Estudo de métodos de interface imersa para as equações de Navier-Stokes

Gabriela Aparecida dos Reis

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Gabriela Aparecida dos Reis

Estudo de métodos de interface imersa para as equações de Navier-Stokes

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, como parte dos requisitos para obtenção do título de Doutora em Ciências - Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientador: Prof. Dr. José Alberto Cuminato

Coorientador: Prof. Dr. Leandro Franco de Souza

USP – São Carlos
Junho de 2016

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados fornecidos pelo(a) autor(a)

d375e dos Reis, Gabriela Aparecida
Estudo de métodos de interface imersa para as
equações de Navier-Stokes / Gabriela Aparecida dos
Reis; orientador José Alberto Cuminato; co-
orientador Leandro Franco de Souza. -- São Carlos,
2016.
134 p.

Tese (Doutorado - Programa de Pós-Graduação em
Ciências de Computação e Matemática Computacional) --
Instituto de Ciências Matemáticas e de Computação,
Universidade de São Paulo, 2016.

1. métodos de projeção. 2. métodos de interface
imersa. 3. diferenças finitas compactas. 4. métodos
de alta ordem. 5. equações de Navier-Stokes. I.
Cuminato, José Alberto, orient. II. de Souza,
Leandro Franco, co-orient. III. Título.

Gabriela Aparecida dos Reis

Study of immersed interface methods for the Navier-Stokes equations

Doctoral dissertation submitted to the Instituto de Ciências Matemáticas e de Computação - ICMC-USP, in partial fulfillment of the requirements for the degree of the Doctorate Program in Computer Science and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Prof. Dr. José Alberto Cuminato

Coadvisor: Prof. Dr. Leandro Franco de Souza

USP – São Carlos
June 2016

“Talvez não tenha conseguido fazer o melhor, mas lutei para que o melhor fosse feito. Não sou o que deveria ser, mas Graças a Deus, não sou o que era antes.”

— MARTIN LUTHER KING

*A meu pai, Sebastiao, minha mãe, Anerli,
meu irmao, Vinicius, e meu namorado, Italo.*

Agradecimentos

A Deus pelo dom da vida e a oportunidade de estar terminando mais essa etapa.

À minha família pelo apoio.

Aos meus amigos pela compreensão nos dias fáceis e nos difíceis.

Aos professores que conheci durante a minha trajetória acadêmica.

À FAPESP pelo apoio financeiro.

Resumo

Uma grande limitação dos métodos de diferenças finitas é que eles estão restritos a malhas e domínios retangulares. Para descrever escoamentos em domínios complexos, como, por exemplo, problemas com superfícies livres, faz-se necessário o uso de técnicas acessórias. O método de interfaces imersas é uma dessas técnicas.

Nesse trabalho, primeiramente foi desenvolvido um método de projeção, totalmente livre de pressão, para as equações de Navier-Stokes com variáveis primitivas em malha deslocada. Esse método é baseado em diferenças finitas compactas, possuindo segunda ordem temporal e quarta ordem espacial.

Esse método foi combinado com o método de interface imersa de Linnick e Fasel [2] para resolver numericamente as equações de Stokes com quarta ordem de precisão. A verificação do código foi feita por meio do método das soluções manufaturadas e da comparação com resultados de outros autores em problemas clássicos da literatura.

Palavras-chave: métodos de projeção, métodos de interface imersa, diferenças finitas compactas, métodos de alta ordem, equações de Navier-Stokes, equações de Stokes.

Abstract

A great limitation of finite differences methods is that they are restricted to rectangular meshes and domains. In order to describe flows in complex domains, e.g. free surface problems, it is necessary to use accessory techniques. The immersed interface method is one of such techniques.

In the present work, firstly, a projection method was developed, which is completely pressure-free, for the Navier-Stokes equations with primitive variables in a staggered mesh. This method is based on compact finite differences, with temporal second-order precision and spatial fourth-order precision.

This method was combined with the immersed interface method from Linnick e Fasel [2] in order to numerically solve the Stokes equations with fourth-order precision. The verification of the code was performed with the manufactured solutions method and by comparing results with other authors for some classical problems in the literature.

Keywords: projection methods, immersed interface methods, compact finite differences, high-order methods, Navier-Stokes equations, Stokes equations.

Sumário

1	Introdução	27
2	Equações de Navier-Stokes	33
2.1	Discretização temporal	34
2.2	Discretização espacial	35
2.2.1	Aproximações por diferenças finitas	35
2.2.1.1	Aproximação para a primeira derivada	36
2.2.1.2	Aproximação para a segunda derivada	36
2.2.1.3	Aproximação para a equação de Poisson	36
2.2.1.4	Condições de contorno	37
2.2.2	Aproximações por diferenças finitas compactas	37
2.2.2.1	Aproximação para o termo viscoso usando diferenças compactas .	38
2.2.2.2	Aproximação para o termo convectivo usando diferenças compactas	38
2.2.2.3	Aproximação para o gradiente de pressão usando diferenças compactas	39
3	Métodos de projeção	41
3.1	Método de projeção com pressão incremental e livre de pressão	42
3.2	Estêncil compatível: projeção exata e projeção aproximada	43
3.2.1	Projeção aproximada em 2D	44
3.2.2	Projeção exata em 2D	46
3.3	Condição de compatibilidade	47
3.4	Resultados	48
3.4.1	Método das soluções manufaturadas	48
3.4.2	Escoamento forçado em uma cavidade	50
3.4.3	Vórtices de Taylor-Green	51
3.4.4	Escoamento em uma cavidade	53
3.4.5	Escoamento em um degrau	55

4	Métodos acoplados e discretização temporal	59
4.1	Estabilidade	60
4.2	Análise da estabilidade	61
4.2.1	Euler explícito	61
4.2.2	Euler implícito	61
4.2.3	Crank-Nicolson	63
4.3	Additive Runge-Kutta - ARK	63
4.3.1	Passo de tempo adaptativo	67
4.3.2	Implementação	67
4.3.3	Resolução do sistema não-linear	67
4.4	Resultados	68
5	Método de fronteira imersa para as Equações de Navier-Stokes	71
5.1	Condição de compatibilidade	73
5.2	<i>Outflow</i> continuativo	74
5.3	Resultados	74
5.3.1	Verificação com solução manufaturada	74
5.3.2	Escoamento ao redor de um círculo	76
6	Método de interface imersa explícito para o cálculo de derivadas	83
6.1	Série de Taylor corrigida	84
6.2	Correção para estênceis de diferenças finitas	86
6.3	MII em malha deslocada	88
6.4	Resultados	89
7	Método de interface imersa implícito para a Equação de Poisson	95
7.1	Equação de Poisson 1D	95
7.2	Equação de Poisson 2D	96
7.2.1	Condições de contorno do tipo Neumann	97
7.3	Resultados	98
7.3.1	Equação de Poisson 1D	98
7.3.2	Equação de Poisson 2D	100
7.3.3	Tempo de processamento	101
7.3.4	Condições de contorno do tipo Dirichlet e Neumann	101
8	Método de interface imersa para as Equações de Stokes	105
8.1	Equação do momento	105
8.2	Equação da continuidade	105
8.3	Resultados	106

8.3.1	Exemplo I	106
8.3.2	Exemplo II	106
8.3.3	Exemplo III	107
9	Conclusões	109
9.1	Artigos publicados em periódicos	110
9.2	Artigos publicados em conferências	110
9.3	Trabalhos futuros	110
A	Métodos de diferenças finitas compactas	113
A.1	Desenvolvimento dos esquemas de diferenças finitas compactas	114
A.1.1	Aproximações compactas para a primeira derivada	115
A.1.2	Aproximações compactas para a segunda derivada	116
A.1.3	Análise de ordem	116
A.2	Análise de Fourier dos erros	118
A.2.1	Estêncil colocado para a primeira derivada	119
A.2.2	Estêncil deslocado para a primeira derivada	120
A.2.3	Estêncil colocado uniforme para a segunda derivada	122
A.2.4	Estêncil colocado não uniforme para a segunda derivada	122
B	Filtro	125
	Referências Bibliográficas	127

Lista de Figuras

2.1	Ilustração de uma malha deslocada e sua representação computacional.	36
2.2	Ilustração do estêncil próximo ao contorno.	37
3.1	Teste de convergência espacial baseado na solução manufaturada para o problema do escoamento forçado em uma cavidade. A simulação foi desenvolvida até o estado estacionário ser atingido. O erro é calculado como a diferença entre a solução numérica e a solução manufaturada na norma L_∞ . A linha contínua representa quarta ordem de convergência. A viscosidade cinemática usada é $\nu = 10^{-2}$, o passo de tempo é $\Delta t = 10^{-4}$ e as malhas foram 8×8 , 16×16 , 32×32 , 64×64 e 128×128 . $\Delta x = \Delta y = h$	52
3.2	Evolução do máximo do divergente discreto numa malha fixa 8×8 com $\Delta t = 10^{-4}$ e $\nu = 10^{-2}$	53
3.3	Convergência temporal para o problema vórtices de Taylor-Green. A simulação foi executada até $t = 1$. A viscosidade cinemática usada é $\nu = 10^{-2}$. Os erros são calculados na norma L_∞ . Os erros são calculados em relação à solução de referência com $\Delta t = 10^{-4}$. A malha é fixada em 20×20 . Em (a) e (c) a linha contínua representa segunda ordem enquanto que em (b) primeira ordem.	54
3.4	Perfis de velocidade no estado estacionário para o problema da cavidade: corte do perfil de velocidade vertical v em $y = 0.5$; corte do perfil de velocidade horizontal u em $x = 0.5$. Viscosidade cinemática $\nu = 10^{-2}$, $\nu = 2.5 \times 10^{-3}$ e $\nu = 10^{-3}$, respectivamente, em uma malha 130×130	55
3.5	Contornos da função corrente e da pressão para o problema da cavidade. Malha 130×130 e $\nu = 10^{-3}$	56
3.6	Escoamento sobre um degrau: geometria.	56
3.7	Contornos da função corrente, da componente u da velocidade e da pressão. Malha 300×50 , $\nu = 5 \times 10^{-3}$	57
4.1	Região de estabilidade do método de Euler explícito.	62
4.2	Região de estabilidade do método de Euler implícito.	62

4.3	Região de estabilidade do método Crank-Nicolson.	63
4.4	Diagrama das classes de métodos de Runge-Kutta: IRK= <i>implicit</i> ; DIRK= <i>diagonally implicit</i> ; SIRD= <i>singly implicit</i> ; SDIRK= <i>singly diagonally implicit</i> ; ERK= <i>explicit</i> . Figura extraída de [1].	66
4.5	Evolução do passo de tempo adaptativo durante a simulação da solução manufaturada (4.26) usando o método ARK3(2)4L[2]SA.	69
5.1	Ilustração de um domínio com interface imersa. Células sólidas são marcadas com S, e o domínio de fluido, com F.	71
5.2	Direção normal.	72
5.3	Direção tangencial.	73
5.4	Domínios para a simulação da solução manufaturada 5.8.	75
5.5	Evolução do passo de tempo adaptativo durante a simulação da solução manufaturada (5.8) (domínio sem interface) usando o método ARK3(2)4L[2]SA.	76
5.6	Campo de velocidades na simulação da solução manufaturada (5.8) em $t = 100s$. $\nu = 10^{-2}$	77
5.7	Valor do divergente na simulação da solução manufaturada (5.8), onde o quadrado que delimita o domínio é formado por uma interface imersa em $t = 100s$. $\nu = 10^{-2}$. Malha 100×100	78
5.8	Escoamento ao redor de um círculo: geometria.	79
5.9	Passo de tempo adaptativo na simulação do escoamento ao redor de um círculo. $\nu = 0.25 \times 10^{-3}$	80
5.10	Perfil de velocidade em $t = 20s$ na simulação do escoamento ao redor de um círculo. $\nu = 0.25 \times 10^{-3}$	81
6.1	Ilustração do domínio contendo uma interface imersa $\partial\Omega_i$, extraído de [2].	83
6.2	Função $f(x)$ com uma descontinuidade no ponto $x = x_\alpha$	84
6.3	Casos possíveis onde a interface intersecta a malha. Em (a) tanto o valor da função (em $x_{i+1/2}$) quanto a derivada (em x_{i+1}) estão depois da descontinuidade. Em (b) apenas a derivada (em x_{i+1}) está depois da descontinuidade.	88
6.4	Em (a) a borda do domínio coincide com a malha computacional. Nesse caso, usam-se estênceis não centrais próximo à borda. Em (b) a borda do domínio intersecta as células computacionais. Nesse caso, usa-se o método de interface imersa.	89
6.5	Comparação da ordem de convergência do MII com estênceis não centrais.	90
6.6	Comparação da ordem de convergência do MII com estênceis não centrais.	91
6.7	Comparação da ordem de convergência do MII com estênceis não centrais.	92
6.8	Comparação da ordem de convergência do MII com estênceis não centrais.	93

7.1	Estêncil interceptando uma interface imersa, extraído de [2]. Na figura, $\times$ representam os pontos do estêncil de diferenças finitas compactas, $\square$ os pontos internos ao domínio usados para o cálculo das correções e $\circ$ os pontos sobre a interface imersa, também usados no cálculo da correção.	98
7.2	Ilustração do erro na aproximação da equação de Poisson. Malha 100×100	103
8.1	Posições da interface.	106
A.1	Estênceis centrais em malhas colocalizadas (a) e deslocadas (b). $f_i = f(x_i)$, $F_i^{(k)} \approx \frac{\partial^k f}{\partial x^k} \Big _{x_i}$, $k = \{1, 2\}$ e h é o espaçamento da malha.	114
A.2	Estênceis para malhas uniforme (a), não uniforme (b) e deslocada (c) próximo ao contorno. $f_i = f(x_i)$, $F_i^{(k)} \approx \frac{\partial^k f}{\partial x^k} \Big _{x_i}$, $k = \{1, 2\}$	114
A.3	Análise de Fourier da primeira derivada para esquemas em malha colocada uniforme.	120
A.4	Análise de Fourier da primeira derivada para esquemas em malha deslocada.	122
A.5	Análise de Fourier da segunda derivada para esquemas em malha colocada uniforme.	123
A.6	Análise de Fourier da segunda derivada para esquemas em malha colocada não uniforme.	124

Lista de Tabelas

3.1	Comparação dos resultados para o escoamento em um degrau, comprimento da região de recirculação, comprimento do vórtice primário e localização, para $\nu = 5 \times 10^{-3}$. . .	57
4.1	Erro e ordem de convergência na norma L_∞	68
5.1	Erro e ordem de convergência na norma L_∞ . Domínio sem interface imersa (Figura 5.4a).	75
5.2	Erro e ordem de convergência na norma L_∞ . Uma interface imersa de forma circular foi adicionada ao domínio (Figura 5.4b).	76
5.3	Erro e ordem de convergência na norma L_∞ . O quadrado que delimita o domínio é formado por uma interface imersa (Figura 5.4c).	76
7.1	Ordem de convergência MII-E - Poisson 1D: $f = 0$ se $\phi < 0$, $\beta = 0.3$	99
7.2	Ordem de convergência MII-I - Poisson 1D: $f = 0$ se $\phi < 0$	99
7.3	Ordem de convergência MII-E - Poisson 1D: domínio com 3 funções, $\beta = 0.001$. . .	100
7.4	Ordem de convergência MII-I - Poisson 1D: domínio com 3 funções.	100
7.5	Ordem de convergência MII-E - Poisson 2D, $\beta = 0.5$	101
7.6	Ordem de convergência MII-I - Poisson 2D.	101
7.7	Tempo de processamento - Poisson 1D: $f = 0$ se $\phi < 0$	102
7.8	Tempo de processamento - Poisson 1D: domínio com 3 funções.	102
7.9	Tempo de processamento - Poisson 2D	102
7.10	Erro e ordem de convergência - Poisson com Dirichlet.	102
7.11	Erro e ordem de convergência - Poisson com Neumann.	103
8.1	Erro e ordem de convergência na norma L_∞ - Equação de Stokes: exemplo I.	107
8.2	Erro e ordem de convergência na norma L_∞ - Equação de Stokes: exemplo II.	107
8.3	Erro e ordem de convergência na norma L_∞ - Equação de Stokes: exemplo III.	108
A.1	Coeficientes do esquema numérico para aproximar a primeira derivada no contorno. .	115

A.2	Coeficientes do esquema numérico para aproximar a segunda derivada no contorno. . .	116
A.3	Ordem de convergência da primeira derivada.	117
A.4	Ordem de convergência da segunda derivada.	117
A.5	Coeficientes dos estênceis colocados no contorno para a primeira derivada.	120
A.6	Coeficientes dos estênceis deslocados no contorno para a primeira derivada.	121
A.7	Coeficientes dos estênceis colocados no contorno para a segunda derivada.	123
A.8	Coeficientes dos estênceis colocados não uniformes no contorno para a segunda derivada.	124
B.1	Ordem de convergência na aplicação do filtro na direção x	126

Introdução

É um fato bem conhecido que, nas últimas décadas, a velocidade de processamento tem crescido num ritmo muito mais rápido do que a velocidade da memória [3]. Os computadores modernos podem executar dezenas, às vezes centenas, de operações de ponto flutuante na mesma quantidade de tempo necessário para obter alguns dados na memória principal. Assim, a fim de melhorar a velocidade e eficiência, muitas vezes reduz-se o uso de memória, mesmo que isso signifique mais processamento.

Ao lidar com simulações numéricas, métodos tradicionais de diferenças finitas de primeira e segunda ordens, devido à sua baixa precisão, exigem malhas muito finas, isto é, grande uso de memória. Métodos de ordem superior podem reduzir significativamente os requisitos de memória, mantendo a precisão. O ganho na redução do acesso à memória é muitas vezes maior do que o custo de processamento adicional. Esta é uma das principais justificativas para o desenvolvimento de métodos numéricos de alta ordem.

Uma forma de obter discretizações de alta ordem é por meio do uso de diferenças finitas compactas. A literatura é rica em estudos de esquemas de diferenças finitas compactas. Lele em [4] propôs uma família de tais formulações compactas de alta ordem. Adam [5] e Hirsh [6] discutem algumas vantagens dos métodos compactos de quarta ordem em comparação com os métodos tradicionais. Souza et al. em [7] demonstraram que o esquema compacto de sexta ordem é linearmente estável.

De forma geral, a aproximação das derivadas de uma função usando diferenças finitas compactas é implícita no sentido de que os valores da derivada em um nó particular são obtidos não só a partir dos valores da função, mas também a partir dos valores da derivada dos nós vizinhos. Por essa razão, os métodos compactos de diferenças finitas apresentam maior complexidade, quando aplicados às equações de Navier-Stokes e continuam a ser uma área ativa de investigação.

Duas famílias comumente utilizadas de métodos para a solução numérica das equações de Navier-Stokes são: uma composta pelos métodos baseados na formulação função corrente-vorticidade, e a

outra composta pelos métodos baseados em variáveis primitivas. Uma série de estudos sobre esquemas de diferenças finitas compactas para a formulação função corrente-vorticidade foram publicados recentemente [8, 9, 10]. Métodos de função corrente-vorticidade são vantajosos, devido ao seu menor custo computacional, quando comparado à solução das equações de Navier-Stokes como um sistema acoplado. Outra vantagem é que eles não sofrem do problema do desacoplamento da pressão. Por outro lado, os métodos de função corrente-vorticidade são considerados ineficientes quando lidam com problemas em 3D e quando condições de contorno complexas estão presentes [11].

Chorin [12, 13] introduziu uma alternativa baseada em variáveis primitivas, os chamados métodos de projeção. Para problemas 2D, o custo computacional é aproximadamente o mesmo quando comparado com os métodos de função corrente-vorticidade, e para problemas 3D, há ganhos consideráveis. Além disso, o desacoplamento da pressão pode ser tratado pelo uso de uma malha deslocada.

Entre os mais conhecidos métodos de projeção com variáveis primitivas para Navier-Stokes estão os de Kim e Moin [14] e Bell et. al. [15]. Ambos esses métodos são de segunda ordem temporal e espacial. A principal diferença entre eles é que Bell et. al. usa uma aproximação para a pressão na equação do momento, a fim de obter segunda ordem no tempo, enquanto Kim e Moin não a utilizam. Entretanto, Kim e Moin usam condições de contorno mais sofisticadas para o campo de velocidade intermediária, as quais incluem uma aproximação para a pressão. A classe de métodos de projeção que utiliza uma aproximação para a pressão na equação do momento é conhecida como *incremental pressure projection methods*, em contraste com *pressure-free projection methods*, que não incluem a pressão na equação do momento [16]. Outros métodos populares incluem Strikwerda [17], que aborda os métodos de diferenças finitas com malhas não-uniformes, e Gresho [18, 19], que aborda a questão das condições de contorno para o campo de velocidade intermediária em elementos finitos.

Na literatura, alguns métodos de diferenças finitas compactas, com quarta ordem de precisão espacial, combinados com vários tipos de discretizações temporais, foram publicados para escoamentos incompressíveis. Abide e Viazzo [20] aplicaram um esquema semi-implícito Crank-Nicolson/Adams-Bashforth à equação do momento. Segunda ordem de precisão temporal era esperada, mas apenas primeira ordem foi observada para o campo de pressão. Knikker [21] em vez de aplicar métodos compactos diretamente à equação de Poisson, concebeu um método iterativo, onde estênceis com 5 ou 7 pontos são utilizados. Os estênceis compactos usados por Boersma [22] não têm todos a mesma precisão formal. A precisão varia de segunda a sexta ordem. Os resultados refletem esse fato, e o método exibe uma precisão que varia dependendo do problema. Mais recentemente, Zhang et. al. [23] desenvolveu um método de diferenças finitas compacto de quarta ordem para malhas não uniformes. A análise da precisão dessa formulação é validada usando a análise de Fourier. No entanto, a equação de Poisson é resolvida usando diferenças finitas não-compactas de segunda ordem.

Uma grande limitação dos métodos de diferenças finitas compactas é que eles, em sua formulação mais simples, estão restritos a malhas e domínios retangulares. Os escoamentos de fluidos com superfícies livres são exemplos de problemas com domínios complexos. Uma das formas de contornar essa limitação é o uso de técnicas de fronteira ou interface imersas.

Peskin [24] desenvolveu o Método de Fronteira Imersa (MFI) para a simulação do fluxo de sangue nas válvulas cardíacas. Desde então, o MFI tornou-se um método numérico padrão para problemas de interface que envolvem termos forçantes ao longo de uma ou várias interfaces. O termo forçante, computacionalmente, é aproximado por um conjunto de funções delta discretas. No entanto, o MFI é de primeira ordem.

LeVeque e Li desenvolveram o Método de Interfaces Imersas (MII) [25, 26, 27, 28], que se constituiu em uma alternativa para resolver numericamente problemas de valor inicial/contorno em geometrias altamente complexas, que são frequentemente encontradas em muitos campos da ciência e engenharia. Esse método permite a imposição, de maneira simples, de uma grande variedade de condições de contorno [29], incluindo superfícies livres, interfaces fluido-fluido e fluido-sólido. A principal vantagem da técnica de interface imersa é a possibilidade de se utilizar métodos de alta ordem na discretização das geometrias, sem a necessidade de transformação de coordenadas, empregando uma malha cartesiana uniforme. Li e Ito [30] mostraram que o MII permite obter segunda ordem de precisão. Li e Lai [31], Le et al. [32] estenderam o MII para resolver as equações de Navier-Stokes, incluindo problemas com interfaces móveis. O sucesso desse método depende da derivação das condições de salto para velocidade e pressão. Leveque e Li [26] desenvolveram um MII de segunda ordem, para a solução das equações de Stokes com interfaces móveis. Em Tan et al. [33], o MII acoplado com *solvers* iterativos, resolvem as equações de Stokes sem o uso de métodos de projeção. Segunda ordem é observada.

Um avanço significativo no MII foi proposto por Wiegmann e Bube [34], que introduziram o método de interface imersa com salto explícito. Esses pesquisadores fizeram uma importante observação: técnicas de diferenças finitas falham quando aplicadas a funções não suaves, pois a expansão em série de Taylor, em que as técnicas se baseiam, são inválidas para funções não suaves. Nesse contexto, uma expansão em série de Taylor incluindo saltos é utilizada para obter aproximações de segunda ordem de precisão. É de extrema importância para essa formulação do MII ter um conhecimento a priori das condições de salto, que podem ser obtidas a partir de informações físicas ou das equações governantes.

Quarta ordem de precisão foi obtida por Linnick e Fasel [2], que desenvolveram um MII usando a formulação função corrente-vorticidade. Nesse MII, os saltos na função e em suas derivadas são calculados numericamente, por meio de um processo explícito e iterativo, ao invés de analiticamente como nas versões anteriores.

Recentemente Brehm e Fasel [35, 36] desenvolveram um MII robusto e de alta ordem. Diferentemente dos existentes na literatura, que foram projetados para consistência e precisão com uma checagem a posteriori da estabilidade numérica, esse novo método combina uma expansão em série de Taylor localizada em pontos de malha irregulares (pontos próximos à interface) com restrições de estabilidade como parte da construção do método.

Outro método que usa condições de salto na solução e no fluxo é conhecido como Método de Fluido Fantasma (GFM do inglês *Ghost Fluid method*) [37]. Desenvolvido para problemas elípticos

com interface, é mais simples que o MII, e tem a vantagem de resultar num sistema simétrico positivo definido. No entanto, só apresenta primeira ordem de precisão. O método GFM foi generalizado em [38] a fim de obter alta ordem na aproximação da equação de Poisson com coeficientes constantes.

O objetivo principal do presente trabalho é desenvolver um método de interface imersa compatível com diferenças finitas compactas de quarta ordem para a resolução numérica das equações de Navier-Stokes com variáveis primitivas em malha deslocada em duas dimensões. Apesar de esse objetivo não ter sido atingido em sua totalidade, grande progresso foi feito nessa direção. As principais contribuições desse trabalho estão listadas a seguir.

