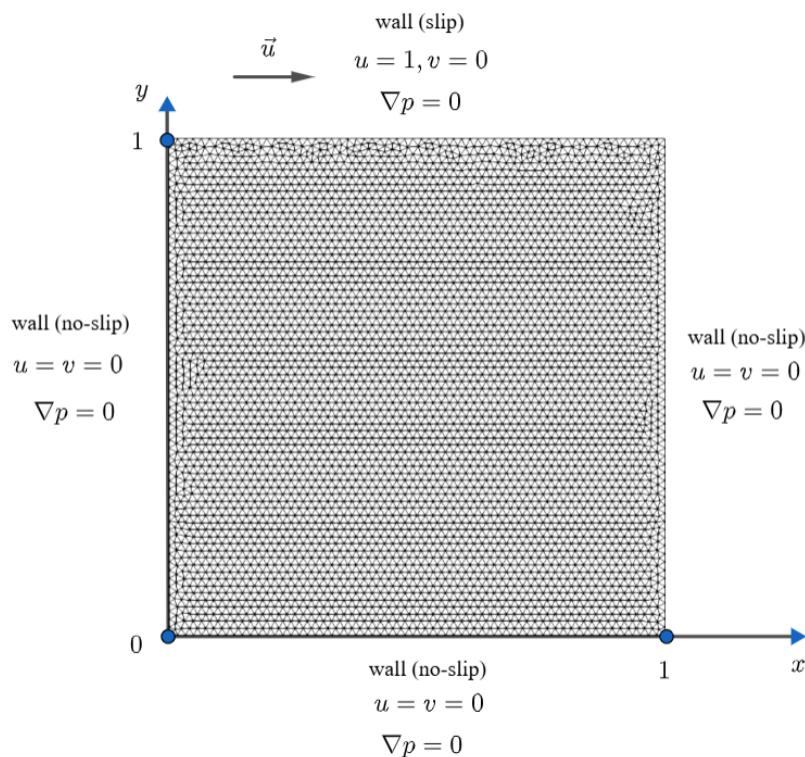


Figura 5.2: Domínio, malha e informações de contorno do problema da cavidade.

Conforme disponível na figura, três das regiões são condições de contorno *wall* em que as paredes são fixas, enquanto a parede na parte superior se movimenta, sendo daí a origem do nome do problema. A densidade nesse caso é considerada como  $\rho = 1 \text{ Kg/m}^3$ , enquanto velocidade característica será a velocidade da tampa, isto é,  $\vec{u} = 1$ . Assim, a relação de  $R_e$  depende somente da viscosidade  $\mu$ :

$$R_e = \frac{1}{\mu} \Leftrightarrow \mu = \frac{1}{R_e}. \quad (5.3)$$

O que é comum a partir de então é configurar os valores de  $R_e$ , e o valor de  $\mu$  será condicionado aos mesmos. Seguindo o que é comum à literatura, simulou-se para os valores de  $R_e = 100$  e  $R_e = 1000$ . No caso dos valores de  $\lambda_{uv}$  e  $\lambda_p$  do SIMPLE, optou-se por serem 0,6 e 0,3. As Figuras 5.3 e 5.4 mostram o campo de velocidade e as linhas de corrente para os respectivos números de Reynolds.

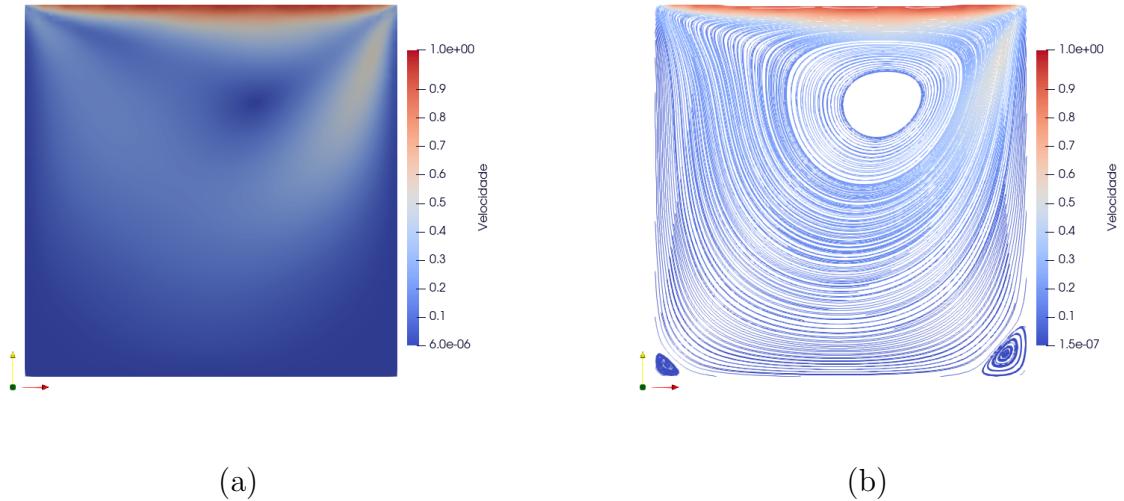


Figura 5.3: Problema da cavidade  $Re = 100$ : (a) magnitude das velocidades e (b) linhas de corrente.