- Desenvolvimento de um método de projeção para as equações de Navier-Stokes com variáveis primitivas e em malha deslocada [39, 40]. Esse método é totalmente baseado em diferenças finitas compactas, incluindo a equação de Poisson. Possui segunda ordem temporal e quarta ordem espacial. Ao contrário da maioria dos métodos de projeção existentes na literatura, trata-se de um método verdadeiramente livre de pressão, ou seja, a cada passo de tempo, não há nenhuma dependência da pressão do tempo anterior, nem na equação do momento e nem nas condições de contorno.
- Implementação de um código acoplado para as equações de Navier-Stokes com segunda ordem espacial, utilizando um método de fronteira imersa.
- Extensão do método de interface imersa de Linnick e Fasel [2] para o cálculo de derivadas em malha deslocada, com quarta ordem de precisão [41]. O método original se aplicava apenas a malhas colocalizadas.
- Extensão do método de interface imersa de Linnick e Fasel [2] para a resolução da Equação de Poisson, com quarta ordem de precisão. O método desenvolvido é implícito e direto, ao contrário do original, que é explícito e iterativo [42].
- Implementação de um código de quarta ordem de precisão, utilizando diferenças finitas compactas e o método de interface imersa para malha deslocada na resolução das equações de Stokes;

O trabalho é dividido da seguinte maneira. No Capítulo 2, são apresentadas as equações de Navier-Stokes, juntamente com as discretizações espacial e temporal. No Capítulo 3, um novo método de projeção baseado em diferenças finitas compactas é apresentado. No capítulo 4, é apresentado um método acoplado para as equações de Navier-Stokes. Em seguida, no Capítulo 5, a esse método é incorporado um método de fronteira imersa de segunda ordem. Esse estudo é relevante, pois influenciou na formulação do MII utilizado para resolver as equações de Stokes. No Capítulo 6, é apresentado o MII de [2], e este é estendido para o cálculo de derivadas em malha deslocada. No Capítulo 7, é feita uma nova extensão do MII de [2], desta vez, para resolver a equação de Poisson. As equações de Stokes são resolvidas com o MII no Capítulo 8. As conclusões e trabalhos futuros

são apresentados no Capítulo 9. O Apêndice A apresenta uma técnica de construção dos métodos compactos utilizados. É feita a análise de Fourier desses métodos para estimar a sua acurácia. No Apêndice B é apresentado uma técnica de filtragem baseada em diferenças finitas compactas.

Equações de Navier-Stokes

Escoamentos transientes podem ser modelados pelas equações de Navier-Stokes. Tais equações traduzem, para a linguagem matemática, o princípio físico de conservação de momento. Neste trabalho, serão considerados escoamentos incompressíveis e isotérmicos de um fluido newtoniano, o que simplifica consideravelmente essas equações.

Em um domínio fechado Ω , as equações de Navier-Stokes para um fluido incompressível, isotérmico e newtoniano são dadas por

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{g}, \quad (2.1)$$

onde ν é a viscosidade cinemática, $\mathbf{u}$ o vetor velocidade, p a pressão (dividida pela densidade) e $\mathbf{g}$ um campo de aceleração. Para obter uma descrição completa do escoamento, deve-se considerar também a equação da continuidade, que reflete a conservação de massa:

$$\nabla \cdot \mathbf{u} = 0. \quad (2.2)$$

As condições de contorno do tipo Dirichlet para a velocidade são

$$\mathbf{u}|_{\partial\Omega} = \mathbf{u}_b. \quad (2.3)$$

Não há condições de contorno para a pressão quando as condições de contorno da velocidade são do tipo Dirichlet.

As equações (2.1) e (2.2) constituem um sistema de EDPs cuja complexidade matemática e computacional têm despertado a atenção de muitos pesquisadores. Os métodos numéricos para tal sistema podem ser classificados, de maneira geral, em métodos segregados e métodos acoplados.

Os métodos acoplados discretizam as equações (2.1) e (2.2) simultaneamente, calculando $\mathbf{u}$ e p em um mesmo processo. Já em métodos segregados, a solução das equações de Navier-Stokes incompressíveis se faz calculando $\mathbf{u}$ e p de forma consecutiva e separada. Esse desacoplamento na solução de $\mathbf{u}$ e p constitui os métodos de projeção [43, 44]. A formulação matemática dos métodos de projeção está ancorada no teorema da Decomposição de Helmholtz-Hodge, cuja demonstração pode ser consultada em [45].

Neste trabalho, foram considerados tanto métodos acoplados quanto segregados. Nos próximos capítulos, serão apresentados os métodos estudados e desenvolvidos.

2.1 Discretização temporal

Neste trabalho, serão utilizadas discretizações de segunda ordem temporal, tanto para a pressão quanto para a velocidade. Uma discretização de segunda ordem temporal para (2.1) e (2.2) é

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} = -\nabla p^{n+1/2} - [\mathbf{u} \cdot \nabla \mathbf{u}]^{n+1/2} + \nu \nabla^2 \mathbf{u}^{n+1/2} + \mathbf{g}^{n+1/2}, \quad (2.4a)$$

$$\nabla \cdot \mathbf{u}^{n+1} = 0, \quad (2.4b)$$

com condições de contorno

$$\mathbf{u}^{n+1}|_{\partial\Omega} = \mathbf{u}_b^{n+1}, \quad (2.5)$$

onde o sobrescrito n indica que a quantidade é amostrada no tempo discreto t^n . $\Delta t = t^{n+1} - t^n$ é o passo de tempo. O termo convectivo $[\mathbf{u} \cdot \nabla \mathbf{u}]^{n+1/2}$ pode ser aproximado, com segunda ordem temporal, de forma implícita, semi-implícita ou explícita.

As aproximações implícitas de segunda ordem mais comuns são baseadas na regra do ponto médio:

$$[\mathbf{u} \cdot \nabla \mathbf{u}]^{n+1/2} = \left(\frac{\mathbf{u}^{n+1} + \mathbf{u}^n}{2} \right) \cdot \nabla \left(\frac{\mathbf{u}^{n+1} + \mathbf{u}^n}{2} \right) + \mathcal{O}(\Delta t^2). \quad (2.6)$$

ou no método de Crank-Nicolson [46]:

$$[\mathbf{u} \cdot \nabla \mathbf{u}]^{n+1/2} = \frac{[\mathbf{u} \cdot \nabla \mathbf{u}]^{n+1} + [\mathbf{u} \cdot \nabla \mathbf{u}]^n}{2} + \mathcal{O}(\Delta t^2). \quad (2.7)$$

Essas aproximações produzem sistemas de equações não-lineares que podem ser resolvidos por métodos convencionais, como o método de Newton.

A aproximação semi-implícita ou linearizada, combina aproximações explícitas e implícitas. Por exemplo, pode-se combinar os esquemas Adams-Bashforth e a regra do ponto médio:

$$[\mathbf{u} \cdot \nabla \mathbf{u}]^{n+1/2} = \left(\frac{3}{2} \mathbf{u}^n - \frac{1}{2} \mathbf{u}^{n-1} \right) \cdot \nabla \left(\frac{\mathbf{u}^{n+1} + \mathbf{u}^n}{2} \right) + \mathcal{O}(\Delta t^2). \quad (2.8)$$

Outra possibilidade proposta por [47] é:

$$u_i^{n+1}u_j^{n+1} = u_i^{n+1}u_j^n + u_i^n u_j^{n+1} - u_i^n u_j^n + \mathcal{O}(\Delta t^2), \quad (2.9)$$

onde u_i representa uma componente do vetor $\mathbf{u}$.

Uma vantagem das aproximações semi-implícitas está no fato de o sistema resultante ser linear.

Uma aproximação explícita é dada pelo esquema de Adams-Bashforth:

$$[\mathbf{u} \cdot \nabla \mathbf{u}]^{n+1/2} = \frac{3}{2}[\mathbf{u} \cdot \nabla \mathbf{u}]^n - \frac{1}{2}[\mathbf{u} \cdot \nabla \mathbf{u}]^{n-1} + \mathcal{O}(\Delta t^2). \quad (2.10)$$

Nos métodos de projeção, a serem apresentados no próximo capítulo, utiliza-se a aproximação explícita que é mais frequente na literatura [14].

Para o termo viscoso $\nabla^2 \mathbf{u}^{n+1/2}$, aproximações implícitas de segunda ordem temporal são dadas por Crank-Nicolson ou a regra do ponto médio. Por se tratar de um termo linear, ambas as aproximações têm a mesma expressão

$$\nabla^2 \mathbf{u}^{n+1/2} = \frac{\nabla^2 \mathbf{u}^n + \nabla^2 \mathbf{u}^{n+1}}{2} + \mathcal{O}(\Delta t^2). \quad (2.11)$$

A aproximação explícita por Adams-Bashforth é

$$\nabla^2 \mathbf{u}^{n+1/2} = \frac{3}{2}\nabla^2 \mathbf{u}^n - \frac{1}{2}\nabla^2 \mathbf{u}^{n-1} + \mathcal{O}(\Delta t^2). \quad (2.12)$$

A aproximação implícita é bastante popular pois relaxa restrições no passo de tempo, que podem ser muito severas no caso de esquemas explícitos [48].

2.2 Discretização espacial

Nesta seção serão apresentados os métodos numéricos a serem utilizados, à saber, um método de diferenças finitas de segunda ordem e um método de diferenças finitas compactas de quarta ordem. Por simplicidade, serão apresentadas as discretizações apenas na direção x . Na direção y a derivação é análoga.

Os códigos a serem desenvolvidos utilizam malha deslocada [49, 50], onde as velocidades são calculadas nas arestas e a pressão no centro da célula. Isso evita o conhecido problema do desacoplamento da pressão, que produz oscilações espúrias [51, 52]. A Figura 2.1 ilustra uma célula da malha e sua representação computacional.

2.2.1 Aproximações por diferenças finitas

O método das diferenças finitas é um método de discretização de equações diferenciais que se baseia na expansão em série de Taylor da função derivada [53]. Esta seção apresenta as aproximações numéricas utilizadas no cálculo das derivadas presentes em (2.1) e (2.2).

Figura 2.1: Ilustração de uma malha deslocada e sua representação computacional.

2.2.1.1 Aproximação para a primeira derivada

As aproximações de segunda ordem para o termo convectivo e o gradiente de pressão são dadas por

$$u \frac{\partial u}{\partial x} \Big|_{i+1/2,j} = u_{i+1/2,j} \left(\frac{u_{i+1/2,j} - u_{i-1/2,j}}{\Delta x} \right) + \mathcal{O}(\Delta x^2), \quad (2.13)$$

$$\frac{\partial p}{\partial x} \Big|_{i,j} = \frac{p_{i+1,j} - p_{i-1,j}}{2\Delta x} + \mathcal{O}(\Delta x^2). \quad (2.14)$$

Para o termo convectivo, $v \frac{\partial u}{\partial y}$, é necessário, primeiramente, interpolar a velocidade v para a posição da velocidade u , como a seguir

$$v \frac{\partial u}{\partial y} \Big|_{i+1/2,j} = \left(\frac{v_{i-1/2,j+1} + v_{i+1/2,j+1} + v_{i-1/2,j} + v_{i+1/2,j}}{4} \right) \frac{u_{i+1/2,j+1} - u_{i+1/2,j-1}}{2\Delta y} + \mathcal{O}(\Delta y^2). \quad (2.15)$$

2.2.1.2 Aproximação para a segunda derivada

Para o termo viscoso, usa-se

$$\frac{\partial^2 u}{\partial x^2} \Big|_{i+1/2,j} = \frac{u_{i-1/2,j} - 2u_{i+1/2,j} + u_{i+3/2,j}}{\Delta x^2} + \mathcal{O}(\Delta x^2). \quad (2.16)$$

2.2.1.3 Aproximação para a equação de Poisson

A aproximação para a equação de Poisson resulta num sistema pentadiagonal:

$$(\nabla^2 p)_{i,j} = (\nabla \cdot \mathbf{u})_{i,j}, \quad (2.17)$$

$$(\nabla^2 p)_{i,j} = \frac{p_{i+1,j} - 2p_{i,j} + p_{i-1,j}}{\Delta x^2} + \frac{p_{i,j+1} - 2p_{i,j} + p_{i,j-1}}{\Delta y^2} + \mathcal{O}(\Delta x^2, \Delta y^2), \quad (2.18a)$$

$$(\nabla \cdot \mathbf{u})_{i,j} = \frac{u_{i+1/2,j} - u_{i-1/2,j}}{\Delta x} + \frac{v_{i,j+1/2} - v_{i,j-1/2}}{\Delta y} + \mathcal{O}(\Delta x^2, \Delta y^2). \quad (2.18b)$$

2.2.1.4 Condições de contorno

Ao se aproximar as derivadas, próximo à borda, o estêncil pode englobar um ponto fora do domínio, como na Figura 2.2. O valor da velocidade nesse ponto é obtido por interpolação

$$u_{i+1/2,j-1} = 2U - u_{i+1/2,j}, \quad (2.19)$$

onde $u_{i+1/2,j-1}$ está fora do domínio, $u_{i+1/2,j}$ está dentro do domínio e U é o valor da velocidade na borda, dado pela condição de contorno.

Figura 2.2: Ilustração do estêncil próximo ao contorno.

2.2.2 Aproximações por diferenças finitas compactas

Seguindo [4], esta seção apresenta as aproximações numéricas utilizadas no cálculo das derivadas presentes em (2.1) e (2.2). A técnica utilizada é o método de diferenças finitas compactas em uma malha deslocada com quarta ordem. O Apêndice A descreve os passos para construir os estêncis não centrais próximos à borda. Em alguns casos, mais de um estêncil é possível. Para a escolha do mais adequado, a análise de Fourier, que mede a resolução de cada método, também é apresentada no Apêndice A. A discretização para a equação de Poisson será mostrada na Seção 3.2.1.

2.2.2.1 Aproximação para o termo viscoso usando diferenças compactas

Interior do domínio

Para o termo viscoso, usa-se aproximações para a segunda derivada conforme mostrado em (A.8):

$$\frac{\partial^2 u}{\partial x^2} \Big|_{i-1/2,j} + 10 \frac{\partial^2 u}{\partial x^2} \Big|_{i+1/2,j} + \frac{\partial^2 u}{\partial x^2} \Big|_{i+3/2,j} = \frac{12(u_{i-1/2,j} - 2u_{i+1/2,j} + u_{i+3/2,j})}{\Delta x^2} + \mathcal{O}(\Delta x^4), \quad (2.20a)$$

$$\frac{\partial^2 u}{\partial y^2} \Big|_{i+1/2,j-1} + 10 \frac{\partial^2 u}{\partial y^2} \Big|_{i+1/2,j} + \frac{\partial^2 u}{\partial y^2} \Big|_{i+1/2,j+1} = \frac{12(u_{i+1/2,j-1} - 2u_{i+1/2,j} + u_{i+1/2,j+1})}{\Delta y^2} + \mathcal{O}(\Delta y^4). \quad (2.20b)$$

Contorno

$$12 \frac{\partial^2 u}{\partial x^2} \Big|_{i+1/2,j} + 120 \frac{\partial^2 u}{\partial x^2} \Big|_{i+3/2,j} = \frac{1}{h^2} (145u_{i+1/2,j} - 304u_{i+3/2,j} + 174u_{i+5/2,j} - 16u_{i+7/2,j} + u_{i+9/2,j}) + \mathcal{O}(\Delta x^4), \quad (2.21a)$$

$$1134 \frac{\partial^2 u}{\partial y^2} \Big|_{i+1/2,j} + 378 \frac{\partial^2 u}{\partial y^2} \Big|_{i+1/2,j+1} = \frac{1}{h^2} (-5481u_{i+1/2,j} + 1722u_{i+1/2,j+1} + 54u_{i+1/2,j+3} - 7u_{i+1/2,j+4} + 3712u_{i+1/2,j-1}) + \mathcal{O}(\Delta y^4). \quad (2.21b)$$

2.2.2.2 Aproximação para o termo convectivo usando diferenças compactas

Interior do domínio

O termo convectivo é aproximado por

$$\frac{\partial u^2}{\partial x} \Big|_{i-1/2,j} + 4 \frac{\partial u^2}{\partial x} \Big|_{i+1/2,j} + \frac{\partial u^2}{\partial x} \Big|_{i+3/2,j} = \frac{3(u_{i+3/2,j}^2 - u_{i-1/2,j}^2)}{\Delta x} + \mathcal{O}(\Delta x^4), \quad (2.22a)$$

$$\frac{\partial uv}{\partial y} \Big|_{i+1/2,j-1} + 22 \frac{\partial uv}{\partial y} \Big|_{i+1/2,j} + \frac{\partial uv}{\partial y} \Big|_{i+1/2,j+1} = \frac{24((uv)_{i+1/2,j+1/2} - (uv)_{i+1/2,j-1/2})}{\Delta x} + \mathcal{O}(\Delta x^4). \quad (2.22b)$$

Em (2.22b), é necessário interpolar os valores de u e v das arestas para o canto da célula computacional, o que é obtido por

$$u_{i+1/2,j-1/2} + 6u_{i+1/2,j+1/2} + u_{i+1/2,j+3/2} = 4(u_{i+1/2,j} + u_{i+1/2,j+1}) + \mathcal{O}(\Delta y^4), \quad (2.23a)$$

$$v_{i-1/2,j+1/2} + 6v_{i+1/2,j+1/2} + v_{i+3/2,j+1/2} = 4(v_{i,j+1/2} + v_{i+1,j+1/2}) + \mathcal{O}(\Delta x^4). \quad (2.23b)$$

Contorno

$$6 \frac{\partial u^2}{\partial x} \Big|_{i+1/2,j} + 18 \frac{\partial u^2}{\partial x} \Big|_{i+3/2,j} = \frac{1}{\Delta x} (-17u_{i+1/2,j}^2 + 9u_{i+3/2,j}^2 + 9u_{i+5/2,j}^2 - u_{i+7/2,j}^2) + \mathcal{O}(\Delta x^4), \quad (2.24a)$$

$$24 \frac{\partial uv}{\partial y} \Big|_{i+1/2,j} - 120 \frac{\partial uv}{\partial y} \Big|_{i+1/2,j+1} = \frac{1}{\Delta y} (-27(uv)_{i+1/2,j} + 152(uv)_{i+1/2,j+1} - 126(uv)_{i+1/2,j+2} + (uv)_{i+1/2,j+4}) + \mathcal{O}(\Delta y^4). \quad (2.24b)$$

2.2.2.3 Aproximação para o gradiente de pressão usando diferenças compactas

A discretização do gradiente de pressão é feita no ponto (i, j) e um esquema para malha deslocada, semelhante àquele usando no termo convectivo dado na Subseção 2.2.2.2, é usado.

Métodos de projeção

Em geral, um método de projeção é definido pelos seguintes passos [54]:

Passo 1: Define-se um campo de velocidade intermediário $\mathbf{u}^*$ que é calculado por meio de

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = -\nabla q - [\mathbf{u} \cdot \nabla \mathbf{u}]^{n+1/2} + \nu \nabla^2 \mathbf{u}^{n+1/2} + \mathbf{g}^{n+1/2}, \quad (3.1a)$$

$$B(\mathbf{u}^*) = 0, \quad (3.1b)$$

onde q representa uma aproximação para $p^{n+1/2}$ e $B(\mathbf{u}^*) = 0$ as condições de contorno para $\mathbf{u}^*$. Cada método especifica escolhas diferentes para q e $B(\mathbf{u}^*)$. Essas escolhas influenciam significativamente a ordem temporal do método. Se os termos viscoso ou convectivo forem aproximados de forma implícita ou semi-implícita, nessa aproximação deve-se substituir $\mathbf{u}^{n+1}$ por $\mathbf{u}^*$.

Passo 2: Faz-se a projeção

$$\mathbf{u}^* = \mathbf{u}^{n+1} + \Delta t \nabla \phi^{n+1}, \quad (3.2a)$$

$$\nabla \cdot \mathbf{u}^{n+1} = 0, \quad (3.2b)$$

usando condições de contorno consistentes com $B(\mathbf{u}^*) = 0$ e $\mathbf{u}^{n+1}|_{\partial\Omega} = \mathbf{u}_b^{n+1}$.

Esse sistema reduz-se a uma equação de Poisson quando se aplica o operador divergente em (3.2a)

$$\nabla \cdot \nabla \phi^{n+1} = \frac{1}{\Delta t} \nabla \cdot \mathbf{u}^*, \quad (3.3)$$

com condições de contorno

$$\hat{\mathbf{n}} \cdot \nabla \phi^{n+1}|_{\partial\Omega} = \hat{\mathbf{n}} \cdot (\mathbf{u}^*|_{\partial\Omega} - \mathbf{u}_b^{n+1}). \quad (3.4)$$

Passo 3: Atualiza-se a pressão

$$p^{n+1/2} = q + L(\phi^{n+1}), \quad (3.5)$$

onde a função L representa a dependência de $p^{n+1/2}$ sobre ϕ^{n+1} . Para obtê-la, basta substituir a equação (3.2a) em (3.1a) e comparar com (2.4a). Essa função depende do tipo de discretização temporal usada nos termos viscoso e convectivo. A nova velocidade $\mathbf{u}^{n+1}$ é calculada por (3.2a).

Os passos 1-3 determinam o avanço de um passo no tempo.

3.1 Método de projeção com pressão incremental e livre de pressão

A escolha da aproximação da pressão q , da condição de contorno $B(\mathbf{u}^*)$ e das aproximações temporais para os termos viscoso e convectivo são importantes na definição do método. As condições de contorno para $\mathbf{u}^*$ devem ser consistentes com a equação (3.2a), embora elas dependam de valores de ϕ^{n+1} que ainda são desconhecidos e precisam ser aproximados.

Em [15, 55] considera-se $q = p^{n-1/2}$ e $B(\mathbf{u}^*) = (\mathbf{u}^* - \mathbf{u}_b^{n+1})|_{\partial\Omega} = 0$. A pressão é atualizada por

$$p^{n+1/2} = p^{n-1/2} + \phi^{n+1}. \quad (3.6)$$

Essa expressão de atualização da pressão é válida quando se utiliza métodos explícitos para os termos viscoso e convectivo. Se outros métodos não totalmente explícitos forem utilizados para os termos convectivo ou viscoso, outros termos apareceriam em (3.6).

Esse método é considerado como sendo um **método de projeção com pressão incremental** (do inglês *incremental-pressure projection method*). A ideia por trás dessa classe de métodos é que o termo q , que aproxima $p^{n+1/2}$, faça com que $\mathbf{u}^*$ fique mais próximo de $\mathbf{u}^{n+1}$, e com isso seja possível usar as mesmas condições de contorno para ambos.

Por outro lado, em [14] considera-se $q = 0$, e como condição de contorno para $\mathbf{u}^*$, usa-se $\mathbf{u}^*|_{\partial\Omega} = \mathbf{u}_b^{n+1} + \Delta t \nabla \phi^n|_{\partial\Omega}$. A pressão é dada por

$$p^{n+1/2} = \phi^{n+1}, \quad (3.7)$$

para discretizações temporais explícitas. Visto que não há pressão na equação do momento, esse tipo de método é chamado de **método de projeção livre de pressão** (do inglês *pressure-free projection method*). Isso é interessante pois erros no gradiente da pressão são eliminados da equação do

momento, os quais podem se acumular com o tempo.

A falta de consistência na escolha de q e $B(\mathbf{u}^*)$ ou no cálculo de $L(\phi^{n+1})$ pode introduzir erros numéricos não somente nos transientes como também nos estados estacionários. Por exemplo, a escolha $q = 0$ e $B(\mathbf{u}^*) = (\mathbf{u}^* - \mathbf{u}_b^{n+1})|_{\partial\Omega} = 0$ pode introduzir um erro proporcional a Δt nos transientes e no estado estacionário [56, 57].

Tanto no método com pressão incremental quanto no livre de pressão, se desejada, a pressão no tempo t^{n+1} pode ser obtida por interpolação:

$$p^{n+1} = \frac{3}{2}p^{n+1/2} - \frac{1}{2}p^{n-1/2} + \mathcal{O}(\Delta t^2). \quad (3.8)$$

3.2 Estêncil compatível: projeção exata e projeção aproximada

O fato de um método de projeção ser exato ou aproximado, está relacionado com a resolução da equação de Poisson (3.3). Quando se resolve esta equação

$$\nabla \cdot \nabla p = \frac{1}{\Delta t} \nabla \cdot \mathbf{u}^*, \quad (3.9)$$

ou, em coordenadas cartesianas

$$\frac{\partial}{\partial x} \frac{\partial}{\partial x} p + \frac{\partial}{\partial y} \frac{\partial}{\partial y} p = \frac{1}{\Delta t} \left(\frac{\partial u^*}{\partial x} + \frac{\partial v^*}{\partial y} \right), \quad (3.10)$$

especial atenção deve se ter com o operador $(\nabla \cdot \nabla)$. Enquanto no caso contínuo esse operador é o mesmo que o laplaciano ∇^2 , no caso discreto isso não é sempre verdade. Em particular, quando se trata de diferenças finitas compactas, não é possível obter um operador laplaciano compacto que seja exatamente igual ao operador $(\nabla \cdot \nabla)$. Se um operador laplaciano compacto for usado, o método de projeção não será exato, mas apenas aproximado, e a condição de incompressibilidade será satisfeita a menos da ordem do método.

Para exemplificar, em 1D, usando o método de diferenças finitas padrão (não compacto),

$$\begin{aligned} \nabla \cdot \nabla p_i &= \frac{\partial}{\partial x} \frac{\partial}{\partial x} p_i = \\ \frac{\partial}{\partial x} \left(\frac{p_{i+1/2} - p_{i-1/2}}{\Delta x} \right) + \mathcal{O}(\Delta x^2) &= \\ \frac{\frac{p_{i+1} - p_i}{\Delta x} - \frac{p_i - p_{i-1}}{\Delta x}}{\Delta x} + \mathcal{O}(\Delta x^2) &= \\ \frac{p_{i+1} - 2p_i + p_{i-1}}{\Delta x^2} + \mathcal{O}(\Delta x^2) &= \frac{\partial^2}{\partial x^2} p_i = \nabla^2 p_i. \end{aligned} \quad (3.11)$$

Portanto, o uso de diferença finitas padrão produz um estêncil compatível. Com isso, espera-se que a

condição de incompressibilidade seja satisfeita exatamente a cada passo de tempo, a menos de erros de arredondamento da máquina.

Por outro lado, o uso do método de diferenças finitas compactas,

$$\nabla^2 p_i = \frac{\partial^2}{\partial x^2} p_i = p_{xx_i}, \quad (3.12)$$

$$p_{xx_{i-1}} + 10p_{xx_i} + p_{xx_{i+1}} = \frac{12}{\Delta x^2}(p_{i-1} + 2p_i + p_{i+1}) + \mathcal{O}(\Delta x^4). \quad (3.13)$$

$$\nabla \cdot \nabla p_i = \frac{\partial}{\partial x} \frac{\partial}{\partial x} p_i = p_{xx_i} \Leftrightarrow \begin{cases} \frac{\partial}{\partial x} p_i = p_{x_i}, \\ \frac{\partial}{\partial x} p_{x_i} = p_{xx_i}, \end{cases} \quad (3.14)$$

$$\begin{cases} p_{x_{i-1}} + 4p_{x_i} + p_{x_{i+1}} = \frac{3}{\Delta x}(p_{i+1} - p_{i-1}) + \mathcal{O}(\Delta x^4), \\ p_{xx_{i-1}} + 4p_{xx_i} + p_{xx_{i+1}} = \frac{3}{\Delta x}(p_{x_{i+1}} - p_{x_{i-1}}) + \mathcal{O}(\Delta x^4), \end{cases} \quad (3.15)$$

onde p_{x_i} , p_{xx_i} representam a aproximação numérica para a primeira e a segunda derivada de p .