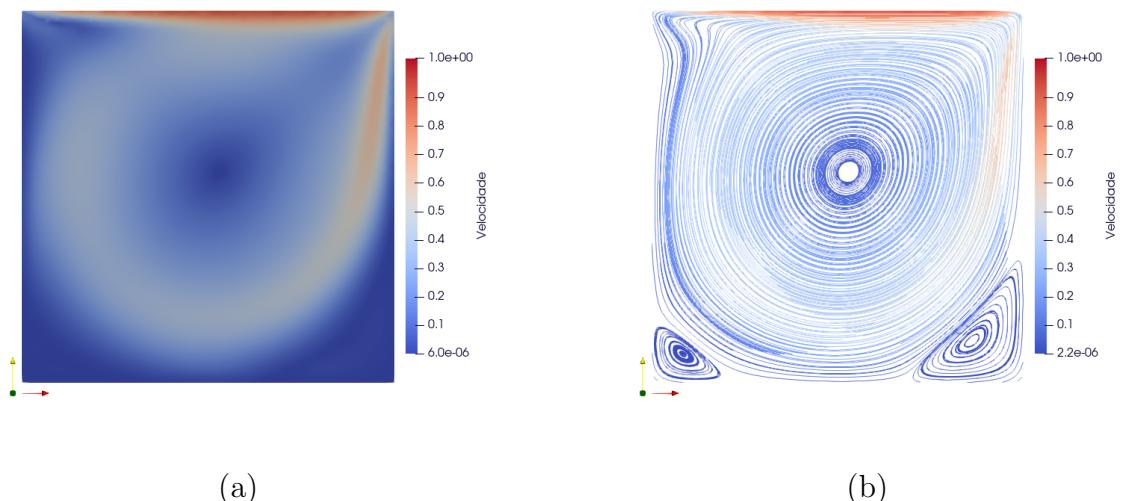


Figura 5.4: Problema da cavidade  $Re = 1000$ : (a) magnitude das velocidades e (b) linhas de corrente.

Como esse problema não possui solução exata, as soluções obtidas foram comparadas com a literatura disponível em (GHIA; GHIA; SHIN, 1982), que resolveu este problema usando uma abordagem *multigrid* com diferenças finitas. Para fazer essa comparação, verificou-se a variação da velocidade  $u$  ao longo de uma linha vertical passando pelo centro da cavidade, e a variação da velocidade  $v$  ao longo de uma linha horizontal também passando pelo centro. Os resultados estão dispostos nas Figuras 5.5 e 5.6.

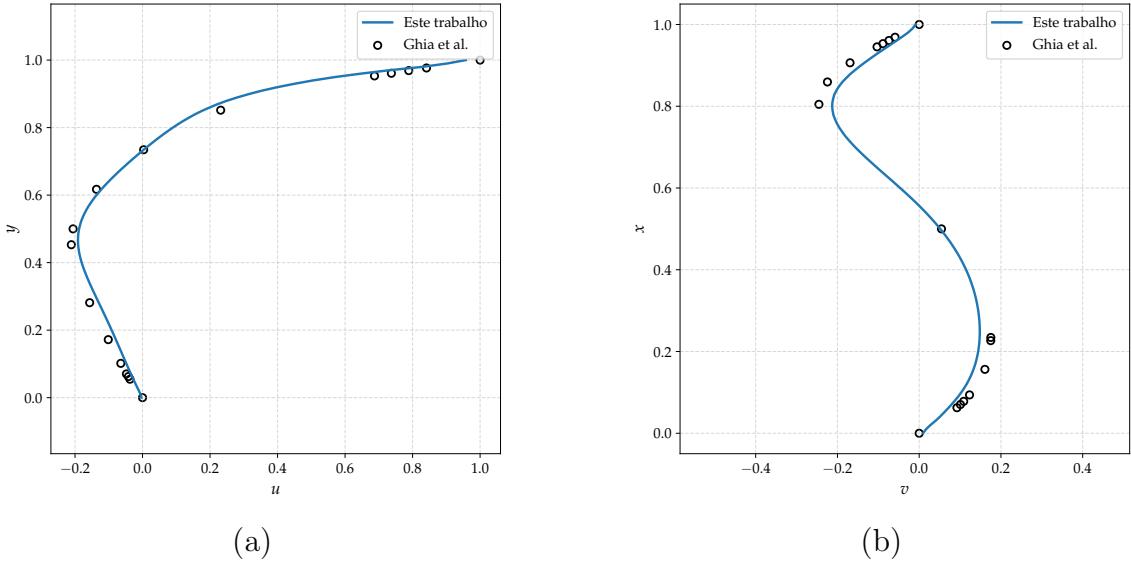


Figura 5.5: Comparaçāo com (GHIA; GHIA; SHIN, 1982) em  $Re = 100$ : (a) perfil vertical e (b) perfil horizontal.

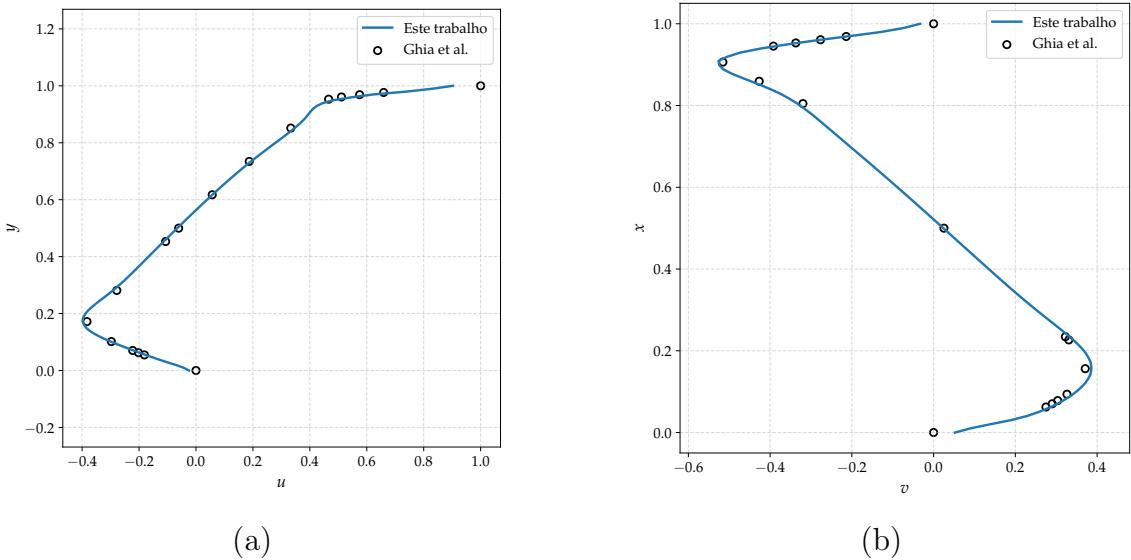


Figura 5.6: Comparaçāo com (GHIA; GHIA; SHIN, 1982) em  $Re = 1000$ : (a) perfil vertical e (b) perfil horizontal.

Por fim, é apresentado os gráficos relacionados à convergência do método SIMPLE

para ambos os Reynolds (Figura 5.7), onde se percebe que o processo converge de forma bem estável para todas as variáveis nos dois casos.

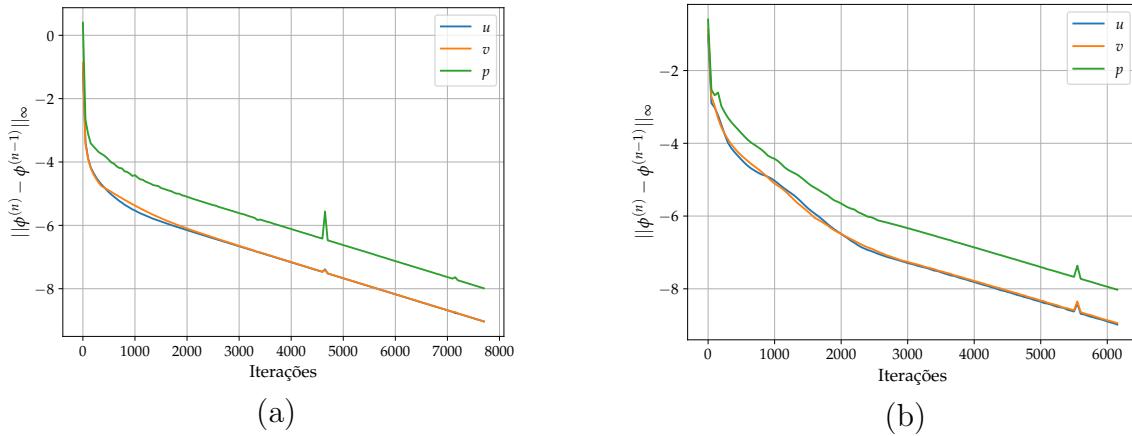


Figura 5.7: Comparaçāo da convergência para ambos os Reynolds, comparando o valor da norma do máximo e a quantidade de iterações para o problema da cavidade: (a)  $R_e = 100$  (b)  $R_e = 1000$ .

### 5.3 Problema do escoamento sobre um degrau

Nesta parte, é estudado o escoamento sobre um degrau, problema também clássico na área de CFD. O domínio é retangular, possuindo altura  $H = 1$  e comprimento  $L = 10$ . O degrau, por sua vez, representado por uma extrusão, possui altura  $h = 0.5$  e comprimento  $l = 2$ . Um importante valor relacionado a esse problema é sua taxa de expansão:  $H/h = 2.0$ , usada para poder comparar soluções da literatura. A Figura 5.8 ilustra maiores detalhes sobre as condições de contorno e malha utilizados. Dessa vez, a malha possui 8450 células triangulares.

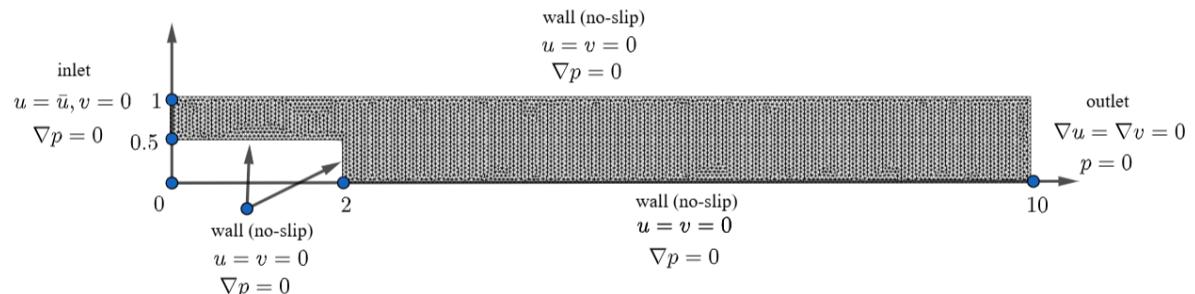


Figura 5.8: Domínio, malha e informações de contorno do problema do degrau.