Pode-se mostrar [20] que a solução do sistema de equações (3.15) não é igual à solução da equação (3.13). Se a equação de Poisson for discretizada como em (3.13) o método será aproximado e a condição de incompressibilidade será satisfeita a menos da ordem do método. Já o uso do sistema (3.15) produz um resultado que satisfaz a condição de incompressibilidade exatamente, a menos de erros de máquina.

3.2.1 Projeção aproximada em 2D

A seguir, será ilustrado como determinar um estêncil 2D a partir de estênceis 1D. A discretização para a equação de Poisson no ponto (i, j) é

$$(\nabla^2 \phi)_{i,j} = (\nabla \cdot \mathbf{u}^*)_{i,j}, \quad (3.16)$$

em coordenadas cartesianas,

$$\left(\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} \right) \Big|_{i,j} = \left(\frac{\partial u^*}{\partial x} + \frac{\partial v^*}{\partial y} \right) \Big|_{i,j}. \quad (3.17)$$

Nessa equação, temos que o valor do laplaciano de ϕ é conhecido, mas o valor da função ϕ não. A discretização da equação (3.16), para determinar ϕ , é baseada em duas aproximações de quarta ordem

de diferenças compactas em uma dimensão:

$$\sum_{\xi=-1}^1 L_{i+\xi}^{xx} \phi_{i+\xi,j}^{xx} = \sum_{\xi=-1}^1 R_{i+\xi}^{xx} \phi_{i+\xi,j}, \quad (3.18a)$$

$$\sum_{\eta=-1}^1 L_{j+\eta}^{yy} \phi_{i,j+\eta}^{yy} = \sum_{\eta=-1}^1 R_{j+\eta}^{yy} \phi_{i,j+\eta}, \quad (3.18b)$$

onde $L^{xx}, L^{yy}, R^{xx}, R^{yy}$ denotam os coeficientes no esquema, enquanto que ϕ^{xx} e ϕ^{yy} representam aproximações numéricas para a segunda derivada nas direções x e y . O esquema numérico é determinado como em [2]. Nas equações seguintes, o somatório será omitido e estará implícito (notação de Einstein).

$$\phi_{ij}^{xx} + \phi_{ij}^{yy} = (\nabla \cdot \mathbf{u}^*)_{i,j}, \quad (3.19a)$$

$$L_i^{xx} L_j^{yy} (\phi_{ij}^{xx} + \phi_{ij}^{yy}) = L_i^{xx} L_j^{yy} (\nabla \cdot \mathbf{u}^*)_{i,j}, \quad (3.19b)$$

$$L_j^{yy} (L_i^{xx} \phi_{ij}^{xx}) + L_i^{xx} (L_j^{yy} \phi_{ij}^{yy}) = L_i^{xx} L_j^{yy} (\nabla \cdot \mathbf{u}^*)_{i,j}. \quad (3.19c)$$

Usando (3.18a) e (3.18b) segue que

$$L_j^{yy} (R_i^{xx} \phi_{ij}) + L_i^{xx} (R_j^{yy} \phi_{ij}) = L_i^{xx} L_j^{yy} (\nabla \cdot \mathbf{u}^*)_{i,j}, \quad (3.20a)$$

$$(R_i^{xx} L_j^{yy} + L_i^{xx} R_j^{yy}) \phi_{ij} = L_i^{xx} L_j^{yy} (\nabla \cdot \mathbf{u}^*)_{i,j}. \quad (3.20b)$$

Assim, para calcular ϕ , o esquema compacto em 2D de nove pontos centrado no ponto (i, j) é dado por

$$\mathcal{L}_{ij} \phi_{ij} = \mathcal{R}_{ij} (\nabla \cdot \mathbf{u}^*)_{i,j}, \quad (3.21)$$

onde

$$\mathcal{L}_{ij} = R_i^{xx} L_j^{yy} + L_i^{xx} R_j^{yy}, \quad (3.22a)$$

$$\mathcal{R}_{ij} = L_i^{xx} L_j^{yy}. \quad (3.22b)$$

Essa forma de determinar estênceis 2D a partir de estênceis 1D é feita ponto a ponto e se torna genérica, servindo para esquemas de qualquer ordem, e estênceis de qualquer tamanho, sejam compactos ou não.

As velocidades u e v são calculadas nas faces, mas o termo $(\nabla \cdot \mathbf{u}^*)_{i,j}$ é calculado no centro da célula. Por isso, um esquema para malha deslocada semelhante ao da Subseção 2.2.2.2, será necessário.

Um estêncil 1D da forma

$$-5760\phi_i + 5760\phi_{i+1} \approx 5760\phi_{i-1/2}^{(1)}h + (6007\phi_i^{(2)} - 981\phi_{i+1}^{(2)} + 981\phi_{i+2}^{(2)} - 247\phi_{i+3}^{(2)})h^2, \quad (3.23)$$

é utilizado na equação de Poisson nos pontos próximos à fronteira rígida, onde o ponto $(i - 1/2)$ está sobre o contorno. Foi adotada a condição de contorno do tipo Neumann homogênea para ϕ , ou seja,

$$\left. \frac{\partial \phi}{\partial \mathbf{n}} \right|_{i-1/2} = \phi_{i-1/2}^{(1)}. \quad (3.24)$$

3.2.2 Projeção exata em 2D

Para obter um método exato com diferenças finitas compactas, a equação diferencial (3.10), que é bidimensional e de segunda ordem, pode ser reescrita como

$$p_x - \frac{\partial}{\partial x} p = 0, \quad (3.25a)$$

$$p_y - \frac{\partial}{\partial y} p = 0, \quad (3.25b)$$

$$p_{xx} - \frac{\partial}{\partial x} p_x = 0, \quad (3.25c)$$

$$p_{yy} - \frac{\partial}{\partial y} p_y = 0, \quad (3.25d)$$

$$p_{xx} + p_{yy} = \frac{1}{\Delta t} \left(\frac{\partial u^*}{\partial x} + \frac{\partial v^*}{\partial y} \right), \quad (3.25e)$$

que é um sistema de equações diferenciais unidimensionais de primeira ordem, com exceção da última que é puramente algébrica, uma vez que o termo do lado direito pode ser calculado previamente. Desta forma, as diferenças finitas compactas podem ser facilmente aplicadas para discretizar cada equação. Se a mesma discretização for usada para todas as equações, um método de projeção exata será obtido.

Em malhas deslocadas, as quantidades p , p_{xx} e p_{yy} estão no centro das células, enquanto p_x e p_y estão sobre as arestas das células, na mesma posição que u^* e v^* , respectivamente.

Quando se calcula $\mathbf{u}^*$, assume-se que $\mathbf{u}^* = \mathbf{u}_b^{n+1}$ no contorno, o que é equivalente a considerar $p_x = 0$ e $p_y = 0$ no contorno. Dessa forma, nem u^* , v^* , p_x ou p_y precisam ser calculados no contorno. Essa escolha de condições de contorno não compromete a precisão do método, mesmo no caso de projeção livre de pressão, como será mostrado na seção de resultados.

Essa abordagem tem várias vantagens:

- não é necessário construir estênceis para o operador laplaciano, o que é bastante complexo quando se quer um método exato baseado em diferenças finitas compactas. Este procedimento é, no entanto, possível, mas o estêncil resultante não seria compacto, e poderia se estender ao longo do comprimento do domínio. Um exemplo de como tratar esse caso pode ser encontrado em [20];
- além de calcular a pressão p , as quantidades p_x e p_y já são calculadas para o próximo passo do método de projeção, ou seja, a correção da velocidade;

- não são necessárias células fantasmas, interpolações ou extrapolações;
- cantos no domínio computacional não necessitam de tratamento especial, uma vez que apenas equações unidimensionais estão presentes.

A única desvantagem encontrada é que o sistema linear é maior em relação ao número de equações e incógnitas. No entanto, a matriz é esparsa quando comparada àquela usada em [20]. Na presente abordagem, cada linha da matriz tem, no máximo, 6 elementos não nulos, independentemente do tamanho do domínio. Em [20], o número de não zeros é proporcional ao comprimento do domínio. Desta forma, um domínio $n \times n$ produziria uma matriz com $\mathcal{O}(n^2)$ não zeros na presente formulação, enquanto que com [20] a matriz poderia ter $\mathcal{O}(n^3)$ não zeros.

3.3 Condição de compatibilidade

Quando se resolve a equação de Poisson (3.3), sujeita exclusivamente a condições de contorno do tipo Neumann, duas preocupações surgem. Uma é conhecida como a condição de compatibilidade, que requer que

$$\int_{\Omega} \left(\frac{1}{\Delta t} \nabla \cdot \mathbf{u}^* \right) dV = \oint_{\partial\Omega} \frac{\partial \phi}{\partial \mathbf{n}} dS. \quad (3.26)$$

Se verdadeiro, a segunda preocupação é que p será determinada a menos de uma constante.

No caso discreto, existe uma condição de compatibilidade análoga que deve ser satisfeita pelo sistema linear resultante da aproximação. As condições de contorno que satisfazem a condição de compatibilidade no caso contínuo podem não a satisfazer no caso discreto. Correções devem ser aplicadas no campo de velocidade ou no termo independente do sistema linear. Técnicas para garantir a compatibilidade podem ser encontradas em [58, 59, 60].

A condição de compatibilidade discreta está diretamente relacionada à alternativa de Fredholm [61]. Esse teorema, em uma formulação matricial simplificada, diz que: dado um sistema linear $A\mathbf{x} = \mathbf{b}$, exatamente uma das seguintes afirmações é verdadeira. Ou $A\mathbf{x} = \mathbf{b}$ tem uma solução, ou $\mathbf{v}A = 0$ tem uma solução com $\mathbf{v} \cdot \mathbf{b} \neq 0$. Em outras palavras, para que $A\mathbf{x} = \mathbf{b}$ tenha solução, $\mathbf{v} \cdot \mathbf{b}$ deve ser zero para algum $\mathbf{v}$ com $\mathbf{v}A = 0$. Em [60], é sugerido que o vetor do lado direito $\mathbf{b}$ seja substituído por

$$\mathbf{b}^* = \mathbf{b} - \frac{\mathbf{v} \cdot \mathbf{b}}{\mathbf{v} \cdot \mathbf{v}} \mathbf{v}. \quad (3.27)$$

É claro que $\mathbf{v} \cdot \mathbf{b}^* = 0$, e portanto $A\mathbf{x} = \mathbf{b}^*$ tem uma solução.

No presente trabalho, são usadas apenas condições de contorno do tipo Dirichlet para a velocidade. Quando não há velocidade normal às fronteiras, ou seja, não há *inflow* ou *outflow*, a condição de compatibilidade é automaticamente satisfeita para a discretização usada. Mas a presença de *inflow* ou *outflow* pode fazer com que o sistema seja incompatível.

Quando o sistema é compatível, a matriz é singular, e existem infinitas soluções para p . Alguns *solvers* de sistema linear, em especial os iterativos, conseguem convergir para uma única solução sem nenhum esforço adicional. Se isso não acontecer, como é frequentemente o caso com *solvers* diretos,

a pressão pode ser fixada em único ponto para forçar uma solução única. Neste trabalho adotou-se essa estratégia.

Quando o sistema não é compatível, um pico no valor da divergência da velocidade final aparece no ponto em que a pressão foi prescrita. Isso é resolvido por meio de um processo iterativo que ajusta as condições de contorno. Primeiro as velocidades no contorno são ajustadas de forma a satisfazer a condição de compatibilidade contínua. Em seguida, essas velocidades são ajustadas ligeiramente, multiplicando seus valores por uma constante próxima de um. Esta constante é ajustada numericamente por meio de um método de bisseção, até que o pico de erro se aproxime de zero.

3.4 Resultados

3.4.1 Método das soluções manufaturadas

No presente trabalho, foi utilizado o método das soluções manufaturadas, do inglês *method of manufactured solutions* (MMS), para calcular as ordens de convergência espacial e temporal do esquema desenvolvido.

Segundo [62] a ideia principal do MMS é produzir uma solução exata sem estar interessado na realidade física do problema. Desta forma, um conjunto de procedimentos desenvolvidos para encontrar erros que afetam a discretização numérica pode ser aplicado. O MMS combinado com um teste de convergência é recomendado para fornecer a ordem de precisão do método. Para usar o MMS, primeiramente, são admitidas soluções para as equações envolvidas e posteriormente, essas soluções são substituídas nas equações diferenciais que modelam o problema obtendo-se termos fontes.

A construção de soluções manufaturadas exige alguns cuidados, [63] apresenta alguns detalhes na construção delas:

1. utilizar funções suaves como polinômios, funções trigonométricas ou exponenciais, de modo que sejam facilmente calculadas;
2. a solução deve ser o mais geral possível em todos os termos da equação governante;
3. a solução deve ser não trivial de forma que suas derivadas não sejam constantes;
4. soluções manufaturadas não devem crescer exponencialmente com o tempo para evitar instabilidades numéricas;
5. não necessitam de rigor físico, mas deve-se tomar algum cuidado para que as equações a serem resolvidas façam sentido e o código funcione de forma robusta.

No caso das equações de Navier-Stokes, dadas as soluções manufaturadas $U(x, y, t)$, $V(x, y, t)$ e

$P(x, y, t)$ os termos fontes S_u , S_v e S_p podem ser calculados de:

$$\frac{\partial U}{\partial t} + \frac{\partial(U^2)}{\partial x} + \frac{\partial(VU)}{\partial y} + \frac{\partial P}{\partial x} - \nu \left(\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} \right) = S_u, \quad (3.28a)$$

$$\frac{\partial V}{\partial t} + \frac{\partial(UV)}{\partial x} + \frac{\partial(V^2)}{\partial y} + \frac{\partial P}{\partial y} - \nu \left(\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} \right) = S_v, \quad (3.28b)$$

$$\frac{\partial U}{\partial x} + \frac{\partial V}{\partial y} = S_p. \quad (3.28c)$$

Note que a manipulação das equações para obtenção dos termos fontes pode não ser simples. Assim, o uso de códigos de manipulação simbólica pode ser útil. Neste trabalho, foi utilizado o *software* Maple, que possui o atrativo de exportar as expressões dos termos fontes para uma linguagem de programação, neste caso, C.

O próximo passo consiste em testar o código. O teste de convergência é feito utilizando o cálculo do erro nas normas L_∞ , L_1 e L_2 ,

$$L_1 = \frac{1}{N} \left(\sum_{i=1}^N E_i \right), \quad L_2 = \sqrt{\frac{1}{N} \left(\sum_{i=1}^N E_i^2 \right)}, \quad L_\infty = \max_i (E_i), \quad (3.29)$$

onde $E_i = |f_{\text{numérica}} - f_{\text{referência}}|$ e N é a quantidade de pontos no domínio.

O cálculo da ordem é obtido da expressão

$$\frac{\text{erro}(h_1)}{\text{erro}(h_2)} = \left(\frac{h_1}{h_2} \right)^p, \quad (3.30)$$

ou então

$$p = \frac{\log \left(\frac{\text{erro}(h_1)}{\text{erro}(h_2)} \right)}{\log \left(\frac{h_1}{h_2} \right)}. \quad (3.31)$$

onde h representa o espaçamento da malha ou o passo de tempo e *erro* é calculado em uma das normas L_∞ , L_1 , L_2 . No teste de convergência espacial, o erro é calculado como a diferença entre a solução numérica encontrada e a solução analítica do MMS. O teste de convergência temporal é mais difícil, pois os erros espaciais dominam o problema, em particular, quando uma discretização temporal explícita é usada, pois o passo de tempo tem que ser muito pequeno. Nesse caso, uma solução numérica é calculada com o passo de tempo bem pequeno e o erro é calculado em relação a essa solução de referência.

A seguir são apresentados os resultados usando projeção exata e projeção aproximada. Neste trabalho, ambos os termos, convectivo e viscoso, serão aproximados explicitamente. Isso se faz necessário, principalmente na combinação de discretização espacial por diferenças finitas compactas com discretização temporal implícita, pois tem complexidade muito alta. Em todas as simulações

utilizou-se o método de Adams-Bashforth de segunda ordem (Seção 2.1) para evolução temporal. Para a discretização espacial, utilizou-se o método de diferenças finitas compactas de quarta ordem (Seção 2.2.2).

3.4.2 Escoamento forçado em uma cavidade

Na solução manufaturada proposta por [64], os termos fontes são

$$S_p = 0, \quad (3.32a)$$

$$S_u = 0, \quad (3.32b)$$

$$S_v = -8\nu[24F(x) + 2f^{(1)}(x)g^{(2)}(y) + f^{(3)}(x)g(y)] - 64[F_2(x)G_1(y) - g(y)g^{(1)}(y)F_1(x)], \quad (3.32c)$$

onde

$$f(x) = x^4 - 2x^3 + x^2,$$

$$g(y) = y^4 - y^2,$$

$$F(x) = \int f(x)dx = 0.2x^5 - 0.5x^4 + x^3/3,$$

$$F_1(x) = f(x)f^{(2)}(x) - [f^{(1)}(x)]^2 = -4x^6 + 12x^5 - 14x^4 + 8x^3 - 2x^2,$$

$$F_2(x) = \int f(x)f^{(1)}(x)dx = 0.5[f(x)]^2,$$

$$G_1(y) = g(y)g^{(3)}(y) - g^{(1)}(y)g^{(2)}(y) = -24y^5 + 8y^3 - 4y.$$

A solução manufaturada correspondente é

$$U(x, y) = 8f(x)g^{(1)}(y) = 8(x^4 - 2x^3 + x^2)(4y^3 - 2y), \quad (3.33a)$$

$$V(x, y) = -8f^{(1)}(x)g(y) = -8(4x^3 - 6x^2 + 2x)(y^4 - y^2), \quad (3.33b)$$

$$P(x, y) = 8\nu[F(x)g^{(3)}(y) + f^{(1)}(x)g^{(1)}(y)] + 64F_2(x)\{g(y)g^{(2)}(y) - [g^{(1)}(y)]^2\}. \quad (3.33c)$$

O domínio é um quadrado unitário com condições de contorno do tipo Dirichlet.

Para avaliar o código, a simulação foi executada até atingir o estado estacionário. Isso efetivamente isola erros espaciais de erros temporais. A simulação é considerada em estado estacionário quando a diferença entre os campos de velocidades sucessivos é menor do que uma dada tolerância absoluta, neste caso, 10^{-10} . Os resultados são apresentados na Figura 3.1. A viscosidade cinemática usada é $\nu = 10^{-2}$, o passo de tempo é $\Delta t = 10^{-4}$ e as malhas foram 8×8 , 16×16 , 32×32 , 64×64 e 128×128 . O erro é calculado como a diferença entre a solução numérica e a analítica na norma L_∞ . Para evitar repetições, as seguintes notações serão usadas

	Método	Sigla
Método de projeção livre de pressão e projeção exata	LP-E	
Método de projeção livre de pressão e projeção aproximada	LP-A	
Método de projeção com pressão incremental e projeção aproximada	PI-A	

Observa-se na Figura 3.1a que no LP-A o refinamento da malha deixa de produzir ganho de precisão a partir de um certo nível de refinamento. Com a redução de Δt , Figura 3.1b, há um ganho de precisão para malhas mais finas no estado estacionário. Isso evidencia que esse método introduz um erro proporcional a Δt e que esse erro está presente inclusive no estado estacionário. Esse resultado é esperado [54], pois as condições de contorno usadas para a velocidade intermediária $\mathbf{u}^*$ são as mesmas da velocidade $\mathbf{u}$. As condições de contorno propostas em [14], que incluem a pressão, poderiam reduzir esse erro, mas não foram testadas. Alternativamente, incluiu-se uma estimativa da pressão na equação do momento, que corresponde ao PI-A. Como pode ser visto na Figura 3.1c, o erro proporcional a Δt não é perceptível com o PI-A e quarta ordem foi observada em todas as malhas testadas.

O LP-E apresenta uma característica interessante, que pode ser observado na Figura 3.1d. O método apresenta quarta ordem de convergência para todas as malhas testadas e não é evidente que o erro é proporcional a Δt . Isso ocorre sem a necessidade de correção nas condições de contorno para $\mathbf{u}^*$, que são as mesmas de $\mathbf{u}$, e sem a necessidade de introduzir uma estimativa para a pressão na equação do momento. A total ausência da pressão torna esse método verdadeiramente livre de pressão. Isso é interessante, pois elimina o acúmulo de erros no gradiente da pressão [54].

A Figura 3.2 mostra a evolução do máximo do divergente discreto. No LP-A, o erro proporcional a Δt afeta também o divergente. No PI-A, o divergente tem erro menor e tende a melhorar à medida que se aproxima do estado estacionário. No entanto, o menor erro para o divergente é obtido com o LP-E. Por se tratar de um método de projeção exata, espera-se um divergente exatamente igual a zero, limitado apenas pela precisão da máquina. O observado na Figura 3.2 está de acordo com essa expectativa.

3.4.3 Vórtices de Taylor-Green

Os vórtices de Taylor-Green são um problema transiente geralmente usado para medir a ordem temporal de métodos numéricos [20, 65]. Nele, nenhum termo forçante é usado e a solução analítica é

$$U(x, y, t) = -\exp(-2\pi^2 t\nu) \sin(\pi x) \cos(\pi y), \quad (3.34a)$$

$$V(x, y, t) = \exp(-2\pi^2 t\nu) \cos(\pi x) \sin(\pi y), \quad (3.34b)$$

$$P(x, y, t) = \frac{1}{4} \exp(-4\pi^2 t\nu) (\cos(2\pi x) + \cos(2\pi y)). \quad (3.34c)$$

Figura 3.1: Teste de convergência espacial baseado na solução manufaturada para o problema do escoamento forçado em uma cavidade. A simulação foi desenvolvida até o estado estacionário ser atingido. O erro é calculado como a diferença entre a solução numérica e a solução manufaturada na norma L_∞ . A linha contínua representa quarta ordem de convergência. A viscosidade cinemática usada é $\nu = 10^{-2}$, o passo de tempo é $\Delta t = 10^{-4}$ e as malhas foram 8×8 , 16×16 , 32×32 , 64×64 e 128×128 . $\Delta x = \Delta y = h$.

A simulação é realizada em um domínio quadrado $\frac{1}{4} \leq x, y \leq 2\frac{1}{4}$ com condições de contorno do tipo Dirichlet dependentes do tempo. O domínio é deslocado de $\frac{1}{4}$ para evitar condições de contorno triviais. Testes de precisão são realizados com uma viscosidade cinemática fixa em $\nu = 10^{-2}$. O erro é medido no tempo $t = 1$. A malha é fixa em 20×20 e o passo de tempo varia de modo que a precisão temporal possa ser investigada.

Figura 3.2: Evolução do máximo do divergente discreto numa malha fixa 8×8 com $\Delta t = 10^{-4}$ e $\nu = 10^{-2}$.

Medir a precisão temporal não é tão simples como medir a precisão espacial. A combinação de uma discretização temporal explícita com a rigidez das equações de Navier-Stokes para baixos Reynolds impõem uma restrição severa no passo de tempo máximo permitido. Portanto, erros espaciais frequentemente dominam os temporais. Uma maneira de contornar esse problema é executar uma simulação com Δt pequeno e assumir a solução resultante como uma solução de referência, com a qual outras soluções com passos de tempo maiores serão comparadas.

A Figura 3.3 mostra os resultados do teste de convergência temporal. Os erros são calculados em relação à uma solução de referência com $\Delta t = 10^{-4}$. Para os esquemas PI-A e LP-E, foi obtida segunda ordem de precisão, não só para o campo de velocidade, mas também para a pressão. Os trabalhos na literatura consultada sobre métodos de projeção reportam primeira ordem para a pressão ou não reportam a ordem encontrada. Para o esquema LP-A, a adoção de condições de contorno simplificadas para $\mathbf{u}^*$ reduzem a ordem do método para primeira ordem.

O esquema LP-E possui a interessante característica de produzir segunda ordem de convergência, sem a necessidade de se incluir uma aproximação para o gradiente da pressão na equação do momento e sem a necessidade de corrigir as condições de contorno para $\mathbf{u}^*$. Visto que não foi encontrado na literatura método de projeção com as características do LP-E, esse método foi publicado no Journal Computer & Fluids [39]. Os próximos resultados utilizam apenas o LP-E.

3.4.4 Escoamento em uma cavidade

O escoamento em uma cavidade é um dos problemas mais populares usados para verificar a confiabilidade dos esquemas numéricos para Navier-Stokes. Muitas soluções numéricas de referência estão disponíveis na literatura [20, 66, 67]. As equações de Navier-Stokes são resolvidas em um quadrado unitário, onde a parede superior se move à velocidade uniforme tangencial ($u = 1, v = 0$). Condições

Figura 3.3: Convergência temporal para o problema vórtices de Taylor-Green. A simulação foi executada até $t = 1$. A viscosidade cinemática usada é $\nu = 10^{-2}$. Os erros são calculados na norma L_∞ . Os erros são calculados em relação à solução de referência com $\Delta t = 10^{-4}$. A malha é fixada em 20×20 . Em (a) e (c) a linha contínua representa segunda ordem enquanto que em (b) primeira ordem.

de contorno de não-escorregamento ($u = 0, v = 0$) são aplicadas nas outras paredes.

Os resultados deste estudo foram comparados com os obtidos por [66], na Figura 3.4. Ela mostra um corte do perfil de velocidade vertical v em $y = 0.5$. No mesmo gráfico, se observa um corte do perfil de velocidade horizontal u em $x = 0.5$. As viscosidades cinemáticas adotadas foram $\nu = 10^{-2}$, $\nu = 2.5 \times 10^{-3}$ e $\nu = 10^{-3}$ e a malha possui 130×130 células. Observa-se uma boa concordância entre os resultados. A Figura 3.5 mostra os contornos da função corrente e pressão para $\nu = 10^{-3}$.

Figura 3.4: Perfis de velocidade no estado estacionário para o problema da cavidade: corte do perfil de velocidade vertical v em $y = 0.5$; corte do perfil de velocidade horizontal u em $x = 0.5$. Viscosidade cinemática $\nu = 10^{-2}$, $\nu = 2.5 \times 10^{-3}$ e $\nu = 10^{-3}$, respectivamente, em uma malha 130×130 .

3.4.5 Escoamento em um degrau

Outro problema popular é o escoamento em um degrau (*backward-facing step*). Este problema é particularmente interessante, pois envolve velocidades normais não-nulas no contorno. Serve para testar a validade do procedimento de renormalização descrito na Seção 3.3. A geometria do domínio é mostrada na figura 3.6.

As condições de contorno usadas neste trabalho foram:

- Placa inferior: $u = 0$, $v = 0$ em $y = 0$, $0 \leq x \leq 10$

Figura 3.5: Contornos da função corrente e da pressão para o problema da cavidade. Malha 130×130 e $\nu = 10^{-3}$.

Figura 3.6: Escoamento sobre um degrau: geometria.

- Placa superior: $u = 0, v = 0$ em $y = 1, 0 \leq x \leq 10$
- Inflow: $24(y - 0.5)(1 - y), v = 0$ em $x = 0, 0.5 \leq y \leq 1$
- Degrau: $u = 0, v = 0$ em $x = 0, 0 \leq y \leq 0.5$
- Outflow: $3y(1 - y), v = 0$, em $x = 10, 0 \leq y \leq 1$

A Figura 3.7 mostra os contornos da função corrente, da componente u da velocidade e da pressão para $\nu = 5 \times 10^{-3}$. A Tabela 3.1 compara, o comprimento da região de recirculação, o comprimento do vórtice primário e a localização com os resultados de outros autores. Novamente, os resultados obtidos estão em concordância com a literatura.

Figura 3.7: Contornos da função corrente, da componente u da velocidade e da pressão. Malha 300×50 , $\nu = 5 \times 10^{-3}$.

Tabela 3.1: Comparação dos resultados para o escoamento em um degrau, comprimento da região de recirculação, comprimento do vórtice primário e localização, para $\nu = 5 \times 10^{-3}$.

	Comprimento de recirculação x_r	$\psi_{\min}$	localização $\psi_{\min}$
Barragy [68]	2.67	-0.0331	(1.0021,-0.2030)
Chinchapatnam et al. [69]	2.72	-0.0315	(1.3330,-0.2167)
Bourantas et al. [70]	2.64	-0.0331	(1.0001,-0.2000)
Loukopoulos et al. [67]	2.66	-0.0330	(1.0011,-0.2070)
LP-E 200×20	2.62	-0.0328	(0.9926,-0.2032)
LP-E 400×40	2.64	-0.0330	(0.9919,-0.2010)
LP-E 600×60	2.65	-0.0331	(0.9920,-0.2006)
LP-E 800×80	2.65	-0.0331	(0.9920,-0.1992)
LP-E 1000×100	2.66	-0.0331	(0.9920,-0.1994)
LP-E 2000×200	2.66	-0.0331	(0.9920,-0.1993)

Métodos acoplados e discretização temporal

Os métodos segregados para Navier-Stokes têm algumas desvantagens, como a necessidade de condições de contorno numéricas para a pressão e velocidade intermediária e também a dificuldade em se obter altas ordens temporais. A segregação temporal pode inclusive introduzir erros em soluções estacionárias [57].

Uma alternativa são os métodos acoplados, que resolvem todas as equações juntas e não se utilizam de equações de correção para as velocidades. Os métodos acoplados apresentam vantagem em relação aos métodos segregados quando o escoamento é complexo, especialmente pela sua robustez. Porém, exigem capacidade de processamento e armazenamento de memória maior. Isso fez com que esses métodos fossem pouco utilizados no passado. Mas, com o avanço no desempenho dos computadores, esses métodos são cada vez mais utilizados.

Métodos acoplados, em geral, podem atingir altas ordens temporais e não requerem condições de contorno artificiais.

Nesse capítulo um estudo sobre a estabilidade de métodos numéricos para a discretização temporal é apresentado. A discretização temporal no método acoplado é feita utilizando os métodos ARK (do inglês *Additive Runge-Kutta*). Esses dividem a equação em várias partes e aplicam um método de Runge-Kutta (mais adequado) para cada termo. No próximo capítulo, a esse método é incorporado um método de fronteira imersa de segunda ordem.

4.1 Estabilidade

Segundo Lambert [71], uma equação diferencial ordinária da forma

$$\frac{du}{dt} = f(t, u), \quad (4.1)$$

é dita ser rígida se

$$(i) \operatorname{Re}(\lambda_\xi) < 0,$$

$$(ii) \max_{\xi} |\operatorname{Re}(\lambda_\xi)| \gg \min_{\xi} |\operatorname{Re}(\lambda_\xi)|,$$

onde λ_ξ são os autovalores da matriz $A = \frac{\partial f}{\partial u}$. Métodos numéricos têm dificuldades em tratar esse tipo de problema, principalmente os métodos explícitos, pois o tamanho do passo temporal necessário para a estabilidade do método é determinado pela componente da solução que varia mais rapidamente, ou seja, $\max_{\xi} |\operatorname{Re}(\lambda_\xi)|$.

Um exemplo simples de equação rígida, comumente usada como caso teste para métodos numéricos, é

$$\frac{du}{dt} = ku, \quad (4.2)$$

onde $\operatorname{Re}(k) < 0$, com condição inicial $u(0) = 1$. A solução exata, $u(t) = e^{kt}$, tende para zero quando $t \rightarrow \infty$.

Se um método numérico aplicado em (4.2) produzir uma solução

$$u_{n+1} = \phi(\Delta tk)u_n, \quad (4.3)$$

que também tende para zero quando o número de passos tende para infinito, para um passo de tempo de qualquer tamanho finito, esse método é dito ser A -estável. Essa condição é equivalente a $|\phi(\Delta tk)| < 1$ [75]. A função ϕ será chamada de função de estabilidade. O método será L -estável se for A -estável e a solução tender a zero em um único passo de tempo, com o passo de tempo tendendo a infinito [53].

Nas equações de Navier-Stokes o termo viscoso é responsável pela característica rígida dessas equações, uma vez que esse termo sofre a influência do número de Reynolds, que é pequeno neste estudo. Por isso, é fundamental buscar métodos que sejam L -estáveis ou, pelo menos, A -estáveis. Caso contrário, o método pode restringir drasticamente o passo de tempo. Isso é ainda mais importante quando se usam métodos acoplados, pois o custo de cada passo de tempo é alto.

Métodos explícitos nunca são A -estáveis [76, 77]. O método de Euler implícito é L -estável, mas é de primeira ordem. O método de Crank-Nicolson, que é de segunda ordem, é A -estável, mas não é L -estável. Por outro lado, métodos implícitos de Runge-Kutta podem ser L -estáveis e possuir ordens altas (> 2) [78]. Na próxima seção será feito um estudo da estabilidade desses métodos.

Outra característica interessante em um método numérico é a possibilidade de se ajustar o passo de tempo automaticamente, controlando o erro. Isso é possível com métodos de Runge-Kutta embutidos, como, por exemplo, o método de Runge-Kutta-Fehlberg 4(5) [79]. Nele, duas soluções com ordens diferentes são calculadas e comparadas, permitindo uma estimativa do erro. O método é dito embutido pois os termos intermediários usados em umas das soluções são reaproveitados na outra. Outros métodos de passo de tempo adaptativo são encontrados em [80, 81, 82, 83, 84].

4.2 Análise da estabilidade

Uma maneira de verificar se um método é A -estável é usando a função de estabilidade ϕ [71]. Um método é dito ser A -estável, se sua região de estabilidade absoluta

$$\{z \in \mathbb{C}/, |\phi(z)| < 1\}, \quad (4.4)$$

contiver a metade negativa (esquerda) do plano complexo, e.g. $\{z \in \mathbb{C}/\text{Re}(z) < 0\}$. Um método é L -estável se for A -estável e $|\phi(z)| \rightarrow 0$ quando $z \rightarrow \infty$.

4.2.1 Euler explícito

O método de Euler explícito aplicado a (4.2) produz

$$u_{n+1} = u_n + \Delta t f(t_n, u_n), \quad (4.5)$$

$$u_{n+1} = u_n + \Delta t (ku_n), \quad (4.6)$$

$$u_{n+1} = u_n(1 + \Delta tk), \quad (4.7)$$

ou seja, $u_n = u_0(1 + \Delta tk)^n$ e portanto $\phi(z) = 1 + z$. A região de estabilidade absoluta para esse método é

$$\{z \in \mathbb{C}/|1 + z| < 1\}, \quad (4.8)$$

que corresponde à região representada na Figura 4.1. Portanto, o método de Euler explícito não é A -estável.

4.2.2 Euler implícito

O método de Euler implícito aplicado a (4.2) produz

$$u_{n+1} = u_n + \Delta t f(t_{n+1}, u_{n+1}), \quad (4.9)$$

$$u_{n+1} = u_n + \Delta t (ku_{n+1}), \quad (4.10)$$

$$u_{n+1} = \frac{u_n}{1 - \Delta tk}. \quad (4.11)$$

Figura 4.1: Região de estabilidade do método de Euler explícito.

ou seja, $u_n = u_0 \left(\frac{1}{1 - \Delta t k} \right)^n$ e portanto $\phi(z) = \frac{1}{1 - z}$. A região de estabilidade absoluta para esse método é

$$\left\{ z \in \mathbb{C} / \left| \frac{1}{1 - z} \right| < 1 \right\}, \quad (4.12)$$

que corresponde à região representada na Figura 4.2. Portanto, o método de Euler implícito é A -estável.

Figura 4.2: Região de estabilidade do método de Euler implícito.

Além de ser A -estável, o método de Euler implícito é L -estável, pois $\left| \frac{1}{1 - z} \right| \rightarrow 0$ quando $z \rightarrow \infty$.

4.2.3 Crank-Nicolson

O método de Crank-Nicolson aplicado a (4.2) produz

$$u_{n+1} = u_n + \frac{1}{2}\Delta t (f(t_n, u_n) + f(t_{n+1}, u_{n+1})), \quad (4.13)$$

$$u_{n+1} = u_n + \frac{1}{2}\Delta t (ku_n + ku_{n+1}), \quad (4.14)$$

$$u_{n+1} = u_n \left(\frac{1 + \frac{1}{2}hk}{1 - \frac{1}{2}hk} \right). \quad (4.15)$$

A função de estabilidade é

$$\phi(z) = \frac{1 + \frac{1}{2}z}{1 - \frac{1}{2}z}, \quad (4.16)$$

que corresponde à região de estabilidade absoluta

$$\left\{ z \in \mathbb{C} / \left| \frac{1 + \frac{1}{2}z}{1 - \frac{1}{2}z} \right| < 1 \right\}, \quad (4.17)$$

representada na Figura 4.3.

Figura 4.3: Região de estabilidade do método Crank-Nicolson.

O método de Crank-Nicolson é A -estável, mas não é L -estável, pois $\left| \frac{1 + \frac{1}{2}z}{1 - \frac{1}{2}z} \right| \rightarrow 1$ quando $z \rightarrow \infty$.

4.3 Additive Runge-Kutta - ARK

Seja a EDO

$$\frac{du}{dt} = F(u) = \sum_{\nu=1}^N F^{[\nu]}(u), \quad (4.18)$$

onde $F(u)$ pode ser decomposta em N termos. Os métodos ARK (do inglês *Additive Runge-Kutta*) são construídos de forma a permitir que se aplique um método de Runge-Kutta diferente para cada termo [85]. Assim, é possível aplicar a cada termo o método de Runge-Kutta mais apropriado.

Segundo [86], os métodos ARK_N , aplicados a equações da forma (4.18), podem ser generalizados como:

$$U^{(i)} = U^{(n)} + (\Delta t) \sum_{\nu=1}^N \sum_{j=1}^s a_{ij}^{[\nu]} F^{[\nu]}(U^{(j)}), \quad (4.19a)$$

$$U^{(n+1)} = U^{(n)} + (\Delta t) \sum_{\nu=1}^N \sum_{i=1}^s b_i^{[\nu]} F^{[\nu]}(U^{(i)}), \quad (4.19b)$$

$$\hat{U}^{(n+1)} = U^{(n)} + (\Delta t) \sum_{\nu=1}^N \sum_{i=1}^s \hat{b}_i^{[\nu]} F^{[\nu]}(U^{(i)}). \quad (4.19c)$$

Cada um dos N termos é integrado pelo seu próprio método de Runge-Kutta de s -estágios. Além disso, $U^{(n)} \approx u(t^n)$, $U^{(i)} \approx u(t^n + c_i \Delta t)$ é o valor do vetor U no estágio i , e $U^{(n+1)} \approx u(t^n + \Delta t)$. Ambos $U^{(n)}$ e $U^{(n+1)}$ são aproximações de ordem q . O valor aproximado de U no método embutido, $\hat{U}^{(n+1)}$, é de ordem p . Os coeficientes de Butcher $a_{ij}^{[\nu]}$, $b_i^{[\nu]}$, $\hat{b}_i^{[\nu]}$, $\nu = 1, 2, \dots, N$ são restritos pelas condições de ordem de precisão e estabilidade.

Um método ARK bastante popular é o ARK_2 ($N = 2$) para o qual a equação (4.18) toma a forma

$$\frac{du}{dt} = F_{ns} + F_s, \quad (4.20)$$

com F_{ns} representando os termos não-rígidos e F_s representando os termos rígidos. Métodos de Runge-Kutta explícitos são geralmente utilizados para integrar os termos não-rígidos, enquanto os termos rígidos são tratados por métodos implícitos de Runge-Kutta. Métodos que combinam tratamentos implícitos e explícitos são conhecidos como IMEX. Essa divisão permite, por exemplo, discretizar os termos rígidos com métodos L -estáveis e ao mesmo tempo discretizar os termos não-lineares de forma explícita, pois, o custo da discretização implícita de termos não-lineares é alta.

Nesse trabalho, adotamos o método ARK3(2)4L[2]SA [78], que foi desenvolvido especialmente para resolver equações de convecção-difusão-reação. A sigla significa que se trata de um método ARK_2 IMEX, de terceira ordem no método principal, segunda ordem no método embutido, 4 estágios, L -estável (parte implícita), ordem de estágio 2, preciso para equações rígidas (*stiff accurate*) e usa um método da família ESDIRK na parte implícita.

A ordem de cada estágio [1] de um método de s -estágios é definida como o maior inteiro $\tilde{p}$ para o

qual os coeficientes de Butcher satisfazem

$$\sum_{j=1}^s b_j c_j^{k-1} = \frac{1}{k}, \quad (1 \leq k \leq \tilde{p}), \quad (4.21a)$$

$$\sum_{j=1}^s a_{ij} c_j^{k-1} = \frac{1}{k} c_i^k, \quad (1 \leq k \leq \tilde{p}, 1 \leq i \leq s). \quad (4.21b)$$

A ordem de estágio do método como um todo é definida como a menor das ordens de cada estágio. Em problemas rígidos, os métodos de Runge-Kutta podem não alcançar a ordem do método. Nesse caso, a ordem é limitada pela ordem de estágio. Isso quer dizer que, apesar do método ARK3(2)4L[2]SA ser de terceira ordem, devido a rigidez do problema, é esperado ordem 2.

A sigla ESDIRK significa *explicit, singly diagonally implicit* Runge-Kutta. Para os casos em que a matriz $\mathbf{A}$ dos coeficientes é triangular inferior, os métodos são referidos como *diagonally implicit* Runge-Kutta (DIRK).

$$\mathbf{A} = \begin{bmatrix} \times & & & \\ \times & \times & & \\ \vdots & \ddots & \ddots & \\ \times & \times & \times & \times \end{bmatrix} \quad (4.22)$$

Métodos para os quais $\mathbf{A}$ tem um único autovalor de multiplicidade s são chamados de *singly implicit* Runge-Kutta (SIRK). E no caso em que os métodos são DIRK e SIRK:

$$\mathbf{A} = \begin{bmatrix} \gamma & & & \\ \times & \gamma & & \\ \times & \times & \gamma & \\ \vdots & \ddots & \ddots & \\ \times & \times & \times & \gamma \end{bmatrix} \quad (4.23)$$

estes são chamados de *singly diagonally implicit* Runge-Kutta (SDIRK).

A importância desses casos especiais emerge quando um método de Runge-Kutta implícito de s -estágios é aplicado a um sistema de M EDOs. Isso normalmente requer a solução de um sistema de $s \times M$ equações. Quando o sistema é resultante da semi-discretização de um sistema de EDPs não-lineares, M é tipicamente muito grande e o sistema de EDOs será não-linear, tornando a solução desse sistema muito cara. Usando uma transformação envolvendo a forma de Jordan de $\mathbf{A}$, o custo pode ser reduzido [87]. Essa transformação é eficiente para métodos SIRK, pois $\mathbf{A}$ tem apenas um autovalor distinto, de modo que a fatoração da matriz pode ser reutilizada. Por outro lado, nos métodos DIRK, onde $\mathbf{A}$ é triangular inferior, a solução pode ser implementada de forma eficiente sem a transformação para a forma de Jordan de $\mathbf{A}$. Os métodos SDIRK incorporam todas essas vantagens, por serem ambos DIRK e SIRK [1]. A relação entre essas classes de métodos de Runge-Kutta é ilustrada na Figura 4.4.

Figura 4.4: Diagrama das classes de métodos de Runge-Kutta: IRK=*implicit*; DIRK=*diagonally implicit*; SIRK=*singly implicit*; SDIRK=*singly diagonally implicit*; ERK=*explicit*. Figura extraída de [1].

A parte implícita do método a ser utilizado nesse trabalho possui o primeiro estágio explícito e outros três implícitos com a seguinte tabela de Butcher

0	0	0	0	0
$\frac{1767732205903}{2027836641118}$	$\frac{1767732205903}{4055673282236}$	$\frac{1767732205903}{4055673282236}$	0	0
$\frac{3}{5}$	$\frac{2746238789719}{10658868560708}$	$\frac{-640167445237}{6845629431997}$	$\frac{1767732205903}{4055673282236}$	0
1	$\frac{1471266399579}{7840856788654}$	$\frac{-4482444167858}{7529755066697}$	$\frac{11266239266428}{11593286722821}$	$\frac{1767732205903}{4055673282236}$
b_i	$\frac{1471266399579}{7840856788654}$	$\frac{-4482444167858}{7529755066697}$	$\frac{11266239266428}{11593286722821}$	$\frac{1767732205903}{4055673282236}$
$\hat{b}_i$	$\frac{2756255671327}{12835298489170}$	$\frac{-10771552573575}{22201958757719}$	$\frac{9247589265047}{10645013368117}$	$\frac{2193209047091}{5459859503100}$

Nota-se que de fato, a matriz dos coeficientes desse método tem a forma dada em (4.23). A existência do estágio explícito requer uma condição inicial. Na prática, isso é obtido ao se realizar um passo de tempo com Δt bem pequeno usando o método de Euler implícito.

A parte explícita do método ARK3(2)4L[2]SA é dada pela tabela de Butcher:

0	0	0	0	0
$\frac{1767732205903}{2027836641118}$	$\frac{1767732205903}{2027836641118}$	0	0	0
$\frac{3}{5}$	$\frac{5535828885825}{10492691773637}$	$\frac{788022342437}{10882634858940}$	0	0
1	$\frac{6485989280629}{16251701735622}$	$\frac{-4246266847089}{9704473918619}$	$\frac{10755448449292}{10357097424841}$	0
b_i	$\frac{1471266399579}{7840856788654}$	$\frac{-4482444167858}{7529755066697}$	$\frac{11266239266428}{11593286722821}$	$\frac{1767732205903}{4055673282236}$
$\hat{b}_i$	$\frac{2756255671327}{12835298489170}$	$\frac{-10771552573575}{22201958757719}$	$\frac{9247589265047}{10645013368117}$	$\frac{2193209047091}{5459859503100}$

4.3.1 Passo de tempo adaptativo

Em 1960, Fehlberg propôs métodos de Runge-Kutta embutidos. Tais métodos utilizam dois esquemas de Runge-Kutta de ordens diferentes (p e $p - 1$) de tal forma que a diferença entre as aproximações produz uma estimativa do erro local, que é então utilizada para ajustar o passo temporal [88].

O método ARK3(2)4L[2]SA utiliza um passo de tempo adaptativo com a estimativa de erro dada por

$$\delta^{(n+1)} = U^{(n+1)} - \hat{U}^{(n+1)}, \quad (4.24)$$

sendo $U^{(n+1)}$ e $\hat{U}^{(n+1)}$ as soluções dadas por (4.19b) e (4.19c).

4.3.2 Implementação

Uma breve descrição do método ARK3(2)4L[2]SA foi apresentada. Mais detalhes podem ser encontrados em [78]. Foi utilizada a implementação fornecida pela biblioteca PETSc [89].

4.3.3 Resolução do sistema não-linear

Neste trabalho, para resolver o sistema não-linear resultante da discretização das equações, utilizou-se o método de Newton implementado pelo PETSc. É necessário passar como parâmetro de entrada, somente o valor do resíduo $\mathbf{R}$, que contém a parte temporal e espacial:

$$\mathbf{R} = \begin{bmatrix} \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p - \nu \nabla^2 \mathbf{u} - \mathbf{g} \\ \nabla \cdot \mathbf{u} \end{bmatrix}. \quad (4.25)$$

4.4 Resultados

A discretização temporal é feita utilizando o método ARK3(2)4L[2]SA (apresentado acima) e a discretização espacial usando o método de diferenças finitas padrão de segunda ordem (Seção 2.2.1). A verificação do código é feita utilizando o método das soluções manufaturadas (Seção 3.4.1).

Sejam as soluções prescritas

$$U(x, y) = -\sin(\pi x) \cos(\pi y), \quad (4.26a)$$

$$V(x, y) = \cos(\pi x) \sin(\pi y), \quad (4.26b)$$

$$P(x, y) = 0.25(\cos(2\pi x) + \cos(2\pi y)), \quad (4.26c)$$

no domínio $0 \leq x, y \leq 1$. Considerando, $\nu = 10^{-2}$ e $t = 100s$, a Tabela 4.1 apresenta a ordem de convergência na norma L_∞ . É observado segunda ordem de convergência tanto para a velocidade quanto para a pressão. Na Figura 4.5 observa-se a evolução do passo de tempo. À medida que se aproxima do estado estacionário, o passo de tempo cresce consideravelmente, o que proporciona um ganho em tempo de simulação.

Tabela 4.1: Erro e ordem de convergência na norma L_∞ .

Malha	u		v		p	
	erro	ordem	erro	ordem	erro	ordem
25×25	1.907773e-03	-	1.907773e-03	-	2.177975e-03	-
50×50	4.854651e-04	1.9745	4.854651e-04	1.9745	5.460536e-04	1.9959
100×100	1.223869e-04	1.9879	1.223869e-04	1.9879	1.364656e-04	2.0005

Figura 4.5: Evolução do passo de tempo adaptativo durante a simulação da solução manufaturada (4.26) usando o método ARK3(2)4L[2]SA.

Método de fronteira imersa para as Equações de Navier-Stokes

O método de fronteira imersa utilizado segue em parte as ideias presentes em [90, 91, 92]. A Figura 5.1 ilustra um domínio com uma interface imersa. A interface é representada pela conhecida técnica *level set*, onde a localização da interface é um conjunto de nível (e.g. curva de nível ou superfície de nível) de uma função ϕ . Em geral, a função escolhida é a função distância (com sinal) até a interface, o que faz com que a localização da interface coincida com o nível zero da função ϕ . A região onde $\phi > 0$ é a região de interesse, ou seja, a região onde há fluido. Em $\phi < 0$, a solução é considerada nula, por se tratar de contorno rígido.

Figura 5.1: Ilustração de um domínio com interface imersa. Células sólidas são marcadas com S, e o domínio de fluido, com F.

A função ϕ é amostrada no centro das células do domínio. Cada uma das células é marcada de acordo com o respectivo valor de ϕ . Células com $\phi < 0$ são tratadas como células de contorno rígido, enquanto que as demais são células de fluido. Valores de ϕ nas arestas ou cantos das células são obtidos por interpolação linear. Se ϕ , ou sua interpolação, for nulo, ele é perturbado de forma que fique positivo ou negativo, mas muito pequeno em módulo.

As velocidades u e v situadas em arestas adjacentes a pelo menos uma célula de contorno são consideradas condições de contorno. Seu valor é obtido por meio da interpolação dos valores internos ao domínio e do valor na interface. As equações de Navier-Stokes são resolvidas apenas nas arestas entre duas células de fluido. O divergente da velocidade, presente na equação da continuidade, assim como a pressão, são calculados apenas no centro das células de fluido.

A Figura 5.2 e as equações (5.1a) e (5.1b) correspondem ao cálculo da velocidade normal ao contorno.

Figura 5.2: Direção normal.

$$d = 0.5 + \frac{\phi_{i,j}}{\phi_{i-1,j} - \phi_{i,j}}, \quad (5.1a)$$

$$u_{i-1/2,j} = \frac{u_I - du_{i+1/2,j}}{1 - d}. \quad (5.1b)$$

A Figura 5.3 e as equações (5.2a), (5.2b) e (5.2c) correspondem ao cálculo da velocidade tangencial ao contorno.

$$\phi_{i+1/2,j} = \frac{\phi_{i,j} + \phi_{i+1,j}}{2}, \quad (5.2a)$$

$$d = \frac{\phi_{i+1/2,j}}{\phi_{i+1/2,j} - \phi_{i+1/2,j+1}}, \quad (5.2b)$$

$$u_{i+1/2,j+1} = \frac{u_I + (d - 1)u_{i+1/2,j}}{d}. \quad (5.2c)$$

No caso normal $-1/2 < d < 1/2$ enquanto que no caso tangencial $0 < d < 1$. Para evitar problemas

Figura 5.3: Direção tangencial.

de divisão por zero, no caso tangencial, quando $d < 10^{-6}$, impõe-se $d = 10^{-6}$.

5.1 Condição de compatibilidade

A condição de compatibilidade derivada da equação da continuidade estabelece uma condição necessária a ser satisfeita pela solução quando se tem apenas condições de contorno do tipo Dirichlet. Partindo de

$$\nabla \cdot \mathbf{u} = 0, \quad (5.3)$$

integrando no domínio Ω

$$\int_{\Omega} (\nabla \cdot \mathbf{u}) dV = 0, \quad (5.4)$$

e, usando o teorema da divergência de Gauss, deduz-se a condição de compatibilidade

$$\oint_{\partial\Omega} (\mathbf{u} \cdot \mathbf{n}) dS = 0. \quad (5.5)$$

No caso discreto, a soma, ao longo das bordas, das componentes u e v da velocidade entrando no domínio (*inflow*) tem que ser igual, em módulo, à soma das velocidades que saem do domínio (*outflow*). Uma pequena diferença entre *inflow* e *outflow* pode prejudicar a ordem do método.

Mesmo que a velocidade ao longo da interface satisfaça a condição de compatibilidade, as velocidades interpoladas para as bordas do domínio computacional podem não satisfazê-la com precisão suficiente. Para sanar esse problema, adotou-se uma técnica de regularização que visa corrigir levemente a velocidade de *outflow* e garantir a condição de compatibilidade discreta.

Primeiramente, as velocidades normais são interpoladas sem nenhuma correção. Calcula-se a soma do módulo das velocidades *inflow* e *outflow* separadamente. A seguir, as velocidades de *outflow* são multiplicadas pelo fator

$$f = \frac{\text{inflow}}{\text{outflow}}. \quad (5.6)$$

Como a diferença entre *inflow* e *outflow* é muito pequena, no código implementado, os somatórios

e fator são calculados com tipos de dados de precisão maior (80 *bits*) do que o usado nas velocidades (64 *bits*).