Conforme definido na formulação do problema, as superfícies superior e inferior do

canal, bem como as superfícies do degrau, são modeladas como paredes com condição de não deslizamento (*no-slip*). Para representar o escoamento entrando no domínio, adota-se na fronteira lateral esquerda uma condição de contorno do tipo *inlet*. O perfil de velocidade prescrito é dado por

$$\begin{cases} \bar{u}(x, y) = -16(y - 0.75)^2 + 1, & \text{se } y > 0.5, \\ 0, & \text{caso contrário.} \end{cases} \quad (5.4)$$

Na fronteira lateral direita, impõe-se uma condição de saída (*outlet*), permitindo a saída do fluido do domínio.

O número de Reynolds é definido a partir de uma densidade  $\rho = 1,0 \text{ kg/m}^3$ , de um comprimento característico  $L = 0,5$  (altura do degrau) e de uma velocidade característica  $U = 1,0 \text{ m/s}$ , correspondente à velocidade máxima do perfil  $\bar{u}$ . Dessa forma,

$$R_e = \frac{\rho U L}{\mu} = \frac{0.5}{\mu}, \quad (5.5)$$

de modo que a viscosidade dinâmica  $\mu$  pode ser especificada em função do valor de  $R_e$ . Nessa simulação, foram considerados os casos  $R_e = 10$ ,  $R_e = 50$  e  $R_e = 100$ .

Para as simulações, adotaram-se coeficientes de relaxação  $\lambda_{uv} = 0.3$  e  $\lambda_p = 0.2$ , valores relativamente baixos em razão de dificuldades de convergência observadas. Essas dificuldades podem ser visualizadas na Figura 5.9, em que se notam oscilações marcantes ao longo do processo iterativo: os resíduos das equações aumentam e diminuem repetidamente, indicando um comportamento mais sensível do método para esse caso.

Nas Figuras 5.10, 5.11 e 5.12 são apresentados os campos de escoamento obtidos para os diferentes números de Reynolds. Observa-se, em todos os casos, a formação de uma região de recirculação imediatamente a jusante do degrau. Nota-se ainda que o comprimento dessa zona de recirculação aumenta progressivamente com o aumento do número de Reynolds, evidenciando a maior influência dos efeitos inerciais no escoamento.

Na Figura 5.13 encontra-se uma comparação do perfil da componente  $u$  da velocidade em diferentes cortes no eixo  $x$ , comparado com dados experimentais retirados de (ARMALY et al., 1983). Para essa comparação, os resultados numéricos foram dimensi-

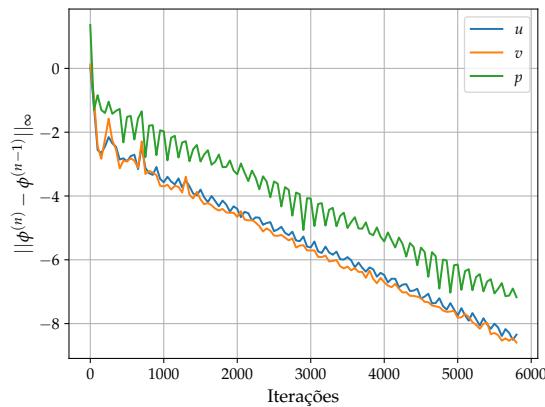


Figura 5.9: Comparaçāo da convergēcia para  $R_e = 100$ , comparando o valor da norma do māximo e a quantidade de iterações para o problema do degrau.

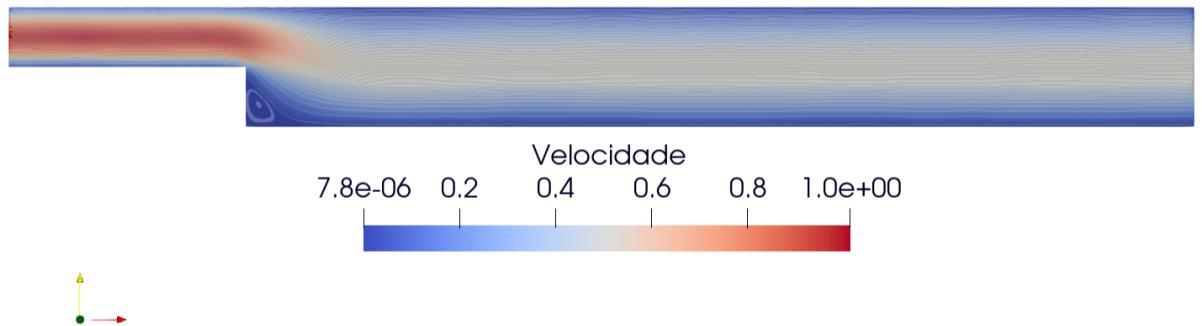


Figura 5.10: Magnitude da velocidade e linhas de corrente do problema do degrau para  $R_e = 10$ .

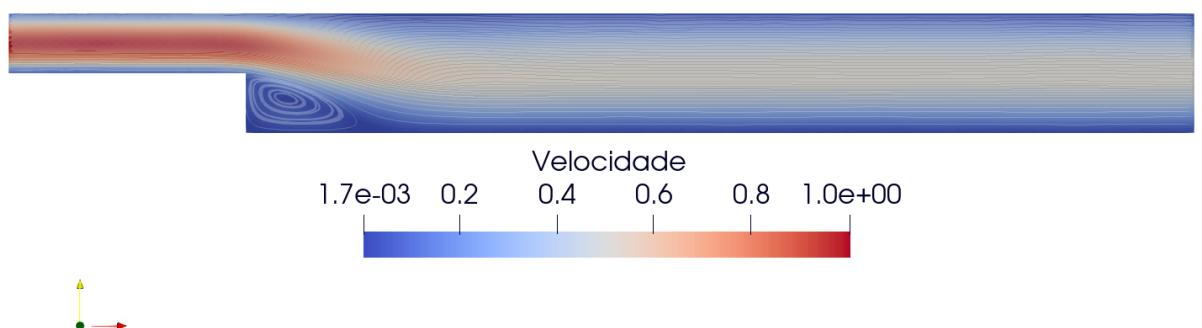


Figura 5.11: Magnitude da velocidade e linhas de corrente do problema do degrau para  $R_e = 50$ .

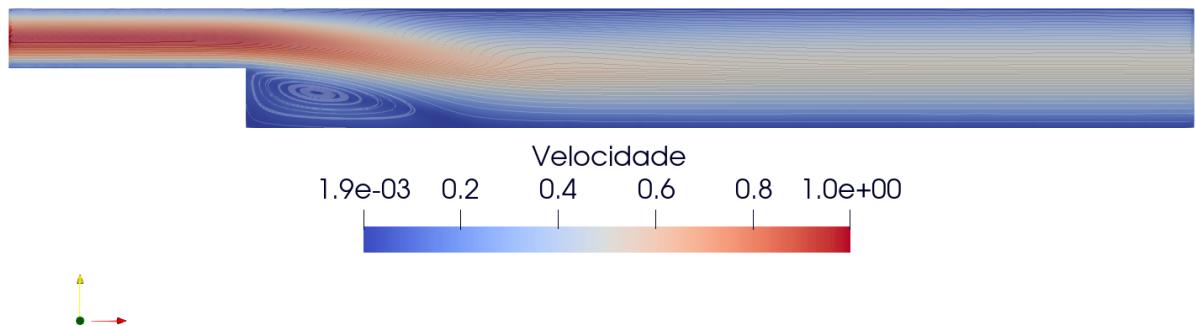


Figura 5.12: Magnitude da velocidade e linhas de corrente do problema do degrau para  $R_e = 100$ .

onalizados conforme os dados disponibilizados na referência, multiplicando os eixos  $x$  e  $y$  por 30 e 10. Os resultados evidenciam uma tendência similar na forma das curvas, mas os pontos em si encontram-se distantes, indicando que talvez o refinamento da solução ainda não tenha conseguido captar suficientemente bem o padrão observado em laboratório. Como um maior refinamento seria impeditivo computacionalmente, os autores optaram por explorar melhor esse ponto em trabalhos futuros.

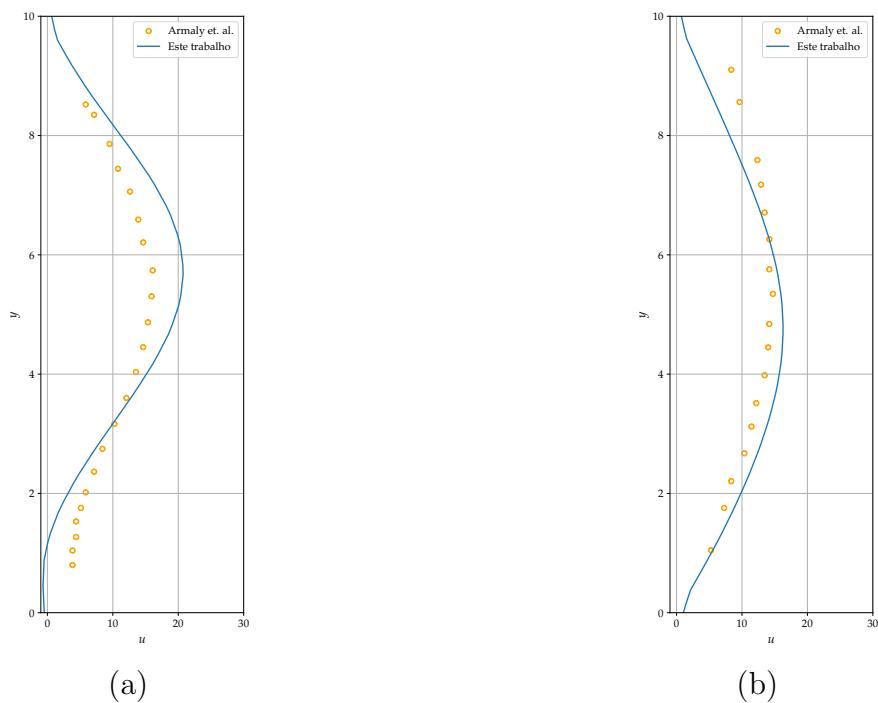


Figura 5.13: Comparaçāo da qualidade da solução comparando com (ARMALY et al., 1983) em  $R_e = 100$ : (a)  $x/h = 3.06$  (b)  $x/h = 6.12$ .

## 5.4 Problema do fluxo ao redor de um cilindro

O último problema utilizado para validação neste trabalho é o problema do escoamento de um fluido ao redor de um cilindro. Esse é um problema também clássico da literatura de CFD (BAI; LI, 2011). Como referência principal, seguiu-se nesse trabalho a abordagem descrita por (SCHÄFER et al., 1996).