5.2 *Outflow* continuativo

A condição de contorno do tipo *outflow* continuativo é um artifício numérico para simular os efeitos de um domínio infinitamente longo. Quer-se que a borda do domínio computacional deixe passar informação sem influenciar o interior do domínio. Uma condição simples, mas eficiente, é tomar uma condição de contorno do tipo Neumann homogênea [93]

$$\frac{\partial \mathbf{u}}{\partial \mathbf{n}} = 0 \Rightarrow \begin{cases} \frac{\partial u}{\partial x} = 0, \\ \frac{\partial v}{\partial y} = 0. \end{cases} \quad (5.7)$$

5.3 Resultados

5.3.1 Verificação com solução manufaturada

Sejam as soluções prescritas

$$U(x, y) = -\sin(\pi x) \cos(\pi y), \quad (5.8a)$$

$$V(x, y) = \cos(\pi x) \sin(\pi y), \quad (5.8b)$$

$$P(x, y) = 0.25(\cos(2\pi x) + \cos(2\pi y)), \quad (5.8c)$$

no domínio $0 \leq x, y \leq 1$.

Para a evolução temporal, foi usado o método ARK3(2)4L[2]SA descrito na Seção 4.3. A discretização espacial foi feita por meio do método de diferenças finitas padrão de segunda ordem (Seção 2.2.1). Foram considerados $\nu = 10^{-2}$ e $t = 100s$. A Figura 5.4 ilustra os domínios considerados.

A Figura 5.5 mostra o passo de tempo adaptativo ao usar o método ARK3(2)4L[2]SA para diversas malhas para um domínio sem interface imersa. Nota-se que o refinamento da malha não interfere no passo de tempo, o mesmo tem um comportamento semelhante em todas as simulações.

A Tabela 5.1 mostra a ordem de convergência na norma L_∞ para um domínio sem interface imersa, ou seja, o contorno coincide com a malha computacional. Na Tabela 5.2, uma interface imersa de forma circular foi adicionada ao domínio. Na Tabela 5.3, o quadrado que delimita o domínio é formado por uma interface imersa. Apenas no caso do círculo, não foi obtida ordem 2. Isso pode ser justificado pelo fato de a distância máxima da interface imersa até o ponto de malha não se reduzir na mesma razão do refinamento da malha.

As figuras 5.6a e 5.6b mostram o perfil de velocidade em $t = 100s$ na simulação dos escoamentos: uma interface imersa de forma circular foi adicionada ao domínio e o quadrado que delimita o

(a) Domínio sem interface.

(b) Uma interface imersa de forma circular foi adicionada ao domínio.

(c) O quadrado que delimita o domínio é formado por uma interface imersa.

Figura 5.4: Domínios para a simulação da solução manufaturada 5.8.

Tabela 5.1: Erro e ordem de convergência na norma L_∞ . Domínio sem interface imersa (Figura 5.4a).

Malha	u		v		p	
	erro	ordem	erro	ordem	erro	ordem
25×25	1.907479e-03	-	1.907479e-03	-	9.257572e-04	-
50×50	4.854515e-04	1.9743	4.854515e-04	1.9743	2.322346e-04	1.9951
100×100	1.223893e-04	1.9878	1.223893e-04	1.9878	5.819018e-05	1.9967
150×150	5.454259e-05	1.9933	5.454259e-05	1.9933	2.587593e-05	1.9987

domínio é formado por uma interface imersa. Em 5.7a e 5.7b são mostrados o divergente sem e com a correção dada por (5.6). Observa-se que o erro, onde a pressão é prescrita, é um ponto isolado, os pontos vizinhos não são afetados, e esse desaparece quando se faz a correção, ou seja, a condição de compatibilidade é satisfeita.

Tabela 5.2: Erro e ordem de convergência na norma L_∞ . Uma interface imersa de forma circular foi adicionada ao domínio (Figura 5.4b).

	u		v		p	
Malha	erro	ordem	erro	ordem	erro	ordem
32×32	1.066744e-03	-	1.066744e-03	-	6.815620e-02	-
64×64	3.442819e-04	1.6316	3.442819e-04	1.6316	3.139689e-02	1.1182
128×128	1.003691e-04	1.7783	1.003691e-04	1.7783	3.014452e-02	0.0587
256×256	2.478117e-05	2.0180	2.478117e-05	2.0180	1.377687e-02	1.1296

Tabela 5.3: Erro e ordem de convergência na norma L_∞ . O quadrado que delimita o domínio é formado por uma interface imersa (Figura 5.4c).

	u		v		p	
Malha	erro	ordem	erro	ordem	erro	ordem
25×25	1.144214e-03	-	1.154547e-03	-	5.849160e-04	-
50×50	2.830976e-04	2.0150	2.849836e-04	2.0184	1.462726e-04	1.9996
100×100	7.065844e-05	2.0024	7.090516e-05	2.0069	3.669836e-05	1.9949
150×150	3.141518e-05	1.9991	3.148765e-05	2.0020	1.631445e-05	1.9994

Figura 5.5: Evolução do passo de tempo adaptativo durante a simulação da solução manufaturada (5.8) (domínio sem interface) usando o método ARK3(2)4L[2]SA.

5.3.2 Escoamento ao redor de um círculo

Apresenta-se os resultados da simulação de um problema teste constituindo do escoamento ao redor de um círculo, semelhante a [20]. O círculo, com diâmetro D centrado no interior de um canal plano de altura H , é representado por uma interface imersa. O comprimento do canal é $L/D = 48$,

(a) Uma interface imersa de forma circular foi adicionada ao domínio.

(b) O quadrado que delimita o domínio é formado por uma interface imersa.

Figura 5.6: Campo de velocidades na simulação da solução manufaturada (5.8) em $t = 100s$. $\nu = 10^{-2}$.

$D/H = 1/8$ e o centro da interface imersa está a uma distância $l = L/4$, como na Figura 5.8. Na entrada, tem-se um perfil de velocidade parabólica com velocidade máxima $u_{\max}$ e condições de não-eskorregamento são aplicadas sobre a superfície do círculo e nas paredes superior e inferior do domínio computacional. A condição de saída é dada por um *outflow* continuativo como em (5.7).

A simulação é executada até $t = 20s$ considerando uma malha 192×32 num domínio $L = 6$ e $H = 1$. A viscosidade cinemática é $\nu = 0.25 \times 10^{-3}$.

(a) Divergente sem regularização.

(b) Divergente com regularização.

Figura 5.7: Valor do divergente na simulação da solução manufaturada (5.8), onde o quadrado que delimita o domínio é formado por uma interface imersa em $t = 100s$. $\nu = 10^{-2}$. Malha 100×100 .

A Figura 5.9 mostra o passo de tempo adaptativo. Quando não há restrição ao passo de tempo, este cresce até o fim da simulação e não há formação de vórtices, como pode ser observado nas figuras 5.9c e 5.10c. A restrição no passo de tempo permite a transição para escoamento turbulento, como pode ser observado nas figuras 5.9a e 5.10a e 5.9b e 5.10b. Na Figura 5.9a, o momento da transição para escoamento turbulento pode ser claramente percebido, pois há uma queda no valor do passo de

Figura 5.8: Escoamento ao redor de um círculo: geometria.

tempo.

Figura 5.9: Passo de tempo adaptativo na simulação do escoamento ao redor de um círculo. $\nu = 0.25 \times 10^{-3}$.

(a) Limitado por $\Delta t = 10^{-1}$. (b) Limitado por $\Delta t = 10^{-2}$. (c) Sem limite no Δt .

Figura 5.10: Perfil de velocidade em $t = 20s$ na simulação do escoamento ao redor de um círculo. $\nu = 0.25 \times 10^{-3}$.

Método de interface imersa explícito para o cálculo de derivadas

O método de interfaces imersas (MII) foi proposto em [25] para resolver equações elípticas com coeficientes descontínuos. A EDP é resolvida no domínio Ω^+ com condições de contorno em $\partial\Omega_0$ e interface imersa $\partial\Omega_i$, como na Figura 6.1. A solução em Ω^- pode ser de interesse ou não.

Figura 6.1: Ilustração do domínio contendo uma interface imersa $\partial\Omega_i$, extraído de [2].

Fundamental no MII, é o conceito de salto. O salto é definido como a diferença dos limites, à

esquerda e à direita, do valor da função de interesse, ou suas derivadas, na interface imersa:

$$[f^{(n)}]_{\alpha} = \lim_{x \rightarrow x_{\alpha}^{+}} f^{(n)}(x) - \lim_{x \rightarrow x_{\alpha}^{-}} f^{(n)}(x). \quad (6.1)$$

O termo $[f^{(n)}]_{\alpha}$ representa o salto no valor da n -ésima derivada da função em $x = x_{\alpha}$.

O MII original proposto por [25], de agora em diante denominado MII-analítico (MII-A), é de segunda ordem de precisão, mas tem como desvantagem a necessidade de se conhecer analiticamente os valores dos saltos na descontinuidade. Em problemas complexos, tal tarefa não é trivial.

Alternativamente, em [2] foi desenvolvido um método de interface imersa que calcula numericamente os valores dos saltos. Tal método será chamado aqui de método de interface imersa explícito (MII-E). Os autores conseguiram quarta ordem de precisão na simulação de escoamentos incompressíveis modelados pelas equações de Navier-Stokes na formulação função corrente-vorticidade. O presente trabalho tem como objetivo estender esse método de interface imersa para a formulação de variáveis primitivas em malha deslocada.

6.1 Série de Taylor corrigida

Considere uma função $f(x)$ com uma descontinuidade no ponto $x = x_{\alpha}$, como ilustrado na Figura 6.2. Esse ponto está entre os pontos de malha x_i e x_{i+1} . Ao se realizar cálculos de diferenças finitas no ponto x_i , o valor de $f(x_{i+1})$ não pode ser usado diretamente, devido à presença da descontinuidade. Deve-se extrapolar um valor para $f(x_{i+1})$.

Figura 6.2: Função $f(x)$ com uma descontinuidade no ponto $x = x_{\alpha}$.

Uma expansão em série de Taylor em torno do ponto x_i para aproximar $f(x)$ no ponto x_{i+1} , seria:

$$f^*(x_{i+1}) = f(x_i) + f^{(1)}(x_i)h + f^{(2)}(x_i)\frac{h^2}{2!} + \dots, \quad (6.2)$$

onde $h = x_{i+1} - x_i$. Essa expansão, no entanto, não corresponde ao valor verdadeiro de $f(x_{i+1})$, pois há uma descontinuidade entre x_i e x_{i+1} . Quer-se, então, corrigir essa expansão de forma que ela

represente corretamente o valor de $f(x_{i+1})$:

$$f(x_{i+1}) = f(x_i) + f^{(1)}(x_i)h + f^{(2)}(x_i)\frac{h^2}{2!} + \cdots + J_\alpha, \quad (6.3)$$

onde J_α é o termo corretor.

Tomando o limite quando $x \rightarrow x_\alpha^-$, uma outra expansão para calcular $f(x)$ no ponto x_{i+1} é dada por:

$$f^-(x_{i+1}) = \lim_{x \rightarrow x_\alpha^-} \left\{ f(x) + f^{(1)}(x)(h^+) + f^{(2)}(x)\frac{(h^+)^2}{2!} + \cdots \right\}, \quad (6.4)$$

onde $h^+ = x_{i+1} - x_\alpha$. Visto que ambas as expansões estão à esquerda da descontinuidade, tem-se que (6.2) é igual à (6.4):

$$f^*(x_{i+1}) = f^-(x_{i+1}). \quad (6.5)$$

Pode-se, também, tomar o limite quando $x \rightarrow x_\alpha^+$, gerando a expansão:

$$f^+(x_{i+1}) = \lim_{x \rightarrow x_\alpha^+} \left\{ f(x) + f^{(1)}(x)(h^+) + f^{(2)}(x)\frac{(h^+)^2}{2!} + \cdots \right\}. \quad (6.6)$$

Ao contrário das expansões anteriores, essa está à direita da descontinuidade, do mesmo lado que $f(x_{i+1})$. Portanto,

$$f(x_{i+1}) = f^+(x_{i+1}). \quad (6.7)$$

Combinando (6.5) com (6.7), o valor correto para $f(x_{i+1})$ pode ser escrito como,

$$f(x_{i+1}) = f^*(x_{i+1}) + \{f^+(x_{i+1}) - f^-(x_{i+1})\}. \quad (6.8)$$

Definido o termo corretor como

$$J_\alpha = \{f^+(x_{i+1}) - f^-(x_{i+1})\}, \quad (6.9)$$

e usando a definição de salto dada em (6.1),

$$J_\alpha = [f^{(0)}]_\alpha + [f^{(1)}]_\alpha(h^+) + \frac{1}{2!}[f^{(2)}]_\alpha(h^+)^2 + \cdots. \quad (6.10)$$

Logo, a expansão em série de Taylor em torno de x_i pode ser usada para calcular $f(x_{i+1})$, se o termo de correção J_α for adicionado.

Observe que J_α é a diferença entre o valor verdadeiro de $f(x_{i+1})$ e o valor extrapolado a partir de pontos à esquerda da descontinuidade $f^*(x_{i+1})$. Esse fato será usado na próxima seção para produzir correções aos estênceis próximos da descontinuidade.

6.2 Correção para estênceis de diferenças finitas

Em geral, um esquema de diferenças finitas compactas para aproximação da primeira derivada com quarta ordem de precisão tem a forma:

$$L_{i-1}^1 F_{i-1}^{(1)} + L_i^1 F_i^{(1)} + L_{i+1}^1 F_{i+1}^{(1)} = R_{i-1}^1 f_{i-1} + R_i^1 f_i + R_{i+1}^1 f_{i+1}. \quad (6.11)$$

onde $F_i^{(1)} = f^{(1)}(x_i) + \mathcal{O}(h^4)$, $f_i = f(x_i)$, L_i^1 e R_i^1 são, respectivamente, os coeficientes dos lados esquerdo e direito da aproximação para a primeira derivada.

Se houver uma descontinuidade entre x_i e x_{i+1} , como ilustrado na Figura 6.2, a expressão (6.11) deixa de ser válida. Isso ocorre porque $F_{i+1}^{(1)}$ e f_{i+1} estão à direita da descontinuidade, enquanto os outros termos estão à esquerda.

Uma solução é substituir os valores à direita da descontinuidade por extrapolações a partir de x_i :

$$L_{i-1}^1 F_{i-1}^{(1)} + L_i^1 F_i^{(1)} + L_{i+1}^1 F_{i+1}^{*(1)} = R_{i-1}^1 f_{i-1} + R_i^1 f_i + R_{i+1}^1 f_{i+1}^*, \quad (6.12)$$

onde $F_{i+1}^{*(1)} = f^{*(1)}(x_{i+1}) + \mathcal{O}(h^4)$, $f_{i+1}^* = f^*(x_{i+1})$ e $f^*(x_{i+1})$ é dado por (6.2).

A equação (6.8) pode ser reescrita como:

$$f_{i+1}^* = f_{i+1} - J_{\alpha 0}, \quad (6.13)$$

onde $J_{\alpha 0}$ é igual a J_α definido em (6.9).

Analogamente para a primeira derivada,

$$f_{i+1}^{*(1)} = f_{i+1}^{(1)} - J_{\alpha 1}, \quad (6.14)$$

onde $J_{\alpha 1}$ é a derivada de J_α . Também é válido afirmar que

$$F_{i+1}^{*(1)} = F_{i+1}^{(1)} - J_{\alpha 1} + \mathcal{O}(h^4). \quad (6.15)$$

Visto que $F_{i+1}^{*(1)}$ e $F_{i+1}^{(1)}$ ambos representam aproximações de quarta ordem, podemos fazer escolhas de forma que

$$F_{i+1}^{*(1)} = F_{i+1}^{(1)} - J_{\alpha 1}, \quad (6.16)$$

sem prejuízo da ordem do método.

Substituindo (6.13) e (6.16) em (6.12), tem-se a fórmula do estêncil corrigido:

$$L_{i-1}^1 F_{i-1}^{(1)} + L_i^1 F_i^{(1)} + L_{i+1}^1 F_{i+1}^{(1)} = R_{i-1}^1 f_{i-1} + R_i^1 f_i + R_{i+1}^1 f_{i+1} + (L_{i+1}^1 J_{\alpha 1} - R_{i+1}^1 J_{\alpha 0}). \quad (6.17)$$

Para tratar também o caso onde a descontinuidade está entre x_i e x_{i-1} , pode-se reescrever a equa-

ção (6.17) na forma mais geral,

$$L_{i-1}^1 F_{i-1}^{(1)} + L_i^1 F_i^{(1)} + L_{i+1}^1 F_{i+1}^{(1)} = R_{i-1}^1 f_{i-1} + R_i^1 f_i + R_{i+1}^1 f_{i+1} + (L_I^1 J_{\alpha 1} - R_I^1 J_{\alpha 0}). \quad (6.18)$$

onde $I = i + 1$ se $x_i < x_\alpha < x_{i+1}$ ou $I = i - 1$ se $x_{i-1} < x_\alpha < x_i$.

O mesmo procedimento pode ser adotado para corrigir o estêncil da segunda derivada:

$$L_{i-1}^2 F_{i-1}^{(2)} + L_i^2 F_i^{(2)} + L_{i+1}^1 F_{i+1}^{(2)} = R_{i-1}^2 f_{i-1} + R_i^2 f_i + R_{i+1}^2 f_{i+1} + (L_I^2 J_{\alpha 2} - R_I^2 J_{\alpha 0}). \quad (6.19)$$

onde $J_{\alpha 2}$ é a segunda derivada de J_α .

Para manter a ordem do método $J_{\alpha 0}$, $J_{\alpha 1}$ e $J_{\alpha 2}$ têm que ser calculados com pelo menos quarta ordem de precisão.

Se o salto ocorrer em $x_i < x_\alpha < x_{i+1}$, então $I = i + 1$, $h^+ = x_{i+1} - x_\alpha$ e

$$J_{\alpha 0} \approx [f^{(0)}]_\alpha + (h^+)[f^{(1)}]_\alpha + \frac{(h^+)^2}{2!}[f^{(2)}]_\alpha + \frac{(h^+)^3}{3!}[f^{(3)}]_\alpha + \frac{(h^+)^4}{4!}[f^{(4)}]_\alpha + \frac{(h^+)^5}{5!}[f^{(5)}]_\alpha, \quad (6.20a)$$

$$J_{\alpha 1} \approx [f^{(1)}]_\alpha + (h^+)[f^{(2)}]_\alpha + \frac{(h^+)^2}{2!}[f^{(3)}]_\alpha + \frac{(h^+)^3}{3!}[f^{(4)}]_\alpha + \frac{(h^+)^4}{4!}[f^{(5)}]_\alpha, \quad (6.20b)$$

$$J_{\alpha 2} \approx [f^{(2)}]_\alpha + (h^+)[f^{(3)}]_\alpha + \frac{(h^+)^2}{2!}[f^{(4)}]_\alpha + \frac{(h^+)^3}{3!}[f^{(5)}]_\alpha. \quad (6.20c)$$

Já se o salto ocorrer em $x_{i-1} < x_\alpha < x_i$, então $I = i - 1$, $h^- = x_\alpha - x_{i-1}$ e

$$J_{\alpha 0} \approx -[f^{(0)}]_\alpha + (h^-)[f^{(1)}]_\alpha - \frac{(h^-)^2}{2!}[f^{(2)}]_\alpha + \frac{(h^-)^3}{3!}[f^{(3)}]_\alpha - \frac{(h^-)^4}{4!}[f^{(4)}]_\alpha + \frac{(h^-)^5}{5!}[f^{(5)}]_\alpha, \quad (6.21a)$$

$$J_{\alpha 1} \approx -[f^{(1)}]_\alpha + (h^-)[f^{(2)}]_\alpha - \frac{(h^-)^2}{2!}[f^{(3)}]_\alpha + \frac{(h^-)^3}{3!}[f^{(4)}]_\alpha - \frac{(h^-)^4}{4!}[f^{(5)}]_\alpha, \quad (6.21b)$$

$$J_{\alpha 2} \approx -[f^{(2)}]_\alpha + (h^-)[f^{(3)}]_\alpha - \frac{(h^-)^2}{2!}[f^{(4)}]_\alpha + \frac{(h^-)^3}{3!}[f^{(5)}]_\alpha. \quad (6.21c)$$

Os saltos, como definidos em (6.1), podem ser obtidos por

$$[f^{(n)}]_\alpha = f_+^{(n)} - f_-^{(n)}. \quad (6.22)$$

No método original MII-A, o valor da função e suas derivada eram conhecidos analiticamente. O MII-E inovou ao calculá-los numericamente:

$$f_+^{(n)} \approx c_{n_\alpha+} f_\alpha^+ + c_{n_{i+2}} f_{i+2} + c_{n_{i+3}} f_{i+3} + c_{n_{i+4}} f_{i+4} + c_{n_{i+5}} f_{i+5} + c_{n_{i+6}} f_{i+6}, \quad (6.23a)$$

$$f_-^{(n)} \approx c_{n_\alpha-} f_\alpha^- + c_{n_{i-1}} f_{i-1} + c_{n_{i-2}} f_{i-2} + c_{n_{i-3}} f_{i-3} + c_{n_{i-4}} f_{i-4} + c_{n_{i-5}} f_{i-5}. \quad (6.23b)$$

Os pontos x_i e x_{i+1} foram evitados para contornar problemas de instabilidade. De forma mais geral:

$$f_{\pm}^{(n)} \approx c_{n_{\alpha\pm}} f_{\alpha}^{\pm} + c_{n_{\zeta_1}} f_{\zeta_1} + c_{n_{\zeta_2}} f_{\zeta_2} + c_{n_{\zeta_3}} f_{\zeta_3} + c_{n_{\zeta_4}} f_{\zeta_4} + c_{n_{\zeta_5}} f_{\zeta_5}, \quad (6.24)$$

onde os coeficientes são obtidos resolvendo o sistema linear

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & h_{\zeta_1} & h_{\zeta_2} & h_{\zeta_3} & h_{\zeta_4} & h_{\zeta_5} \\ 0 & h_{\zeta_1}^2 & h_{\zeta_2}^2 & h_{\zeta_3}^2 & h_{\zeta_4}^2 & h_{\zeta_5}^2 \\ 0 & h_{\zeta_1}^3 & h_{\zeta_2}^3 & h_{\zeta_3}^3 & h_{\zeta_4}^3 & h_{\zeta_5}^3 \\ 0 & h_{\zeta_1}^4 & h_{\zeta_2}^4 & h_{\zeta_3}^4 & h_{\zeta_4}^4 & h_{\zeta_5}^4 \\ 0 & h_{\zeta_1}^5 & h_{\zeta_2}^5 & h_{\zeta_3}^5 & h_{\zeta_4}^5 & h_{\zeta_5}^5 \end{bmatrix} \begin{bmatrix} c_{n_{\alpha\pm}} \\ c_{n_{\zeta_1}} \\ c_{n_{\zeta_2}} \\ c_{n_{\zeta_3}} \\ c_{n_{\zeta_4}} \\ c_{n_{\zeta_5}} \end{bmatrix} = \begin{bmatrix} 1\delta_{n0} \\ 1\delta_{n1} \\ 2!\delta_{n2} \\ 3!\delta_{n3} \\ 4!\delta_{n4} \\ 5!\delta_{n5} \end{bmatrix}, \quad (6.25)$$

em que $h_{\zeta_k} = x_{\zeta_k} - x_{\alpha}$ e δ é a função delta de Kronecker.

6.3 MII em malha deslocada

O método descrito na Seção 6.2 é válido para malhas colocadas, ou seja, para os casos onde as derivadas são calculadas nos mesmos pontos em que a função é amostrada. Esse é o caso, por exemplo, das segundas derivadas presentes no termo viscoso.

No caso deslocado, quando as derivadas são calculadas em pontos diferentes dos pontos onde a função é amostrada, é necessário adaptar o método descrito anteriormente. Supondo que a derivada seja calculada nos pontos de índices inteiros e a função seja amostrada nos pontos de índices semi-inteiros, o valor de h^+ (distância até a interface) não poderá ser o mesmo para se calcular $J_{\alpha 0}$ e $J_{\alpha 1}$ (ou $J_{\alpha 2}$ no caso da segunda derivada). A Figura 6.3 ilustra dois casos possíveis.

Figura 6.3: Casos possíveis onde a interface intersecta a malha. Em (a) tanto o valor da função (em $x_{i+1/2}$) quanto a derivada (em x_{i+1}) estão depois da descontinuidade. Em (b) apenas a derivada (em x_{i+1}) está depois da descontinuidade.

Para calcular a derivada em x_i , quando a interface está entre os pontos x_i e $x_{i+1/2}$ (Figura 6.3a),

é necessário calcular duas correções: uma para o valor da função em $x_{i+1/2}$ e uma para o valor da derivada em x_{i+1} . No primeiro caso, a correção será $J_{\alpha 0}$ considerando a distância h_a^+ . No segundo caso, a correção será $J_{\alpha 1}$ (ou $J_{\alpha 2}$ no caso da segunda derivada) considerando a distância h_b^+ .

Para calcular a derivada em x_i , quando a interface está entre os pontos $x_{i+1/2}$ e x_{i+1} (Figura 6.3b), é necessário calcular apenas uma correção para o valor da derivada em x_{i+1} . A correção será $J_{\alpha 1}$ (ou $J_{\alpha 2}$ no caso da segunda derivada) considerando a distância h_b^+ .

6.4 Resultados

Esquemas compactos para os cálculos da primeira e segunda derivadas foram descritos no Capítulo 2. Quando não é usada a técnica de interface imersa (Figura 6.4a), é necessário usar estênceis não centrais próximos à borda do domínio. No Apêndice A, um estudo para determinação desses estênceis é apresentado.

Figura 6.4: Em (a) a borda do domínio coincide com a malha computacional. Nesse caso, usam-se estênceis não centrais próximo à borda. Em (b) a borda do domínio intersecta as células computacionais. Nesse caso, usa-se o método de interface imersa.

A seguir, uma comparação entre o MII e estênceis não centrais é feita usando a solução manufaturada:

$$U(x, y) = \sin(2\pi x) \sin(2\pi y), \quad (6.26a)$$

$$V(x, y) = \cos(2\pi x) \cos(2\pi y), \quad (6.26b)$$

$$P(x, y) = \sin(2\pi x) \cos(2\pi y), \quad (6.26c)$$

no domínio $0 \leq x, y \leq 1$.