Conforme mostrado na Figura 5.14, o domínio do problema é  $\Omega = [0, 2.2] \times [0, 0.41]$ , em que a malha possui 8382 células. No interior desse domínio há uma região sólida circular, que no caso bidimensional representa a seção de um cilindro tridimensional. O círculo ilustrado na figura possui centro em  $(0.2, 0.2)$  e raio  $r = 0.05$ .

As condições de contorno são semelhantes às do problema do degrau, excetuando-se a condição de entrada (*inlet*), definida neste caso por

$$\bar{u}(x, y) = \frac{4 \cdot 0.3 y (0.41 - y)}{0.41^2}. \quad (5.6)$$

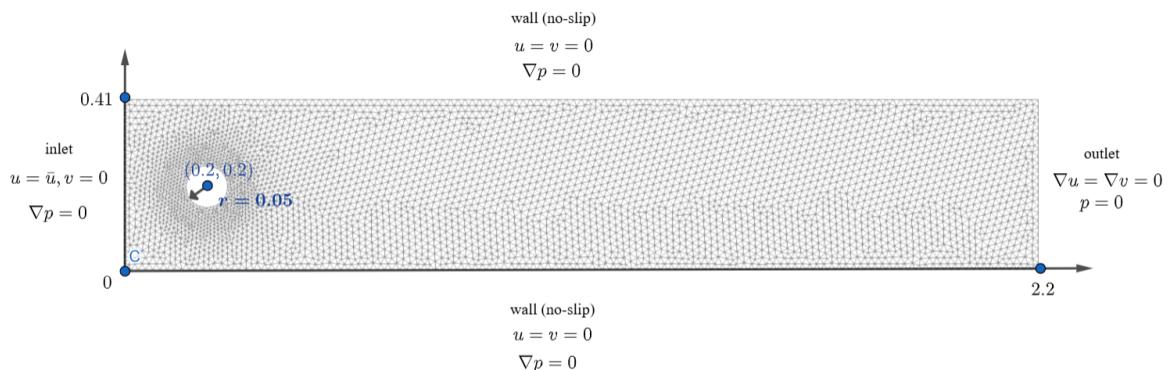


Figura 5.14: Domínio, malha e informações de contorno do problema do cilindro.

Além disso, impõe-se uma condição de contorno do tipo *wall* fixa ao longo da superfície do cilindro. A malha empregada nesse caso, diferentemente dos exemplos anteriores, foi refinada adaptativamente, de modo a concentrar um número maior de volumes de controle nas proximidades do cilindro, região de maior interesse por apresentar as variações mais intensas das grandezas físicas.

Para o algoritmo SIMPLE, adotaram-se os coeficientes de sub-relaxação  $\lambda_{uv} = 0,3$  e  $\lambda_p = 0,2$ . Neste problema, para obter um número de Reynolds igual a  $Re = 20$ , considerando densidade unitária, foi suficiente adotar uma viscosidade dinâmica  $\mu =$

$1 \times 10^{-3}$ .

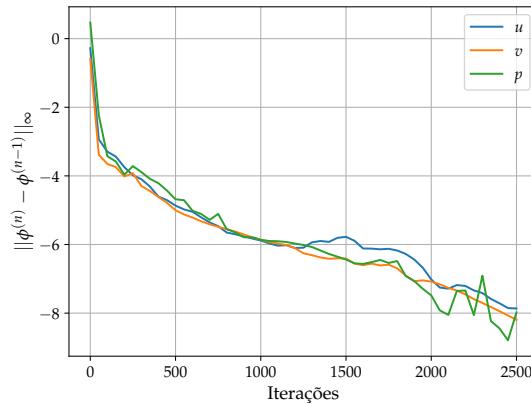


Figura 5.15: Comparação da convergência para  $R_e = 20$ , comparando o valor da norma do máximo e a quantidade de iterações para o problema do cilindro.

A figura 5.15 ilustra a convergência também para o caso do cilindro, e seu gráfico é um pouco mais estável que no caso do degrau, mas menos estável que no caso da cavidade. A solução do problema está exibida na figura 5.16, demonstrando como o fluido que sai da borda esquerda, atinge o obstáculo que seria o cilindro, e com isso adquire uma maior velocidade.

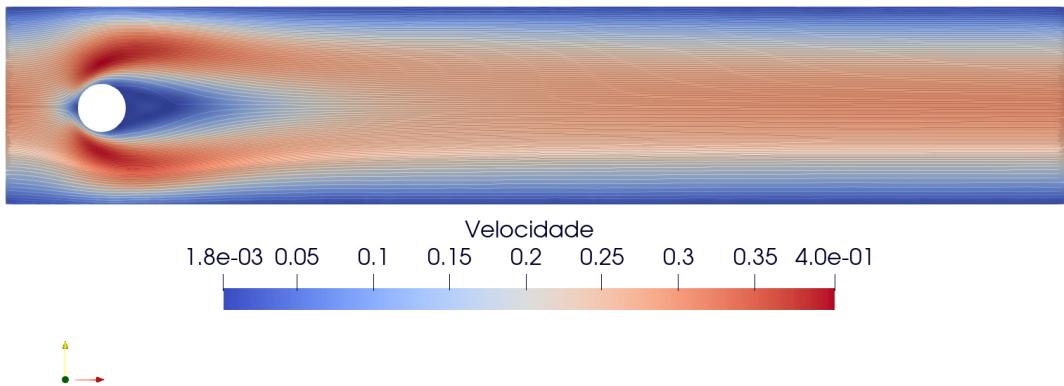


Figura 5.16: Magnitude da velocidade e linhas de corrente do *flow over a cylinder* para  $R_e = 20$ .

Por fim, para uma comparação quantitativa da qualidade da solução numérica, são analisados o comprimento da zona de recirculação a jusante do cilindro e a diferença de pressão. O comprimento da zona de recirculação,  $L_a$ , é definido como:  $L_a = x_r - x_e$ ,

em que  $x_e = 0.25$  corresponde à coordenada  $x$  da extremidade a jusante do cilindro, e  $x_r$  representa a coordenada  $x$  associada ao ponto mais extremo da zona de recirculação do escoamento. Por sua vez, a diferença de pressão,  $\Delta p$ , é calculada a partir da diferença entre as pressões medidas a montante e a jusante do cilindro, ao longo da linha central do escoamento, sendo dada por  $\Delta p = p(0.15, 0.2) - p(0.25, 0.2)$ .

	$L_a$	$\Delta p$
Limite Inferior	0,0842	0,1172
Limite Superior	0,0852	0,1176
Este Trabalho	0,095	0,1126

Tabela 5.1: Comparação entre os limites inferior e máximo de diversos trabalhos reportados em (SCHÄFER et al., 1996) e a solução obtida neste trabalho.

Os valores obtidos para esses parâmetros estão apresentados na Tabela 5.1. Nela, os valores para  $L_a$  e  $\Delta p$  encontram-se próximos do esperado, sendo o valor de  $L_a$  com um desvio de 11,5% e o  $\Delta p$  com um desvio de 4%, indicando uma boa acurácia para o código desenvolvido.

## 6 CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho, foi desenvolvido um solucionador baseado no MVF, implementado em C++, para a simulação de escoamentos de fluidos bidimensionais. O código encontra-se disponível no repositório <https://github.com/ricardo-ervilha/ReFum2D>. Inicialmente, foram apresentados os fundamentos matemáticos do MVF, bem como a estratégia de discretização adotada para a resolução das equações governantes em malhas não estruturadas. Em seguida, discutiram-se os aspectos computacionais do método, descrevendo-se de forma detalhada o fluxo de execução do *software*, desde a leitura dos dados de entrada até a aplicação do algoritmo SIMPLE e a exportação dos resultados em formato compatível com o *ParaView*.

A partir da análise dos resultados obtidos nos *benchmarks* comparativos, verificou-se que o código apresenta segunda ordem de convergência para o campo de velocidades e primeira ordem para a pressão, sendo capaz de reproduzir de forma satisfatória os principais comportamentos físicos observados em problemas clássicos da literatura. Esses resultados evidenciam a robustez da implementação e indicam que o solucionador desenvolvido constitui uma base para futuras aplicações em problemas de engenharia e pesquisa científica.

Como perspectivas de trabalhos futuros, torna-se relevante investigar inicialmente estratégias de paralelização visando à redução do custo computacional associado à resolução dos sistemas lineares e à natureza iterativa do algoritmo SIMPLE, ou ainda técnicas de reordenação nodal que permitam a obtenção de sistemas matriciais com estrutura de banda, possibilitando resolver os sistemas de forma mais eficiente. Nesse contexto, a utilização de bibliotecas mais robustas, como a PETSc, configura-se como uma alternativa promissora para a exploração eficiente do paralelismo. Posteriormente, pretende-se estender o código para a simulação de problemas tridimensionais, uma vez que a maioria das aplicações de interesse prático envolve escoamentos em três dimensões. Por fim,

destaca-se a possibilidade futura de incorporar modelos de turbulência, ampliando a aplicabilidade do *software* a cenários físicos mais complexos.

## Bibliografia

- AHRENS, J.; GEVECI, B.; LAW, C. Paraview: An end-user tool for large data visualization. In: *Visualization Handbook*. [S.l.]: Elsevier, 2005. ISBN 9780123875822. Software: ParaView, <https://www.paraview.org/>.
- ANDERSON, J. *Computational Fluid Dynamics: The Basics with Applications*. McGraw-Hill, 1995. (McGraw-Hill International Editions: Mechanical Engineering). ISBN 9780071132107. Disponível em: <[https://books.google.com.br/books?id=phG\\_QgAACAAJ](https://books.google.com.br/books?id=phG_QgAACAAJ)>.
- ANSYS. *ANSYS Fluent - CFD Software — ANSYS*. 2016. Disponível em: <<http://www.ansys.com/products/fluids/ansys-fluent>>.
- ARMALY, B. F. et al. Experimental and theoretical investigation of backward-facing step flow. *Journal of Fluid Mechanics*, v. 127, p. 473–496, 1983.
- BAI, H.; LI, J. Numerical simulation of flow over a circular cylinder at low reynolds number. *Advanced Materials Research*, v. 255-260, p. 942–946, 05 2011.
- CHEN, X. et al. Some continuous and discontinuous galerkin methods and structure preservation for incompressible flows. *International Journal for Numerical Methods in Fluids*, v. 93, n. 7, p. 2155–2174, 2021. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.4969>>.
- CHÉNIER, E.; EYMARD, R.; TOUAZI, O. Numerical results using a colocated finite-volume scheme on unstructured grids for incompressible fluid flows. *Numerical Heat Transfer, Part B: Fundamentals*, Taylor & Francis, v. 49, n. 3, p. 259–276, 2006. Disponível em: <<https://www.tandfonline.com/doi/abs/10.1080/10407790500292093>>.
- DING, P.; SUN, D. L. A pressure-based segregated solver for incompressible flow on unstructured grids. *Numerical Heat Transfer, Part B: Fundamentals*, Taylor & Francis, v. 64, n. 6, p. 460–479, 2013. Disponível em: <<https://doi.org/10.1080/10407790.2013.831682>>.
- GEUZAIN, C.; REMACLE, J.-F. Gmsh: A 3-d finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, v. 79, n. 11, p. 1309–1331, 2009. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.2579>>.
- GHIA, U.; GHIA, K.; SHIN, C. High-re solutions for incompressible flow using the navier-stokes equations and a multigrid method. *Journal of Computational Physics*, v. 48, n. 3, p. 387–411, 1982. ISSN 0021-9991. Disponível em: <<https://www.sciencedirect.com/science/article/pii/0021999182900584>>.
- GUENNEBAUD, G.; JACOB, B. Eigen v3. 01 2010.
- HARLOW, F. H.; WELCH, J. E. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of Fluids*, v. 8, n. 12, p. 2182–2189, 1965.