As figuras 6.5-6.8 mostram o erro na norma L_∞ para diferentes malhas. Nota-se que o uso do método da interface imersa não interferiu na ordem de convergência e quarta ordem foi obtida. Para as segundas derivadas, observa-se que, para malhas muito finas, há um aumento do erro. Isso acontece,

porque o erro de máquina (arredondamento) começa a predominar sobre o erro do método (truncamento). O cálculo da segunda derivada é mais sensível ao erro de arredondamento, pois há divisão por h^2 , o que amplifica o erro. Na primeira derivada, divide-se por h , e na interpolação, não há divisão por h .

Representar o contorno rígido com uma interface imersa torna o método, e consequentemente o código, mais simples. Caso fossem usados os estênceis não centrais, correções deveriam ser incorporadas a esses estênceis. No Apêndice A foi mostrado que a escolha dos estênceis não centrais não é trivial. Sendo assim, o uso do MII é vantajoso e menos erros numéricos são introduzidos.

Figura 6.5: Comparação da ordem de convergência do MII com estênceis não centrais.

Figura 6.6: Comparação da ordem de convergência do MII com estênceis não centrais.

Figura 6.7: Comparação da ordem de convergência do MII com estênceis não centrais.

Figura 6.8: Comparação da ordem de convergência do MII com estênceis não centrais.

Método de interface imersa implícito para a Equação de Poisson

O estudo de técnicas robustas e eficientes para a solução da equação de Poisson tem sido sempre de grande interesse. Em particular, quando as equações de Navier-Stokes são resolvidas pelo método de projeção, pois uma equação de Poisson precisa ser resolvida a cada passo de tempo, o que consome a maior parte do tempo de processamento. A presença de interfaces no domínio de solução da equação de Poisson aumenta o nível de dificuldade do processo numérico. Do ponto de vista computacional, existem vários métodos numéricos projetados para funções suaves que se comportam mal em problemas com interfaces ou que não funcionam. Uma maneira de contornar esse problema é usar o método de interface imersa, que permite a utilização de aproximações de alta ordem, mesmo na presença de descontinuidades.

7.1 Equação de Poisson 1D

Como visto no Capítulo 6, na presença de descontinuidades, uma aproximação de diferenças finitas compacta para a segunda derivada de uma função $f(x)$ pode ser calculada pela fórmula

$$L_{i-1}^2 f_{i-1}^{(2)} + L_i^2 f_i^{(2)} + L_{i+1}^2 f_{i+1}^{(2)} = R_{i-1}^2 f_{i-1} + R_i^2 f_i + R_{i+1}^2 f_{i+1} + (L_I^2 J_{\alpha 2} - R_I^2 J_{\alpha 0}) + \mathcal{O}(h^4), \quad (7.1)$$

onde $J_{\alpha 2}$, $J_{\alpha 0}$ são apresentadas em (6.20) e (6.21), e $I = i + 1$ se $x_i < x_\alpha < x_{i+1}$ ou $I = i - 1$ se $x_{i-1} < x_\alpha < x_i$.

No caso da equação de Poisson, ao contrário do que foi estudado no Capítulo 6, o valor da segunda derivada $f_i^{(2)}$ é conhecido. A incógnita agora é o valor da função f_i . Esse fato representa um desafio para o cálculo das correções $J_{\alpha 2}$ e $J_{\alpha 0}$, visto que elas dependem dos valores de f dentro do domínio

e na interface imersa.

Os autores do MII-E [2] adotam o procedimento iterativo descrito no Algoritmo 1 para calcular as correções. Um valor tentativo para f , inicialmente zero, é usado para calcular as correções. A equação de Poisson é resolvida produzindo um novo valor f_{new} . O processo é repetido, usando f_{new} como o novo valor tentativo para f , até que f_{new} seja suficientemente próximo a f . Um fator de relaxação β pode ser utilizado para melhorar a convergência.

Algoritmo 1 Processo do MII-E

```

 $f = 0, f_{\text{new}} = 0$ 
for  $i = 0 \rightarrow \text{iter}_{\text{max}}$  do
    poisson_solve( $f, f^{(2)}, f_{\text{new}}, \phi$ )
    if  $|f - f_{\text{new}}| < \varepsilon$  then
        break;
    else
         $f = \beta f_{\text{new}} + (1 - \beta)f$ 
    end if
end for

```

Esse método possui algumas desvantagens. O valor de β precisa ser calibrado manualmente para cada problema a ser resolvido. Um valor ruim de β pode significar convergência lenta ou até mesmo nenhuma convergência. Mesmo para valores ótimos de β , o tempo de processamento é grande, como será visto mais adiante.

Neste trabalho, foi desenvolvido um procedimento alternativo, chamado de MII-Implícito (MII-I), que calcula as correções $J_{\alpha 0}$ e $J_{\alpha 2}$ em (7.1) implicitamente. Todos os coeficientes que aparecem em (6.20), (6.21) e (6.23) são lançados na matriz que aproxima a equação de Poisson. Isso permite que a solução seja obtida em apenas um passo. A desvantagem é que a matriz perde sua característica tridiagonal, o que implica em tempo maior de processamento. Mesmo assim, como será visto adiante, o tempo de se resolver o sistema resultante é inferior àquele necessário para se resolver por meio do algoritmo iterativo explícito.

7.2 Equação de Poisson 2D

A discretização da equação de Poisson 2D com diferenças finitas compactas já foi apresentado na Seção 3.2.1. Na presença de interface imersa, deve-se acrescentar termos de correção a (3.21):

$$\mathcal{L}_{ij} f_{ij} = \mathcal{R}_{ij} f_{ij}^{(2)} + (L_j^{yy} J_{\alpha x I(j)}) + (L_i^{xx} J_{\alpha x J(i)}), \quad (7.2)$$

onde

$$\begin{aligned}\mathcal{L}_{ij} &= R_i^{xx} L_j^{yy} + L_i^{xx} R_j^{yy}, \\ \mathcal{R}_{ij} &= L_i^{xx} L_j^{yy}, \\ J_{\alpha x I} &= -(L_I^{xx} J_{\alpha 2x} - R_I^{xx} J_{\alpha 0x}), \\ J_{\alpha x J} &= -(L_J^{yy} J_{\alpha 2y} - R_J^{yy} J_{\alpha 0y}),\end{aligned}$$

$R_i^{xx}, L_i^{xx}, J_{\alpha 0x}, J_{\alpha 2x}$ correspondem a $R_i^2, L_i^2, J_{\alpha 0}, J_{\alpha 2}$ na direção x . O mesmo vale para a direção y . Ainda

$$\begin{aligned}I &= \begin{cases} i-1, & \text{se } x_{i-1} < x_\alpha < x_i, \\ i+1, & \text{se } x_i < x_\alpha < x_{i+1}, \end{cases} \\ J &= \begin{cases} j-1, & \text{se } x_{j-1} < x_\alpha < x_j, \\ j+1, & \text{se } x_j < x_\alpha < x_{j+1}. \end{cases}\end{aligned}$$

Note que $I = I(j)$ e $J = J(i)$, ou seja, as correções são feitas separadamente para cada direção, usando o procedimento 1D.

A Figura 7.1 mostra os pontos utilizados ao se calcular a equação de Poisson em um ponto próximo à interface imersa. Os pontos do estêncil de diferenças finitas compactas são representados por $\times$. Os pontos internos ao domínio usados para o cálculo das correções são representados por $\square$. Os pontos sobre a interface imersa, também usados no cálculo da correção, são representados por $\circ$. Os pontos $\circ$, sobre a interface, são conhecidos quando se trabalha com condições de contorno do tipo Dirichlet. Os pontos $\square$, internos ao domínio, são aproximados iterativamente no MII-E ou tratados implicitamente no MII-I.

7.2.1 Condições de contorno do tipo Neumann

Em situações com condições de contorno do tipo Neumann, o valor da função e suas derivadas não podem ser aproximados como em (6.24). Ao invés do valor da função na descontinuidade, agora, conhece-se o valor da derivada. Em (6.24),

$$f_\pm^{(n)} = c_{n_{\alpha\pm}} f_\alpha^\pm + \sum_i c_{n_{\zeta_i}} f_{\zeta_i}. \quad (7.4)$$

Em particular, para $n = 1$

$$f_\pm^{(1)} = c_{1_{\alpha\pm}} f_\alpha^\pm + \sum_i c_{1_{\zeta_i}} f_{\zeta_i}. \quad (7.5)$$

Isolando $f_\alpha^\pm$,

Figura 7.1: Estêncil interceptando uma interface imersa, extraído de [2]. Na figura, $\times$ representam os pontos do estêncil de diferenças finitas compactas, $\square$ os pontos internos ao domínio usados para o cálculo das correções e $\circ$ os pontos sobre a interface imersa, também usados no cálculo da correção.

$$f_{\alpha}^{\pm} = \frac{1}{c_{1_{\alpha\pm}}} \left(f_{\pm}^{(1)} - \sum_i c_{1_{\zeta_i}} f_{\zeta_i} \right). \quad (7.6)$$

Por fim, substituindo (7.6) em (7.4), chega-se à expressão da aproximação para o caso de condições de contorno do tipo Neumann:

$$f_{\pm}^{(n)} = \frac{c_{n_{\alpha\pm}}}{c_{1_{\alpha\pm}}} f_{\pm}^{(1)} + \sum_i \left(c_{n_{\zeta_i}} - \frac{c_{n_{\alpha\pm}} c_{1_{\zeta_i}}}{c_{1_{\alpha\pm}}} \right) f_{\zeta_i}. \quad (7.7)$$

7.3 Resultados

Nesta seção, apresentam-se os resultados do MII-I para a equação de Poisson. É feita uma comparação com o MII-E para os casos unidimensional e bidimensional.

7.3.1 Equação de Poisson 1D

Para o primeiro teste, considerou-se a equação de Poisson 1D definida em $0 \leq x \leq 1$

$$\frac{\partial^2 f}{\partial x^2} = g, \quad (7.8)$$

onde o termo fonte g é dado por

$$g(x) = \begin{cases} 0, & \text{se } 1/3 < x < 2/3, \\ -\cos(x), & \text{caso contrário.} \end{cases} \quad (7.9)$$

Para o MII-E utilizou-se $\beta = 0.3$. As tabelas 7.1 e 7.2 apresentam a ordem de convergência do MII-E e MII-I. Por se tratar de um método de quarta ordem, a precisão de máquina é logo atingida ao se refinar a malha.

Tabela 7.1: Ordem de convergência MII-E - Poisson 1D: $f = 0$ se $\phi < 0$, $\beta = 0.3$.

N	Norma L_1	ordem	Norma L_2	ordem	Norma L_∞	ordem
25	2.009995e-10	-	2.756129e-10	-	5.156291e-10	-
50	3.903755e-12	5.686185	5.290601e-12	5.703068	1.013256e-11	5.669263
75	6.614504e-13	4.378328	9.107896e-13	4.339153	1.794898e-12	4.268693
100	2.460683e-13	3.437218	3.300722e-13	3.528202	6.278311e-13	3.651364
125	8.683794e-14	4.667692	1.185365e-13	4.589380	2.322587e-13	4.456412
150	3.919896e-14	4.362586	5.374625e-14	4.338198	1.059153e-13	4.306743
175	2.112704e-14	4.009690	2.863491e-14	4.084623	5.562217e-14	4.178104
200	9.934563e-15	5.650610	1.371511e-14	5.512775	2.742251e-14	5.296266

Tabela 7.2: Ordem de convergência MII-I - Poisson 1D: $f = 0$ se $\phi < 0$.

N	Norma L_1	ordem	Norma L_2	ordem	Norma L_∞	ordem
25	3.319261e-10	-	3.564264e-10	-	5.444732e-10	-
50	5.286806e-12	5.972322	5.931969e-12	5.908949	1.058309e-11	5.685028
75	7.840519e-13	4.083468	9.398095e-13	4.544002	1.798117e-12	4.371566
100	2.811322e-13	3.565221	3.400973e-13	3.533224	6.363798e-13	3.610582
125	9.476294e-14	4.873306	1.230957e-13	4.554333	2.391420e-13	4.386136
150	4.379269e-14	4.233791	5.722731e-14	4.200988	1.119105e-13	4.164940
175	2.484346e-14	3.677393	3.211840e-14	3.746998	6.172840e-14	3.859569
200	1.352752e-14	4.552248	1.814548e-14	4.276206	3.541611e-14	4.160649

Ainda para mostrar a eficiência deste método, na segunda função teste definida em $0 \leq x \leq 1$, resolvemos o problema com

$$g = \begin{cases} \cos(10x), & \text{se } x \leq 1/3, \\ e^{2x} - 6, & \text{se } x \geq 2/3, \\ 100(x - 0.5)^2 - 2, & \text{caso contrário.} \end{cases} \quad (7.10)$$

Essa função será identificada por *domínio com 3 funções* e para esse teste, foi preciso um parâmetro de relaxação muito pequeno, $\beta = 0.001$. As tabelas 7.3 e 7.4 apresentam a ordem de convergência do MII-E e MII-I, respectivamente.

Tabela 7.3: Ordem de convergência MII-E - Poisson 1D: domínio com 3 funções, $\beta = 0.001$.

N	Norma L_1	ordem	Norma L_2	ordem	Norma L_∞	ordem
50	5.783032e-07		1.234673e-06		3.549823e-06	
100	3.844134e-08	3.911096	8.116145e-08	3.927191	2.301035e-07	3.947392
200	1.501405e-09	4.678274	2.975560e-09	4.769562	8.584064e-09	4.744478
400	9.409918e-11	3.995987	1.860696e-10	3.999247	5.360790e-10	4.001143
800	5.662565e-12	4.054655	1.116642e-11	4.058604	3.198930e-11	4.066784
1600	4.108027e-13	3.784938	7.405317e-13	3.914462	2.130962e-12	3.908013

Tabela 7.4: Ordem de convergência MII-I - Poisson 1D: domínio com 3 funções.

N	Norma L_1	ordem	Norma L_2	ordem	Norma L_∞	ordem
50	1.268543e-06		1.880323e-06		4.178673e-06	
100	5.666366e-08	4.484605	9.043135e-08	4.378014	2.359709e-07	4.146364
200	1.680979e-09	5.075050	3.048117e-09	4.890833	8.742758e-09	4.754377
400	9.859371e-11	4.091663	1.868826e-10	4.027714	5.373352e-10	4.024194
800	5.733875e-12	4.103907	1.122824e-11	4.056929	3.216571e-11	4.062227
1600	3.570891e-13	4.005155	7.012223e-13	4.001115	2.007727e-12	4.001888

Para o primeiro teste, onde $f = 0$ se $\phi < 0$, o método converge em 78 iterações independentemente do refinamento da malha. Neste caso onde a função é muito simples, nada se pode concluir em relação à ordem de convergência, pois esta varia muito. Isso não ocorre no segundo caso. Para esse teste, domínio com 3 funções, mesmo com a função do meio do domínio desconexa, isto é, não possuindo condições de contorno que a definem bem, o método converge com ordem 4 em torno de 28300 iterações.

7.3.2 Equação de Poisson 2D

Considere, agora, a equação de Poisson 2D

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = g, \quad (7.11)$$

definida em um quadrado unitário, onde o termo fonte g e a solução exata f são dados por

$$g(x, y) = \begin{cases} 0, & \text{se } (x - 0.5)^2 + (y - 0.5)^2 \leq 0.05123^2, \\ -4\pi^2 \sin(2\pi x) \cos(2\pi y), & \text{caso contrário,} \end{cases} \quad (7.12a)$$

$$f(x, y) = 0.5 \sin(2\pi x) \cos(2\pi y). \quad (7.12b)$$

Considerou-se nesse caso $\beta = 0.5$ para o parâmetro de relaxação. As tabelas 7.5 e 7.6 apresentam a análise de convergência para o MII-E e MII-I. Pode-se notar que, em ambos, o método converge com ordem 4. O número de iterações necessárias (MII-E) para a convergência foi 180, 61, 55, 57 e

79 para as malhas 25×25 , 50×50 , 100×100 , 200×200 e 400×400 , respectivamente.

Tabela 7.5: Ordem de convergência MII-E - Poisson 2D, $\beta = 0.5$.

N	Norma L_1	ordem	Norma L_2	ordem	Norma L_∞	ordem
25×25	2.207778e-06		3.539037e-06		1.498071e-05	
50×50	1.003699e-07	4.459197	1.426827e-07	4.632475	4.096901e-07	5.192430
100×100	5.921235e-09	4.083284	8.307886e-09	4.102185	2.460879e-08	4.057287
200×200	3.675163e-10	4.010018	5.127254e-10	4.018223	1.529837e-09	4.007724
400×400	2.297891e-11	3.999426	3.197327e-11	4.003248	9.518258e-11	4.006537

Tabela 7.6: Ordem de convergência MII-I - Poisson 2D.

N	Norma L_1	ordem	Norma L_2	ordem	Norma L_∞	ordem
25×25	2.207778e-06		3.539037e-06		1.498071e-05	
50×50	1.003699e-07	4.459197	1.426827e-07	4.632475	4.096901e-07	5.192430
100×100	5.921237e-09	4.083284	8.307888e-09	4.102185	2.460880e-08	4.057287
200×200	3.675127e-10	4.010032	5.127233e-10	4.018229	1.529891e-09	4.007673
400×400	2.304058e-11	3.995545	3.206159e-11	3.999263	9.541717e-11	4.003036

7.3.3 Tempo de processamento

As tabelas 7.7, 7.8 e 7.9 mostram o tempo de processamento (em segundos) para diferentes malhas. Esses resultados foram obtidos em uma máquina *dual Intel Xeon E5690*. Cada um dos processadores dessas máquinas possui 6 núcleos físicos e 12 núcleos lógicos (*hyper-threading*), além de 12MB de *cache* L3. A frequência base é 3.46 GHz e a máquina possui 32GB RAM. Podemos observar que o uso do MII-I reduziu consideravelmente o tempo de processamento ao considerar a malha mais fina. No caso 2D um ganho de aproximadamente 300% foi observado. Além disso, o tempo gasto para resolver a Equação de Poisson com MII-I é aproximadamente 20% maior que aquele necessário para resolver o mesmo problema sem nenhuma interface imersa. Esse fato é muito relevante para as equações de Navier-Stokes, como já mencionado, a resolução da equação de Poisson consome a maior parte do tempo de processamento. Desta forma, o MII-I mostrou-se tão eficiente quanto o MII-E e ainda conseguiu-se obter a ordem de convergência esperada, além de uma grande redução no tempo de processamento.

7.3.4 Condições de contorno do tipo Dirichlet e Neumann

Considere a solução manufaturada definida em (7.12). A Figura 7.2 mostra o erro na norma L_∞ considerando condições de contorno do tipo Dirichlet e Neumann. Observa-se que o maior valor do erro está distante dos pontos em que a função é conhecida. As tabelas 7.10 e 7.11 mostram a ordem de convergência. Com condições de contorno Dirichlet o método converge com quarta ordem, enquanto

Tabela 7.7: Tempo de processamento - Poisson 1D: $f = 0$ se $\phi < 0$.

Domínio com $f = 0$ se $\phi < 0$		
N	MII-E	MII-I
50	0.00517825s	0.00121225s
100	0.00531295s	0.00146802s
200	0.00578504s	0.00299064s
400	0.00651125s	0.00134199s
800	0.00896739s	0.01439490s

Tabela 7.8: Tempo de processamento - Poisson 1D: domínio com 3 funções.

Domínio com 3 funções		
N	MII-E	MII-I
50	1.81061s	0.00123524s
100	1.85293s	0.00111826s
200	1.96342s	0.00150373s
400	2.19228s	0.00202756s
800	2.65057s	0.00282848s

Tabela 7.9: Tempo de processamento - Poisson 2D

N	sem interface imersa	MII-E	MII-I
25x25	0.003145s	0.256026s	0.005499s
50x50	0.008611s	0.171169s	0.011719s
100x100	0.039583s	0.323859s	0.041598s
200x200	0.161420s	0.845820s	0.159381s
400x400	0.652636s	3.292730s	1.169880s
800x800	2.956840s	15.78380s	4.985200s

que com condições de contorno Neumann o método converge com ordem entre quarto e cinco, na norma L_∞ .

Tabela 7.10: Erro e ordem de convergência - Poisson com Dirichlet.

Malha	Norma L_1	ordem	Norma L_2	ordem	Norma L_∞	ordem
100x100	1.629208e-08	-	2.341993e-08	-	7.479806e-08	-
200x200	1.005615e-09	4.018021	1.464087e-09	3.999663	4.705821e-09	3.990482
300x300	1.983019e-10	4.004200	2.893727e-10	3.998549	9.306554e-10	3.997055
400x400	6.281614e-11	3.996004	9.173812e-11	3.993219	2.951269e-10	3.992197
500x500	2.566508e-11	4.011233	3.749813e-11	4.009288	1.206509e-10	4.008648

(a) Condição de contorno do tipo Dirichlet

(b) Condição de contorno do tipo Neumann

Figura 7.2: Ilustração do erro na aproximação da equação de Poisson. Malha 100×100 .

Tabela 7.11: Erro e ordem de convergência - Poisson com Neumann.

Malha	Norma L_1	ordem	Norma L_2	ordem	Norma L_∞	ordem
100x100	1.322338e-06	-	1.523960e-06	-	2.991127e-06	-
200x200	5.506707e-08	4.585757	6.098441e-08	4.643241	1.047352e-07	4.835871
300x300	9.763850e-09	4.266372	1.060736e-08	4.313739	1.628568e-08	4.590159
400x400	3.231330e-09	3.843802	3.468259e-09	3.885869	4.822848e-09	4.230093
500x500	9.926695e-10	5.289202	1.064355e-09	5.293828	1.467060e-09	5.333357

Método de interface imersa para as Equações de Stokes

O método de projeção exata para as equações de Navier-Stokes, desenvolvido no Capítulo 3, foi estendido para ser aplicado em domínios com interface imersa. No entanto, o uso do método de interface imersa para o termo convectivo mostrou-se instável, mesmo com a utilização de filtros (Apêndice B). A interface imersa produz *outflow* prescrito (Dirichlet) no domínio computacional, mesmo que não haja *outflow* no domínio físico. Isso prejudica a condição de compatibilidade da equação da continuidade e da equação de Poisson, além de causar instabilidades no termo convectivo. Portanto, a seguir são apresentados o desenvolvimento e os resultados para as equações de Stokes

$$\frac{\partial \mathbf{u}}{\partial t} = -\nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{g}, \quad (8.1a)$$

$$\nabla \cdot \mathbf{u} = 0. \quad (8.1b)$$

8.1 Equação do momento

Visto que o método de projeção exata, desenvolvido no Capítulo 3, é verdadeiramente livre de pressão, o cálculo da equação do momento resume-se ao termo viscoso. Trata-se de uma derivada segunda em malha colocada. Na presença de interface imersa, o método apresentado na Seção 6.2 foi aplicado diretamente.

8.2 Equação da continuidade

Os métodos de interface imersa para a equação de Poisson, apresentados no Capítulo 7, são aplicáveis a problemas com condições de contorno de Dirichlet, visto que é necessário conhecer o valor da

função, no caso a pressão, sobre a interface. No caso das equações de Navier-Stokes e Stokes, essas condições não estão definidas.

Portanto, ao invés de se aplicar correções aos estênceis, os valores das velocidades foram extrapolados com precisão de quarta ordem.

A interface pode intersectar a célula de duas formas diferentes, como visto na Figura 8.1. No primeiro caso (Figura 8.1a), o ponto U está além da interface imersa, fora do domínio. Seu valor é obtido por meio de interpolação a partir dos valores internos ao domínio. No segundo caso (Figura 8.1b), mesmo estando U dentro do domínio de interesse, ele deve ser considerado como condição de contorno e seu valor deve ser extrapolado. Isso garante que o sistema originado da discretização da equação de Poisson terá o número correto de incógnitas e equações.

Figura 8.1: Posições da interface.

8.3 Resultados

8.3.1 Exemplo I

Considere a solução manufaturada:

$$U(x, y) = \sin(2\pi x) \sin(2\pi y), \quad (8.2a)$$

$$V(x, y) = \cos(2\pi x) \cos(2\pi y), \quad (8.2b)$$

no domínio $0 \leq x, y \leq 1$. A viscosidade cinemática considerada foi $\nu = 10^{-2}$. A Tabela 8.1 mostra a ordem de convergência na norma L_∞ . Embora o método seja de quarta ordem, observa-se que a ordem varia durante o refinamento da malha.

8.3.2 Exemplo II

Considere a solução manufaturada polinomial:

$$U(x, y) = (x - r_{u1})(x - r_{u2})(x - r_{u3})(x - r_{u4})(x - r_{u5})(x - r_{u6}), \quad (8.3a)$$

$$V(x, y) = (x - r_{v1})(x - r_{v2})(x - r_{v3})(x - r_{v4})(x - r_{v5})(x - r_{v6}), \quad (8.3b)$$

Tabela 8.1: Erro e ordem de convergência na norma L_∞ - Equação de Stokes: exemplo I.

Malha	u		v	
	erro	ordem	erro	ordem
25×25	3.811216e-05	-	3.811216e-05	-
50×50	1.292178e-06	4.8824	1.292178e-06	4.8824
75×75	1.739474e-07	4.9458	1.739474e-07	4.9458
100×100	6.764021e-08	3.2833	6.764021e-08	3.2833

no domínio $0 \leq x, y \leq 1$. Considerou-se $r_{ui} = r_{vi}$, $r_{ui} = ir_{u1}$ com $r_{u1} = 0.1$, $i = 1, \dots, 6$. A viscosidade cinemática considerada foi $\nu = 10^{-2}$.

A Tabela 8.2 mostra a ordem de convergência na norma L_∞ . Percebe-se que a ordem do método está entre quarto e cinco.

Tabela 8.2: Erro e ordem de convergência na norma L_∞ - Equação de Stokes: exemplo II.

Malha	u		v	
	erro	ordem	erro	ordem
25×25	3.189118e-07	-	5.217187e-07	-
50×50	2.347125e-08	3.7642	2.122606e-08	4.6194
75×75	3.956202e-09	4.3912	3.289347e-09	4.5985
100×100	1.066220e-09	4.5577	1.013122e-09	4.0936

Outro teste foi feito adaptando a solução manufaturada do escoamento forçado em uma cavidade, equação 3.4.2. Na malha grossa 25×25 , o erro da solução u e v é da ordem de $O(10^{-10})$. Isso ocorre, porque a solução é polinomial e as equações de Stokes são lineares, e devido à alta ordem do método, as derivadas de altas ordens, relacionados ao erro de truncamento, são iguais a zero, ou seja, a solução convergiu para a solução exata.

8.3.3 Exemplo III

Considere a solução manufaturada

$$U(x, y) = y(\sin(x^2 + y^2) + \epsilon), \quad (8.4a)$$

$$V(x, y) = -x(\sin(x^2 + y^2) + \epsilon), \quad (8.4b)$$

no domínio $0 \leq x, y \leq 1$ e $\epsilon = 10^{-3}$. A viscosidade cinemática considerada foi $\nu = 10^{-2}$.