- HASELBACHER, A. C. A grid-transparent numerical method for compressible viscous flows on mixed unstructured grids. 1 1999. Disponível em: <[https://repository.lboro.ac.uk/articles/thesis/A\\_grid-transparent\\_numerical\\_method\\_for\\_compressible\\_viscous\\_flows\\_on\\_mixed\\_unstructured\\_grids/9219554](https://repository.lboro.ac.uk/articles/thesis/A_grid-transparent_numerical_method_for_compressible_viscous_flows_on_mixed_unstructured_grids/9219554)>.
- JASAK, H. Error analysis and estimation for the finite volume method with applications to fluid flows. *Direct*, M, 01 1996.
- Kitware. *VTK File Formats*. 2026. <[https://docs.vtk.org/en/latest/vtk\\_file\\_formats/index.html](https://docs.vtk.org/en/latest/vtk_file_formats/index.html)>. Acessado em 07/01/2026.
- KOVASZNAY, L. I. G. Laminar flow behind a two-dimensional grid. *Mathematical Proceedings of the Cambridge Philosophical Society*, v. 44, n. 1, p. 58–62, 1948.
- MALISKA, C. *Transferência de calor e mecânica dos fluidos computacional*. Livros Técnicos e Científicos, 2004. ISBN 9788521613961. Disponível em: <<https://books.google.com.br/books?id=8i3qAAAACAAJ>>
- MAZUMDER, S. *Numerical methods for partial differential equations: Finite difference and finite volume methods*. [S.l.: s.n.], 2015. 1-461 p.
- MOUKALLED, F.; MANGANI, L.; DARWISH, M. *The Finite Volume Method in Computational Fluid Dynamics: An Advanced Introduction with OpenFOAM and Matlab*. 1st. ed. [S.l.]: Springer Publishing Company, Incorporated, 2015. ISBN 3319168738.
- PATANKAR, S.; SPALDING, D. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *International Journal of Heat and Mass Transfer*, v. 15, n. 10, p. 1787–1806, 1972. ISSN 0017-9310. Disponível em: <<https://www.sciencedirect.com/science/article/pii/0017931072900543>>.
- PIETRO, D. D.; ERN, A. *Mathematical Aspects of Discontinuous Galerkin Methods*. Springer Berlin Heidelberg, 2011. (Mathématiques et Applications). ISBN 9783642229800. Disponível em: <<https://books.google.com.br/books?id=ak-qQvWGA5oC>>.
- RHIE, C. M.; CHOW, W. L. Numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA Journal*, v. 21, n. 11, p. 1525–1532, 1983. Disponível em: <<https://doi.org/10.2514/3.8284>>.
- SCHÄFER, M. et al. Benchmark computations of laminar flow around a cylinder. In: . [s.n.], 1996. Disponível em: <<https://api.semanticscholar.org/CorpusID:54080193>>.
- SILVA, G. M. F. Trabalho de Conclusão de Curso (Graduação em Engenharia Computacional), *MESHTOOLS: Um solucionador de elementos finitos e manipulador de malhas em C++*. Juiz de Fora, MG, Brasil: [s.n.], 2023. 74 p.
- VASCONCELLOS, J. F. V. d. *Um método de volumes finitos usando malhas não-estruturadas para o estudo de escoamentos com frentes livres*. Tese (Tese (Doutorado)) — Universidade Federal de Santa Catarina, Florianópolis, SC, 1999. Disponível em: <<http://repositorio.ufsc.br/xmlui/handle/123456789/81010>>.
- VERSTEEG, H.; MALALASEKERA, W. *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*. Pearson Education Limited, 2007. ISBN 9780131274983. Disponível em: <<https://books.google.com.br/books?id=RvBZ-UMpGzIC>>.

WARMING, R. F.; BEAM, R. M. Upwind second-order difference schemes and applications in aerodynamic flows. *AIAA Journal*, v. 14, n. 9, p. 1241–1249, 1976. Disponível em: [⟨https://doi.org/10.2514/3.61457⟩](https://doi.org/10.2514/3.61457).

WELLER, H. G. et al. A tensorial approach to computational continuum mechanics using object-oriented techniques. *Computers in Physics*, v. 12, n. 6, p. 620–631, 1998.