A Tabela 8.3 mostra a ordem de convergência na norma L_∞ . Percebe-se que o erro de máquina é atingido, não sendo possível calcular a ordem do método.

Diante dos resultados apresentados, pode-se concluir que o método aproxima de forma satisfatória as equações de Stokes. Em relação a ordens maiores que a ordem formal do método ter sido

Tabela 8.3: Erro e ordem de convergência na norma L_∞ - Equação de Stokes: exemplo III.

Malha	u		v	
	erro	ordem	erro	ordem
25×25	7.016767e-07	-	7.016767e-007	-
50×50	2.292901e-08	4.9356	2.292901e-008	4.9356
100×100	1.195011e-07	-2.3818	1.195011e-007	-2.3818

observada, trabalhos na literatura mostram esse comportamento, como em [31].

Conclusões

O objetivo inicial do presente trabalho era desenvolver um método de interface imersa para a resolução numérica das equações de Navier-Stokes. O método deveria ser compatível com diferenças finitas compactas de quarta ordem, usando variáveis primitivas em malha deslocada. Apesar desse objetivo não ter sido atingido em sua totalidade, grande progresso foi feito.

No Capítulo 3, foi apresentado um método de projeção para as equações de Navier-Stokes com variáveis primitivas e em malha deslocada. Esse método usa diferenças finitas compactas de quarta ordem, inclusive para a condição de incompressibilidade. Na literatura, a restrição de incompressibilidade, usando diferenças finitas compactas, ou era feita de forma aproximada ou baseada em estênceis muito grandes. Esses estênceis não eram compactos e poderiam se estender ao longo do comprimento do domínio. O presente método possui uma implementação simples, pois o sistema resultante envolve equações diferenciais unidimensionais de primeira ordem. Ao contrário da maioria dos métodos de projeção existentes, trata-se de um método verdadeiramente livre de pressão, ou seja, a cada passo de tempo, não há nenhuma dependência da pressão do tempo anterior, nem na equação do momento e nem nas condições de contorno. Essa independência não prejudica a ordem temporal do método, nem os resultados para estados estacionários.

Nos capítulos 4 e 5, foi apresentado um estudo sobre métodos acoplados para as equações de Navier-Stokes e métodos de fronteira imersa. O método de fronteira imersa utilizado considera pontos fantasmas e o valor em tais pontos são obtidos por meio da interpolação dos valores internos ao domínio e do valor na interface. Esse estudo contribuiu para que o grupo avançasse na pesquisa e serviu de base para o código de quarta ordem desenvolvido posteriormente.

O método de interface imersa que serviu como base foi o de Linnick e Fasel [2]. Esse era interessante pelo fato de se calcular numericamente os valores dos saltos na função ou em suas derivadas. As extensões desenvolvidas foram descritas nos capítulos 6 e 7 para o cálculo de derivadas em malha

deslocada e para a resolução da Equação de Poisson. Os resultados mostraram convergência com quarta ordem de precisão.

Por fim, no Capítulo 8, era esperado resolver as equações de Navier-Stokes usando o método de interface imersa compatível com diferenças finitas compactas de quarta ordem. Isso não foi possível, pois a discretização do termo convectivo com diferenças finitas compactas gerou instabilidades numéricas. Tentou-se eliminá-las, sem sucesso, com o uso de filtros. Outras técnicas para contornar esse problema são deixadas para trabalhos futuros. Como contribuição, conseguiu-se resolver as equações de Stokes com quarta ordem de precisão.

9.1 Artigos publicados em periódicos

- G. A. Reis, I. V. M. Tasso, L. F. Souza and J. A. Cuminato, A compact finite differences exact projection method for the Navier-Stokes equations on a staggered grid with fourth-order spatial precision, *Computers & Fluids*, (2015), DOI: 10.1016/j.compfluid.2015.06.015.

9.2 Artigos publicados em conferências

- REIS, G. A. ; TASSO, I. ; SOUZA, L. ; CUMINATO, J. A. . Um novo tipo de método de projeção com diferenças finitas compactas de alta ordem. In: III CMAC - Sudeste 2015 - Congresso de Matemática Aplicada e Computacional, 2015, Vitória-ES.
- REIS, G. A. ; TASSO, I. ; SOUZA, L. ; CUMINATO, J. A. . Resolução numérica das equações de Navier-Stokes usando o método das interfaces imersas com correção do salto implícita em malha deslocada. In: XXXV CNMAC - Congresso Nacional de Matemática Aplicada e Computacional, 2014, Natal-RN.
- COLNAGO, M. ; REIS, G. A. ; TASSO, I. ; SOUZA, L. . Método de Interface Imersa de Alta Ordem para o Cálculo de Derivadas de Funções Descontínuas. In: XXXV CNMAC - Congresso Nacional de Matemática Aplicada e Computacional, 2014, Natal-RN.
- REIS, G. A. ; TASSO, I. ; SOUZA, L. ; CUMINATO, J. A. . Fast and precise solution for the Poisson equation in the presence of interfaces with discontinuities. In: 22nd International Congress of Mechanical Engineering, 2013, Ribeirão Preto-SP.

9.3 Trabalhos futuros

- Desenvolver um método *upwind* compacto de alta ordem ou algo equivalente, para domínios com interface imersa, a fim de estabilizar o termo convectivo.
- Usar o método de interface imersa implícito desenvolvido no Capítulo 7 na etapa de projeção.
- Extensão para escoamentos com fronteiras móveis (*Moving boundary*) e superfícies livres.

- Extensão do método para 3D.

Métodos de diferenças finitas compactas

Os métodos de diferenças finitas compactas são particularmente atrativos devido ao fato de que o ganho na ordem de aproximação não vem exclusivamente da inclusão de pontos adicionais no estêncil computacional. A quantidade a ser calculada, ou seja, a derivada, está incluída na equação que a aproxima. Vários trabalhos foram publicados mostrando as vantagens desta classe de métodos [4, 7, 94].

Por exemplo, para aproximar a primeira ou segunda derivada, é possível obter até quarta ordem de precisão usando não mais do que três pontos, ou seja, usa-se a fórmula

$$L_{i-1}F_{i-1}^{(k)} + L_iF_i^{(k)} + L_{i+1}F_{i+1}^{(k)} = R_{i-1}f_{i-1} + R_if_i + R_{i+1}f_{i+1} \quad (\text{A.1})$$

onde f_i é a função no ponto x_i dada e $F^{(k)}$, $k = \{1, 2\}$, é a aproximação para a k -éssima derivada no ponto x_i , como ilustrado na Figura A.1a. L_i e R_i são os coeficientes da esquerda e da direita, que podem ser obtidos combinando séries de Taylor de várias ordens [4]. Para esses esquemas centrais, vários estudos de precisão e estabilidade existem [4, 95, 96].

Para condições de contorno não periódicas, a equação (A.1) não pode ser aplicada nos pontos da borda do domínio sem a utilização de células fantasmas (extrapolações), uma vez que a utilização de células fantasmas será evitada. Portanto, são necessários esquemas não centrais, como ilustrado na Figura A.2. Muitas vezes, esquemas de ordem inferior são usados para melhorar a estabilidade [4, 23, 97, 98]. Em [95], os autores estudaram o comportamento de esquemas de quarta ordem no contorno quando combinado com esquemas centrais de ordem igual.

Em malhas deslocadas, aparece uma dificuldade, visto que as incógnitas, ou seja, as derivadas, não são amostradas nas mesmas posições que a função, como ilustrado nas figuras A.1b e A.2c. Isso requer o desenvolvimento de estênceis deslocados específicos. Além disso, alguns problemas, como

Figura A.1: Estênceis centrais em malhas colocadas (a) e deslocadas (b). $f_i = f(x_i)$, $F_i^{(k)} \approx \left. \frac{\partial^k f}{\partial x^k} \right|_{x_i}$, $k = \{1, 2\}$ e h é o espaçamento da malha.

Figura A.2: Estênceis para malhas uniforme (a), não uniforme (b) e deslocada (c) próximo ao contorno. $f_i = f(x_i)$, $F_i^{(k)} \approx \left. \frac{\partial^k f}{\partial x^k} \right|_{x_i}$, $k = \{1, 2\}$.

nas equações de Navier-Stokes incompressíveis, requerem esquemas de interpolação, que também podem se beneficiar de técnicas compactas.

Neste trabalho, tanto no interior (central) quanto no contorno (não centrais), os esquemas considerados são de quarta ordem de precisão. As seções que seguem mostram que tal combinação é possível além de ser estável e precisa.

A.1 Desenvolvimento dos esquemas de diferenças finitas compactas

Nos métodos de diferenças finitas compactas as aproximações são obtidas pela combinação das expansões em série de Taylor da função e suas derivadas em torno do ponto x_i . Assim, a equação diferencial dá origem a um sistema de equações algébricas onde as incógnitas são os valores aproxi-

mados da derivada nos pontos escolhidos de uma malha.

A.1.1 Aproximações compactas para a primeira derivada

Para a aproximação da primeira derivada, tomando $k = 1$ em (A.1), obtem-se:

$$L_{i-1}F_{i-1}^{(1)} + L_iF_i^{(1)} + L_{i+1}F_{i+1}^{(1)} = R_{i-1}f_{i-1} + R_if_i + R_{i+1}f_{i+1}. \quad (\text{A.2})$$

Expandindo f_{i-1} , f_{i+1} , $F_{i-1}^{(1)}$ e $F_{i+1}^{(1)}$ em série de Taylor em torno do ponto x_i e, impondo que o coeficiente de $F_i^{(1)}$ seja igual a 1, obtemos o seguinte sistema

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & -h & 0 & h \\ -h & 0 & h & h^2/2! & 0 & h^2/2! \\ h^2/2! & 0 & h^2/2! & -h^3/3! & 0 & h^3/3! \\ -h^3/3! & 0 & h^3/3! & h^4/4! & 0 & h^4/4! \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} L_{i-1} \\ L_i \\ L_{i+1} \\ R_{i-1} \\ R_i \\ R_{i+1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad (\text{A.3})$$

onde h é o espaçamento da malha.

Resolvendo esse sistema, obtem-se

$$L_{i-1} = 1/4, \quad L_i = 1, \quad L_{i+1} = 1/4, \quad R_{i-1} = 3/4h, \quad R_i = 0, \quad R_{i+1} = -3/4h.$$

Assim, (A.2) é reescrita como

$$F_{i-1}^{(1)} + 4F_i^{(1)} + F_{i+1}^{(1)} = \frac{3(f_{i+1} - f_{i-1}))}{h} + O(h^4). \quad (\text{A.4})$$

Para os pontos do contorno, isto é, $i = 0$ (e seu correspondente $i = N$), o valor da derivada pode ser obtido a partir da relação

$$F_0^{(1)} + \alpha F_1^{(1)} = \frac{1}{h}(af_0 + bf_1 + cf_2 + df_3). \quad (\text{A.5})$$

Expandindo f_i , $f_i^{(1)}$ em série de Taylor em torno do ponto x_0 , a Tabela A.1 apresenta os coeficientes a , b , c , d e α para esquemas de terceira e quarta ordens.

Tabela A.1: Coeficientes do esquema numérico para aproximar a primeira derivada no contorno.

	a	b	c	d	α
3ª ordem	-15/6	2	1/2	0	2
4ª ordem	-17/6	3/2	3/2	-1/6	3

A.1.2 Aproximações compactas para a segunda derivada

De maneira análoga tomando $k = 2$ em (A.1), obtem-se

$$L_{i-1}F_{i-1}^{(2)} + L_iF_i^{(2)} + L_{i+1}F_{i+1}^{(2)} = R_{i-1}f_{i-1} + R_if_i + R_{i+1}f_{i+1}. \quad (\text{A.6})$$

Expandindo $f_{i-1}, f_{i+1}, F_{i-1}^{(2)}$ e $F_{i+1}^{(2)}$ em série de Taylor em torno do ponto x_i e, impondo que o coeficiente de $F_i^{(2)}$ seja igual a 1, obtem-se o sistema

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & -h & 0 & h \\ 1 & 1 & 1 & h^2/2! & 0 & h^2/2! \\ -h & 0 & h & -h^3/3! & 0 & h^3/3! \\ h^2/2! & 0 & h^2/2! & h^4/4! & 0 & h^4/4! \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} L_{i-1} \\ L_i \\ L_{i+1} \\ R_{i-1} \\ R_i \\ R_{i+1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}. \quad (\text{A.7})$$

Resolvendo esse sistema, obtem-se

$$L_{i-1} = 1/10, \quad L_i = 1, \quad L_{i+1} = 1/10, \quad R_{i-1} = 12/10h^2, \quad R_i = -24/10h^2, \quad R_{i+1} = 12/10h^2.$$

Assim, (A.6) é reescrita como

$$F_{i-1}^{(2)} + 10F_i^{(2)} + F_{i+1}^{(2)} = \frac{12(f_{i-1} - 2f_i + f_{i+1})}{h^2} + O(h^4). \quad (\text{A.8})$$

Para os pontos do contorno, isto é, $i = 0$ (e seu correspondente $i = N$), o valor da derivada pode ser obtido a partir da relação

$$F_0^{(2)} + \alpha F_1^{(2)} = \frac{1}{h^2}(af_0 + bf_1 + cf_2 + df_3 + ef_4). \quad (\text{A.9})$$

Expandindo $f_i, F_i^{(2)}$ em série de Taylor em torno do ponto x_0 , a Tabela A.2 apresenta os coeficientes a, b, c, d, e e α para esquemas de terceira e quarta ordens.

Tabela A.2: Coeficientes do esquema numérico para aproximar a segunda derivada no contorno.

	a	b	c	d	e	α
3ª ordem	13	-27	15	-1	0	11
4ª ordem	145/12	-76/3	29/2	-4/3	1/12	10

A.1.3 Análise de ordem

A seguir são apresentados os resultados usando a formulação compacta para a aproximação da primeira e segunda derivadas.

Considere a função $f(x) = \sin(x)$ definida no intervalo $0.05 \leq x \leq 2\pi + 0.05$. Sabe-se que para essa função

$$f^{(1)}(x) = \cos(x) \quad \text{e} \quad f^{(2)}(x) = -\sin(x). \quad (\text{A.10})$$

O cálculo da ordem é obtido da expressão

$$p = \frac{\log\left(\frac{\text{erro}(h_1)}{\text{erro}(h_2)}\right)}{\log\left(\frac{h_1}{h_2}\right)}. \quad (\text{A.11})$$

Para a primeira derivada, utilizou-se a combinação do estêncil central (A.4) com o estêncil não central de quarta ordem. Para a segunda derivada, utilizou-se a combinação do estêncil central (A.8) com o estêncil não central de terceira ordem. Em [4, 95, 96], os autores sugerem o uso de terceira ordem no contorno para garantir estabilidade numérica na solução da equação de advecção. No entanto, para o simples cálculo do valor da derivada, questões de estabilidade não se aplicam. Um estudo mais detalhado sobre a análise de estabilidade é feito na próxima seção. Como pode ser observado nas tabelas A.3 e A.4, o método de diferenças finitas compactas converge para a ordem esperada.

Tabela A.3: Ordem de convergência da primeira derivada.

N	Norma L_1	ordem	Norma L_2	ordem	Norma L_∞	ordem
10	3.837223e-03		6.275502e-03		1.452688e-02	
20	1.891559e-04	4.342415	4.163791e-04	3.913761	1.308354e-03	3.472900
40	7.450462e-06	4.666102	2.013292e-05	4.370270	8.803492e-05	3.893534
80	3.037420e-07	4.616411	9.145482e-07	4.460353	5.585960e-06	3.978199
160	1.372464e-08	4.468007	4.115703e-08	4.473848	3.499406e-07	3.996623
320	6.919459e-10	4.309965	1.867361e-09	4.462066	2.186859e-08	4.000178

Tabela A.4: Ordem de convergência da segunda derivada.

N	Norma L_1	ordem	Norma L_2	ordem	Norma L_∞	ordem
10	2.329521e-02		4.673968e-02		1.108763e-01	
20	2.290310e-03	3.346418	6.491884e-03	2.847939	2.093696e-02	2.404827
40	1.578221e-04	3.859172	6.319187e-04	3.360826	2.845038e-03	2.879532
80	1.010224e-05	3.965552	5.717276e-05	3.466339	3.618593e-04	2.974947
160	6.351410e-07	3.991455	5.082731e-06	3.491652	4.536154e-05	2.995887
320	3.975522e-08	3.997861	4.499035e-07	3.497916	5.670070e-06	3.000031

A.2 Análise de Fourier dos erros

A análise de Fourier permite quantificar os erros nos números de onda resolvidos e demonstra a adequabilidade de um esquema de discretização em relação à sua capacidade de representar os números de onda com precisão [99]. Para a análise de Fourier, consideramos uma função $f(x)$ periódica sobre o domínio $[0, L]$ tal que $f(0) = f(L)$. O intervalo $[0, L]$ é dividido em N intervalos de tamanho $h = L/N$. A série de Fourier, considerando-se apenas os números de onda resolvidos, é

$$f(x) = \sum_{k=-N/2}^{N/2} \hat{f}_k e^{\left(\frac{2\pi j k x}{L}\right)}, \quad (\text{A.12})$$

onde $j = \sqrt{-1}$. É conveniente introduzir um número de onda escalar $\omega = 2\pi k h / L$ e uma coordenada escalar $s = x/h$. Isso simplifica os modos de Fourier em $e^{j\omega s}$, produzindo

$$f(s) = \sum_k \hat{f}_k e^{j\omega s}. \quad (\text{A.13})$$

onde $\omega \in [0, \pi]$. Derivando $f(s)$ com respeito a s temos

$$f^{(1)}(s) = \sum_k j\omega \hat{f}_k e^{j\omega s}. \quad (\text{A.14})$$

Além disso, $f^{(1)}(s)$ também pode ser expandido em série de Fourier, produzindo

$$f^{(1)}(s) = \sum_k \hat{f}_k^{(1)} e^{j\omega s}. \quad (\text{A.15})$$

De (A.14) e (A.15), segue que

$$\hat{f}_k^{(1)} = j\omega \hat{f}_k. \quad (\text{A.16})$$

Seja $F^{(1)}(s)$ uma aproximação de diferenças finitas de $f^{(1)}(s)$. Usando expansão em série de Fourier,

$$F^{(1)}(s) = \sum_k \hat{F}_k^{(1)} e^{j\omega s}. \quad (\text{A.17})$$

O erro de discretização para a primeira derivada pode ser medido por comparação dos coeficientes de Fourier $\hat{F}_k^{(1)}$ em (A.17) com $\hat{f}_k^{(1)}$ em (A.15). Para esquemas de diferenças centrais, um número de onda modificado ω^* pode ser definido de forma que satisfaça

$$\hat{F}_k^{(1)} = j\omega^* \hat{f}_k, \quad (\text{A.18})$$

onde ω^* é diferente para cada k e para cada esquema de discretização. Quando não há nenhum erro de discretização, $\hat{F}_k^{(1)} = \hat{f}_k^{(1)}$, o que significa que $\omega^* = \omega$. Portanto, a diferença entre ω^* e ω é uma

medida do erro na aproximação para a primeira derivada.

Da mesma forma, para o cálculo da segunda derivada, ω^{**} pode ser definido como o quadrado do número de onda modificado, de forma que satisfaça

$$\hat{F}_k^{(2)} = -\omega^{**} \hat{f}_k. \quad (\text{A.19})$$

A comparação de ω^{**} com o valor de ω^2 resulta em um indicativo de erros de discretização.

Esquemas não centrais também podem ser analisados dessa forma. Nesse caso, o número de onda modificado ω^* ou ω^{**} será um número complexo, enquanto que o número de onda original ω é sempre real. As partes reais e imaginárias dos números de onda devem ser comparadas. A parte real de $(\omega^* - \omega)$ está associada aos erros de dispersão, enquanto que a parte imaginária corresponde aos erros de dissipação. Esquemas centrais produzem ω^* reais, ou seja, não há erros de dissipação. É importante notar que esta análise aplicada aos esquemas não centrais não é rigorosa [4]. É só uma maneira heurística para estimar os erros de discretização.

A análise de Fourier detalhada para os esquemas centrais pode ser encontrada em [4, 7, 94, 98]. Nas seções seguintes, apenas os esquemas não centrais serão analisados. Essa análise é relevante para o presente trabalho, uma vez que, em alguns casos, não existe uma única escolha de estêncis compactos não centrais que produz quarta ordem de precisão. Além disso, a escolha mais trivial de pontos pode não produzir um sistema viável.

A análise será realizada em uma dimensão, uma vez que nenhum termo presente em (2.1) envolve derivadas em mais de uma direção. A natureza explícita da discretização temporal permite que cada parte possa ser calculada separadamente. Além disso, o método utilizado neste trabalho para resolver a equação de Poisson também permite que as derivadas sejam calculadas separadamente.

A.2.1 Estêncil colocado para a primeira derivada

Para $\frac{\partial(u^2)}{\partial x}$ e $\frac{\partial(v^2)}{\partial y}$, as quantidades conhecidas (velocidades) são amostradas na mesma posição das desconhecidas (derivadas), ou seja, no ponto médio da aresta na célula (Figura 2.1). Um estêncil colocado como o da Figura A.1a é necessário. Para células no interior, a aproximação é dada por

$$F_{i-1}^{(1)} + 4F_i^{(1)} + F_{i+1}^{(1)} = \frac{3(f_{i+1} - f_{i-1}))}{h}. \quad (\text{A.20})$$

Próximo do contorno (Figura A.2a), o estêncil tem a forma

$$F_i^{(1)} + \alpha F_{i\pm 1}^{(1)} = \pm \frac{1}{h} (c_0 f_i + c_1 f_{i\pm 1} + c_2 f_{i\pm 2} + c_3 f_{i\pm 3}), \quad (\text{A.21})$$

onde o ponto x_i está sobre o contorno.

Ao combinar os coeficientes das expansões de séries de Taylor dos termos em (A.21) (ver [4, 23, 98] para mais detalhes), esquemas de terceira e de quarta ordem podem ser derivados. Eles encontram-se resumidos na Tabela A.5.

Substituindo as equações (A.13), (A.17) e (A.18) em (A.21) tem-se uma expressão para ω^* em função de ω :

$$\omega^*(\omega) = \frac{c_0 + c_1 e^{j\omega} + c_2 e^{j2\omega} + c_3 e^{j3\omega}}{1 + \alpha e^{j\omega}}. \quad (\text{A.22})$$

O comportamento desta função pode ser observado na Figura A.3 para os estênceis da Tabela A.5.

Tabela A.5: Coeficientes dos estênceis colocados no contorno para a primeira derivada.

	α	c_0	c_1	c_2	c_3
3ª ordem	2	-5/2	2	1/2	0
4ª ordem	3	-17/6	3/2	3/2	-1/6

Figura A.3: Análise de Fourier da primeira derivada para esquemas em malha colocada uniforme.

A partir da análise da parte real do número de onda modificado, pode-se observar que o esquema de quarta ordem é mais dispersivo do que o esquema de terceira ordem para altos valores do número de onda. Embora essa análise mostre um melhor comportamento do esquema de terceira ordem, a partir da análise da parte imaginária do número de onda, pode ser observado que o esquema de terceira ordem é mais dissipativo do que o esquema de quarta ordem. Portanto, o esquema de quarta ordem foi escolhido para ser utilizado no presente trabalho.

A.2.2 Estêncil deslocado para a primeira derivada

Quando aproximamos os termos $\frac{\partial(uv)}{\partial x}$ e $\frac{\partial(uv)}{\partial y}$, é necessário interpolar as velocidades para os cantos das células. Isso é realizado utilizando um esquema compacto, dado por

$$f_{i-1/2} + 6f_{i+1/2} + f_{i+3/2} = 4(f_i + f_{i+1}) + \mathcal{O}(h^4). \quad (\text{A.23})$$

No estêncil deslocado, as quantidades conhecidas (componentes de velocidade) não são amostrados nas mesmas posições das incógnitas (derivadas). Um estêncil deslocado é necessário, como

mostrado na Figura A.1b. Para as células do interior, utiliza-se a fórmula

$$F_{i-1}^{(1)} + 22F_i^{(1)} + F_{i+1}^{(1)} = \frac{24(f_{i+1/2} - f_{i-1/2})}{h}. \quad (\text{A.24})$$

Próximo ao contorno (Figura A.2c), o estêncil tem a forma

$$F_i^{(1)} + \alpha F_{i\pm 1}^{(1)} = \pm \frac{1}{h} (c_0 f_{i\mp 1/2} + c_1 f_{i\pm 1/2} + c_2 f_{i\pm 3/2} + c_3 f_{i\pm 5/2}), \quad (\text{A.25})$$

onde o ponto $x_{i\mp 1/2}$ esta sobre o contorno.

A partir da expressão acima, um esquema de quarta ordem não pode ser derivado, pois o sistema resultante da expansão em série de Taylor não possui solução. A solução é tomar um ponto adicional na extrema direita (ou esquerda), e remover um ponto intermediário do estêncil (A.25). O estêncil resultante teria uma lacuna, mas ainda produzirá aproximações precisas.

A expressão geral é

$$F_i^{(1)} + \alpha F_{i\pm 1}^{(1)} = \pm \frac{1}{h} (c_0 f_{i\mp 1/2} + c_1 f_{i\pm 1/2} + c_2 f_{i\pm 3/2} + c_3 f_{i\pm 5/2} + c_4 f_{i\pm 7/2}), \quad (\text{A.26})$$

onde um dos $c_i, i = 1, \dots, 4$ será zero.

Ao combinar os coeficientes das expansões de séries de Taylor dos termos em (A.26), esquemas de terceira e de quarta ordem podem ser derivados, e estão resumidos na Tabela A.6.

Substituindo as equações (A.13), (A.17) e (A.18) em (A.26) tem-se uma expressão para ω^* em função de ω :

$$\omega^*(\omega) = \frac{c_0 e^{-j\frac{\omega}{2}} + c_1 e^{j\frac{\omega}{2}} + c_2 e^{j\frac{3}{2}\omega} + c_3 e^{j\frac{5}{2}\omega} + c_4 e^{j\frac{7}{2}\omega}}{1 + \alpha e^{j\omega}}. \quad (\text{A.27})$$

O comportamento desta função pode ser observado na Figura A.4 para os estênceis da Tabela A.6.

Tabela A.6: Coeficientes dos estênceis deslocados no contorno para a primeira derivada.

	α	c_0	c_1	c_2	c_3	c_4
3ª ordem	-1	-1	2	-1	0	0
4ª ordem A	-5	-9/8	19/3	-21/4	0	1/24
4ª ordem B	-1/3	-67/72	13/12	0	-7/36	1/24

A análise do número de onda modificado (parte real) mostra que o esquema de quarta ordem A exibe um melhor comportamento quando comparado com os outros sistemas. Esse comportamento resulta em um menor valor de dispersão para elevados número de onda. Da análise da parte imaginária, é observado que o esquema de quarta ordem B é o mais dissipativo entre todos, e, embora o esquema de terceira ordem mostre um comportamento melhor, a diferença entre o esquema de quarta ordem A e os esquemas de terceira ordem é pequena. Como o esquema de quarta ordem A mostrou-se menos dispersivo este foi a escolha deste trabalho.

Figura A.4: Análise de Fourier da primeira derivada para esquemas em malha deslocada.

A.2.3 Estêncil colocado uniforme para a segunda derivada

Para $\frac{\partial^2 u}{\partial x^2}$ e $\frac{\partial^2 v}{\partial y^2}$, as quantidades conhecidas (componentes da velocidade) são amostradas na mesma posição das desconhecidas (derivadas), ou seja, no ponto médio da aresta da célula (Figura 2.1). Um estêncil colocado é necessário, como aquele mostrado na Figura A.1a. Para células do interior, utiliza-se

$$F_{i-1}^{(2)} + 10F_i^{(2)} + F_{i+1}^{(2)} = \frac{12(f_{i-1} - 2f_i + f_{i+1}))}{h^2}. \quad (\text{A.28})$$

Próximo do contorno (Figura A.2a), o estêncil a ser utilizado é

$$F_i^{(2)} + \alpha F_{i\pm 1}^{(2)} = \frac{1}{h^2}(c_0 f_i + c_1 f_{i\pm 1} + c_2 f_{i\pm 2} + c_3 f_{i\pm 3} + c_4 f_{i\pm 4}), \quad (\text{A.29})$$

onde o ponto x_i está sobre o contorno.

Ao combinar os coeficientes das expansões de séries de Taylor dos termos acima, esquemas de terceira e de quarta ordens são derivados. Eles encontram-se resumidos na Tabela A.7.

Substituindo as equações (A.13) e (A.19) em (A.29) tem-se uma expressão para ω^{**} em função de ω :

$$\omega^{**}(\omega) = \frac{c_b e^{-j\omega} + c_0 + c_1 e^{j\omega} + c_2 e^{j2\omega} + c_3 e^{j3\omega} + c_4 e^{j4\omega}}{1 + \alpha e^{j\omega}}. \quad (\text{A.30})$$

O comportamento desta função pode ser observado na Figura A.5 para os estênceis da Tabela A.7. Ambas as partes real e imaginária do número de onda modificado mostram que o esquema de diferenças finitas de quarta ordem proposto é menos dispersivo e dissipativo que o esquema de terceira ordem.

A.2.4 Estêncil colocado não uniforme para a segunda derivada

Para os termos viscosos $\frac{\partial^2 u}{\partial y^2}$ e $\frac{\partial^2 v}{\partial x^2}$ o estêncil é colocado no interior de domínio. No entanto, os pontos do contorno estão a uma distância com metade do espaçamento de malha em relação à derivada,

Tabela A.7: Coeficientes dos estênceis colocados no contorno para a segunda derivada.

	α	c_0	c_1	c_2	c_3	c_4
3ª ordem	11	13	-27	15	-1	0
4ª ordem	10	145/12	-76/3	29/2	-4/3	1/12

Figura A.5: Análise de Fourier da segunda derivada para esquemas em malha colocada uniforme.

conforme mostrado na Figura A.2b. Portanto, é necessário um esquema não uniforme. E, este tem a forma

$$F_i^{(2)} + \alpha F_{i\pm 1}^{(2)} = \frac{1}{h^2} (c_b f_{i\mp 1/2} + c_0 f_i + c_1 f_{i\pm 1} + c_2 f_{i\pm 2} + c_3 f_{i\pm 3}), \quad (\text{A.31})$$

onde o ponto $x_{i\mp 1/2}$ esta sobre o contorno.

Novamente, a escolha mais simples de pontos não produz um esquema viável, por isso deve-se considerar estênceis com uma lacuna.

A expressão geral é

$$F_i^{(2)} + \alpha F_{i\pm 1}^{(2)} = \frac{1}{h} (c_b f_{i\mp 1/2} + c_0 f_i + c_1 f_{i\pm 1} + c_2 f_{i\pm 2} + c_3 f_{i\pm 3} + c_4 f_{i\pm 4}). \quad (\text{A.32})$$

Ao combinar os coeficientes das expansões de séries de Taylor dos termos em (A.32), esquemas de terceira e de quarta ordens podem ser derivados. Eles encontram-se resumidos na Tabela A.8.

Substituindo as equações (A.13) e (A.19) em (A.32) tem-se uma expressão para ω^{**} em função de ω :

$$\omega^{**}(\omega) = \frac{c_b e^{-j\frac{\omega}{2}} + c_0 + c_1 e^{j\omega} + c_2 e^{j2\omega} + c_3 e^{j3\omega} + c_4 e^{j4\omega}}{1 + \alpha e^{j\omega}}. \quad (\text{A.33})$$

O comportamento desta função pode ser observado na Figura A.6 para os estênceis de Tabela A.8. A parte real do número de onda modificado, para todos os sistemas testados, ultrapassa a curva exata. O único esquema que, eventualmente, se aproxima da curva exata é o esquema de quarta ordem C. Uma vez que todos os esquemas são igualmente dissipativos ao analisar a parte imaginária, o esquema

de quarta ordem C foi escolhido.

Tabela A.8: Coeficientes dos estênceis colocados não uniformes no contorno para a segunda derivada.

	α	c_b	c_0	c_1	c_2	c_3	c_4
3 ^a ordem	1/2	16/5	-9/2	1	3/10	0	0
4 ^a ordem A	1	64/21	-11/3	-1/3	1	-1/21	0
4 ^a ordem B	2/3	256/81	-17/4	16/27	1/2	0	-1/324
4 ^a ordem C	1/3	1856/567	-29/6	41/27	0	1/21	-1/162
4 ^a ordem D	22/25	4864/1575	-1163/300	0	41/50	-16/525	-1/900

(a) Parte real.

(b) Parte imaginária.

Figura A.6: Análise de Fourier da segunda derivada para esquemas em malha colocada não uniforme.

Filtro

Discretizações de diferenças finitas compactas, dentre outros esquemas centrais, não são dissipativos e são suscetíveis a instabilidades devido ao crescimento das altas frequências. A fim de eliminar essas altas frequências, pode-se usar uma técnica de filtragem, como sugerido em [4].

Seja uma componente do vetor solução, denotada por f , os valores filtrados $\hat{f}$ são obtidos resolvendo o sistema tridiagonal

$$\alpha \hat{f}_{i-1} + \hat{f}_i + \alpha \hat{f}_{i+1} = a f_i + \frac{b}{2}(f_{i+1} + f_{i-1}) + \frac{c}{2}(f_{i+2} + f_{i-2}) + \frac{d}{2}(f_{i+3} + f_{i-3}). \quad (\text{B.1})$$

Atribuindo um valor para α , uma aproximação de sexta ordem pode ser obtida por

$$a = \frac{1}{16}(11 + 10\alpha), \quad (\text{B.2a})$$

$$b = \frac{1}{32}(15 + 34\alpha), \quad (\text{B.2b})$$

$$c = \frac{1}{16}(-3 + 6\alpha), \quad (\text{B.2c})$$

$$d = \frac{1}{32}(1 - 2\alpha). \quad (\text{B.2d})$$

Os pontos próximos ao contorno devem ser tratados separadamente. Uma aproximação explícita de quarta ordem em geral é usada

$$\hat{f}_1 = \frac{15}{16}f_1 + \frac{1}{16}(4f_2 - 6f_3 + 4f_4 - f_5), \quad (\text{B.3a})$$

$$\hat{f}_2 = \frac{3}{4}f_2 + \frac{1}{16}(f_1 + 6f_3 - 4f_4 + f_5), \quad (\text{B.3b})$$

$$\hat{f}_3 = \frac{5}{8}f_3 + \frac{1}{16}(-f_1 + 4f_2 + 4f_4 - f_5). \quad (\text{B.3c})$$

Usando a solução manufaturada

$$U(x, y) = \sin(2\pi x) \sin(2\pi y), \quad (\text{B.4a})$$

$$V(x, y) = \cos(2\pi x) \cos(2\pi y), \quad (\text{B.4b})$$

$$P(x, y) = \sin(2\pi x) \cos(2\pi y), \quad (\text{B.4c})$$

a Tabela B.1 mostra a ordem de convergência na aplicação do filtro definido acima, considerando $\alpha = 0.48$.

Tabela B.1: Ordem de convergência na aplicação do filtro na direção x .

N	Norma L_1	ordem	Norma L_2	ordem	Norma L_∞	ordem
25x25	8.821785e-07		1.762298e-06		7.473835e-06	
50x50	1.405086e-08	5.972341	3.988359e-08	5.465519	2.419678e-07	4.948962
100x100	1.235138e-09	5.996819	4.304126e-09	5.490955	3.207206e-08	4.983953
200x200	2.198899e-10	5.999021	8.854628e-10	5.496411	7.628177e-09	4.992096

É interessante notar que apesar de o esquema de borda ser de quarta ordem, o método atinge quinta ordem na norma L_∞ .

Referências Bibliográficas

- [1] S. Gottlieb, D. I. Ketcheson, and C-W. Shu. *Strong stability preserving Runge-Kutta and multistep time discretizations*. World Scientific, 2011.
- [2] M. N. Linnick and H. F. Fasel. A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains. *Journal of Computational Physics*, 204:157–192, 2005.
- [3] J. L. Hennessy and D. A. Patterson. *Computer architecture: a quantitative approach*. Elsevier, 2012.
- [4] S. K. Lele. Compact finite difference schemes with spectral-like resolution. *Journal of Computational Physics*, 103:16–42, 1992.
- [5] Y. Adam. Highly accurate compact implicit methods and boundary conditions. *Journal of Computational Physics*, 24(1):10–22, 1977.
- [6] R. S. Hirsh. Higher order accurate difference solutions of fluid mechanics problems by a compact differencing technique. *Journal of computational physics*, 19(1):90–109, 1975.
- [7] L. F. Souza, M. T. Mendonça, and M. A. F. Medeiros. The advantages of using high-order finite differences schemes in laminar–turbulent transition studies. *International journal for numerical methods in fluids*, 48(5):565–582, 2005.
- [8] M. Li, T. Tang, and B. Fornberg. A compact fourth-order finite difference scheme for the steady incompressible Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 20(10):1137–1151, 1995.
- [9] Z. Tian and Y. Ge. A fourth-order compact finite difference scheme for the steady stream function–vorticity formulation of the Navier–Stokes/Boussinesq equations. *International journal for numerical methods in fluids*, 41(5):495–518, 2003.

- [10] J. Wang, W. Zhong, and J. Zhang. High order compact computation and nonuniform grids for streamfunction vorticity equations. *Applied mathematics and computation*, 179(1):108–120, 2006.
- [11] B. A. DeVantier. A comparison of primitive variables and streamfunction–vorticity for FEM depth-averaged modelling of steady flow. *International Journal for Numerical Methods in Fluids*, 9(11):1369–1379, 1989.
- [12] A. J. Chorin. Numerical solution of the Navier-Stokes equations. *Mathematics of computation*, 22(104):745–762, 1968.
- [13] R. Temam. Sur l’approximation de la solution des équations de Navier-Stokes par la méthode des pas fractionnaires (II). *Archive for Rational Mechanics and Analysis*, 33(5):377–385, 1969.
- [14] J. Kim and P. Moin. Application of a fractional-step method to incompressible navier-stokes equations. *Journal of computational physics*, 59(2):308–323, 1985.
- [15] J. B. Bell, P. Colella, and H. M. Glaz. A second-order projection method for the incompressible navier-stokes equations. *Journal of Computational Physics*, 85(2):257–283, 1989.
- [16] D. L. Brown, R. Cortez, and M. L. Minion. Accurate projection methods for the incompressible Navier–Stokes equations. *Journal of Computational Physics*, 168(2):464–499, 2001.
- [17] J. C. Strikwerda. Finite difference methods for the stokes and navier-stokes equations. *SIAM Journal on Scientific and Statistical Computing*, 5(1):56–68, 1984.
- [18] P. M. Gresho. On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduces a nearly consistent mass matrix. part 1: Theory. *International Journal for Numerical Methods in Fluids*, 11(5):587–620, 1990.
- [19] P. M. Gresho and S. T. Chan. On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduces a nearly consistent mass matrix. part 2: Implementation. *International Journal for Numerical Methods in Fluids*, 11(5):621–659, 1990.
- [20] S. Abide and S. Viazzo. A 2D compact fourth-order projection decomposition method. *Journal of Computational Physics*, 206(1):252–276, 2005.
- [21] R. Knikker. Study of a staggered fourth-order compact scheme for unsteady incompressible viscous flows. *International journal for numerical methods in fluids*, 59(10):1063–1092, 2009.

- [22] B. J. Boersma. A 6th order staggered compact finite difference method for the incompressible Navier–Stokes and scalar transport equations. *Journal of Computational Physics*, 230(12):4940–4954, 2011.
- [23] K. Zhang, M. Yang, and Y. Zhang. A compact finite-difference scheme based on the projection method for natural-convection heat transfer. *Numerical Heat Transfer, Part B: Fundamentals*, 61(4):259–278, 2012.
- [24] C. S. Peskin. Flow patterns around heart valves: A numerical method. *Journal of Computational Physics*, 10:252–271, 1972.
- [25] R. J. LeVeque and Z. Li. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM Journal on Numerical Analysis*, 31:1019–1044, 1994.
- [26] R. J. LeVeque and Z. Li. Immersed interface methods for Stokes flow with elastic boundaries or surface tension. *SIAM Journal on Scientific Computing*, 18(3):709–735, 1997.
- [27] Z. Li. *The Immersed Interface Method – A Numerical Approach for Partial Differential Equations with Interfaces*. PhD thesis, Citeseer, 1994.
- [28] Z. Li. A note on immersed interface method for three-dimensional elliptic equations. *Computers & Mathematics with Applications*, 31(3):9–17, 1996.
- [29] D. A. Von Terzi, M. N. Linnick, J. Seidel, and H. F. Fasel. Immersed boundary techniques for high-order finite-difference methods. *AIAA paper*, 2918:2001, 2001.
- [30] Z. Li and K. Ito. The immersed interface method - numerical solutions of PDEs involving interfaces and irregular domains. *SIAM*, 2006.
- [31] Z. Li and M. C. Lai. The immersed interface method for the Navier–Stokes equations with singular forces. *Journal of Computational Physics*, 171(2):822–842, 2001.
- [32] D.V. Le, B. C. Khoo, and J. Peraire. An immersed interface method for viscous incompressible flows involving rigid and flexible boundaries. *Journal of Computational Physics*, 220(1):109–138, 2006.
- [33] Z. Tan, K. M. Lim, and B. C. Khoo. A fast immersed interface method for solving Stokes flows on irregular domains. *Computers & Fluids*, 38(10):1973–1983, 2009.
- [34] A. Wiegmann and K. P. Bube. The explicit-jump immersed interface method: finite difference methods for pdes with piecewise smooth solutions. *SIAM Journal on Numerical Analysis*, 37:827–862, 2000.

- [35] C. Brehm. *Novel immersed interface method for solving the incompressible navier-stokes equations*. PhD thesis, University of Arizona, 2011.
- [36] C. Brehm and F. H. Fasel. A novel concept for the design of immersed interface methods. *Journal of Computational Physics*, 242:234–267, 2013.
- [37] X. D. Liu, R. P. Fedkiw, and M. Kang. A boundary condition capturing method for Poisson’s equation on irregular domains. *Journal of computational Physics*, 160(1):151–178, 2000.
- [38] A. N. Marques, J. C. Nave, and R. R. Rosales. A correction function method for Poisson problems with interface jump conditions. *Journal of Computational Physics*, 230(20):7567–7597, 2011.
- [39] G. A. Reis, I. V. M. Tasso, L. F. Souza, and J. A. Cuminato. A compact finite differences exact projection method for the Navier-Stokes equations on a staggered grid with fourth-order spatial precision. *Computers & Fluids*, 2015.
- [40] G. A. Reis, I. V. M. Tasso, L. F. Souza, and J. A. Cuminato. Um novo tipo de método de projeção com diferenças finitas compactas de alta ordem. *III CMAC - Sudeste 2015 - Congresso de Matemática Aplicada e Computacional, Vitória-ES*, 2015.
- [41] G. A. Reis, I. V. M. Tasso, L. F. Souza, and J. A. Cuminato. Resolução numérica das equações de Navier-Stokes usando o método das interfaces imersas com correção do salto implícita em malha deslocada. *XXXV CNMAC - Congresso Nacional de Matemática Aplicada e Computacional, Natal-RN.*, 2014.
- [42] G. A. Reis, I. V. M. Tasso, L. F. Souza, and J. A. Cuminato. Fast and precise solution for the Poisson equation in the presence of interfaces with discontinuities. *22nd International Congress of Mechanical Engineering, Ribeirão Preto*, 2013.
- [43] A. Chorin. A numerical method for solving incompressible viscous flow problems. *J. Comp. Physics*, 2:12–26, 1967.
- [44] A. Chorin. Numerical solution of the Navier-Stokes. *Mathematics of Computation*, 2:745–762, 1968.
- [45] A. Chorin and J. E. Marsden. A mathematical introduction to fluid mechanics. *Springer*, 2000.
- [46] K. Kim, S-J. Baek, and H. J. Sung. An implicit velocity decoupling procedure for the incompressible Navier-Stokes equations. *International journal for numerical methods in fluids*, 38(2):125–138, 2002.
- [47] R. M. Beam and R. RF. Warming. An implicit factored scheme for the compressible Navier-Stokes equations. *AIAA journal*, 16(4):393–402, 1978.

- [48] A. O. Fortuna. Técnicas computacionais para dinâmica dos fluidos: conceitos básicos e aplicações. *Edusp*, 2000.
- [49] F. H. Harlow and J. E. Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of Fluids*, 8:2182–2189, 1965.
- [50] A. Amsden and F. H. Harlow. A simplified MAC technique for incompressible fluid flow calculations. *J. Comp. Physics*, 6:332–335, 1970.
- [51] J. H. Ferziger and M. Perić. *Computational methods for fluid dynamics*. Springer-Verlag, 1999.
- [52] S. Patankar. *Numerical heat transfer and fluid flow*. CRC Press, 1980.
- [53] R. L. Burden and J. D. Faires. Numerical analysis. 2001. *Brooks/Cole, USA*, 2001.
- [54] D. L. Brown, R. Cortez, and M. L. Minion. Accurate projection methods for the incompressible Navier-Stokes equations. *J. Comp. Physics*, 168:464–499, 2001.
- [55] J. B. Bell, P. Colella, and L. H. Howell. An efficient second-order projection method for viscous incompressible flow. In *Proceedings of the Tenth AIAA Computational Fluid Dynamics Conference, AIAA*, volume 360, 1991.
- [56] J-L. Guermond and L. Quartapelle. On stability and convergence of projection methods based on pressure Poisson equation. *International Journal for Numerical Methods in Fluids*, 26(9):1039–1053, 1998.
- [57] F. S. Sousa, C. M. Oishi, and G. C. Buscaglia. Spurious transients of projection methods in microflow simulations. *Computer Methods in Applied Mechanics and Engineering*, 285:659–693, 2015.
- [58] C. Pozrikidis. A note on the regularization of the discrete Poisson–Neumann problem. *Journal of Computational Physics*, 172(2):917–923, 2001.
- [59] J. C. Strikwerda. *Finite difference schemes and partial differential equations*. Siam, 2004.
- [60] C. Pozrikidis. *Introduction to theoretical and computational fluid dynamics*. Oxford University Press, 2011.
- [61] I. Fredholm. Sur une classe d’équations fonctionnelles. *Acta mathematica*, 27(1):365–390, 1903.
- [62] C. J. Roy. Review of code and solution verification procedures for computational simulation. *Journal of Computational Physics*, 205(1):131–156, 2005.
- [63] K. Salari and P. Knupp. Code verification by the method of manufactured solutions. 2000.

- [64] T. M. Shih, C. H. Tan, and B. C. Hwang. Effects of grid staggering on numerical schemes. *International Journal for Numerical Methods in Fluids*, 9(2):193–212, 1989.
- [65] B. Sanderse and B. Koren. Accuracy analysis of explicit Runge–Kutta methods applied to the incompressible Navier–Stokes equations. *Journal of Computational Physics*, 231(8):3041–3063, 2012.
- [66] U. K. N. G. Ghia, K. N. Ghia, and C. T. Shin. High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *Journal of computational physics*, 48(3):387–411, 1982.
- [67] V. C. Loukopoulos, G. T. Messaris, and G. C. Bourantas. Numerical solution of the incompressible Navier–Stokes equations in primitive variables and velocity–vorticity formulation. *Applied Mathematics and Computation*, 222:575–588, 2013.
- [68] E. J. Barragy. Parallel finite element methods and iterative solution techniques for viscous incompressible flows. 1993.
- [69] P. P. Chinchapatnam, K. Djidjeli, and P. B. Nair. Radial basis function meshless method for the steady incompressible Navier–Stokes equations. *International Journal of Computer Mathematics*, 84(10):1509–1521, 2007.
- [70] G. C. Bourantas, E. D. Skouras, V. C. Loukopoulos, and G. C. Nikiforidis. Meshfree point collocation schemes for 2D steady state incompressible Navier-Stokes equations in velocity-vorticity formulation for high values of Reynolds number. *Computer Modeling in Engineering & Sciences(CMES)*, 59(1):31–64, 2010.
- [71] J. D. Lambert. *Computational methods in ordinary differential equations*. Wiley London, 1973.
- [72] W. F. Ames. *Numerical methods for partial differential equations*. Academic press, 2014.
- [73] E. Hairer and G. Wanner. Stiff differential equations solved by Radau methods. *Journal of Computational and Applied Mathematics*, 111(1):93–111, 1999.
- [74] J. Crank and P. Nicolson. A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 43, pages 50–67. Cambridge Univ Press, 1947.
- [75] E. Süli and D. F. Mayers. *An introduction to numerical analysis*. Cambridge university press, 2003.
- [76] G. Wanner. Dahlquist’s classical papers on stability theory. *BIT Numerical Mathematics*, 46(3):671–683, 2006.

- [77] G. G. Dahlquist. Stability questions for some numerical methods for ordinary differential equations. Technical report, DTIC Document, 1962.
- [78] C. A. Kennedy and M. H. Carpenter. Additive Runge-Kutta schemes for convection-diffusion-reaction equations. *Applied Numerical Mathematics*, page 139â181, 2003.
- [79] E. Fehlberg. Klassische runge-kutta-formeln vierter und niedrigerer ordnung mit schrittweiten-kontrolle und ihre anwendung auf waermeleitungsprobleme. *Computing*, 6(1-2):61–71, 1970.
- [80] P. M. Gresho and R. L. Sani. Incompressible flow and the finite element method: Vol. 1: Isothermal laminar flow. *John Wiley, Chichester, UK*, 2000.
- [81] P. M. Gresho and R. L. Sani. Incompressible flow and the finite element method: Vol. 2: Isothermal laminar flow. *John Wiley, Chichester, UK*, 2000.
- [82] P. M. Gresho, D. F. Griffiths, and D. J. Silvester. Adaptative time-stepping for incompressible flow. Part I: Scalar advection-diffusion. *SIAM J. Sci. Comput.*, 30:2018–2054, 2008.
- [83] D. A. Kay, P. M. Gresho, D. F. Griffiths, and D. J. Silvester. Adaptative time-stepping for incompressible flow. Part II: Navier-Stokes equations. *SIAM J. Sci. Comput.*, 32:111–128, 2010.
- [84] G. A. Reis. Métodos com passo temporal adaptativo para a simulação de escoamentos com superfícies livres. Master’s thesis, Universidade de São Paulo, 2012.
- [85] A. Murua. Formal series and numerical integrators, Part I: Systems of ODEs and symplectic integrators. *Applied numerical mathematics*, 29(2):221–251, 1999.
- [86] A. L. Araújo, A. Murua, and J. M Sanz-Serna. Symplectic methods based on decompositions. *SIAM journal on numerical analysis*, 34(5):1926–1947, 1997.
- [87] J. C. Butcher. On the implementation of implicit Runge-Kutta methods. *BIT Numerical Mathematics*, 16(3):237–240, 1976.
- [88] E. Hairer, S. P. Norsett, and G. Wanner. Solving ordinary differential equations I: Nonstiff problems. *Computational Mathematics, Springer, Berlin*, 8, 2000.
- [89] S Balay, S Abhyankar, M Adams, J Brown, P Brune, K Buschelman, V Eijkhout, W Gropp, D Kaushik, M Knepley, et al. PETSc users manual revision 3.5. Technical report, Argonne National Laboratory (ANL), 2014.
- [90] E. A. Fadlun, R. Verzicco, P. Orlandi, and J. Mohd-Yusof. Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations. *Journal of Computational Physics*, 161(1):35–60, 2000.

- [91] Y-H. Tseng and J. H. Ferziger. A ghost-cell immersed boundary method for flow in complex geometry. *Journal of computational physics*, 192(2):593–623, 2003.
- [92] P. A. Berthelsen and O. M. Faltinsen. A local directional ghost cell approach for incompressible viscous flow problems with irregular boundaries. *Journal of computational physics*, 227(9):4354–4397, 2008.
- [93] M. A. Ol’Shanskii and V. M. Staroverov. On simulation of outflow boundary conditions in finite difference calculations for incompressible fluid. *International Journal for Numerical Methods in Fluids*, 33(4):499–534, 2000.
- [94] Z. Kopal. Numerical analysis: with emphasis on the application of numerical techniques to problems of infinitesimal calculus in single variable. *Chapman & Hall*, 1961.
- [95] M. H. Carpenter, D. Gottlieb, and S. Abarbanel. Stable and accurate boundary treatments for compact, high-order finite-difference schemes. *Applied Numerical Mathematics*, 12(1):55–87, 1993.
- [96] K. Mahesh. A family of high order finite difference schemes with good spectral resolution. *Journal of Computational Physics*, 145(1):332–358, 1998.
- [97] J. A. Trapp and J. D. Ramshaw. A simple heuristic method for analyzing the effect of boundary conditions on numerical stability. *Journal of Computational Physics*, 20(2):238–242, 1976.
- [98] K. K. Q. Zhang, B. Shotorban, W. J. Minkowycz, and F. Mashayek. A compact finite difference method on staggered grid for Navier–Stokes flows. *International journal for numerical methods in fluids*, 52(8):867–881, 2006.
- [99] S. Nagarajan, S. K. Lele, and J. H. Ferziger. A robust high-order compact method for large eddy simulation. *Journal of Computational Physics*, 191(2):392–419, 2003.