



Universidade do Estado do Rio de Janeiro

Faculdade de Engenharia Mecânica

Wagner Rodrigues Fortes

**Precondicionadores e Solucionadores para resolução
de sistemas lineares obtidos de simulação de
enchimento de reservatórios**

Rio de Janeiro
2008

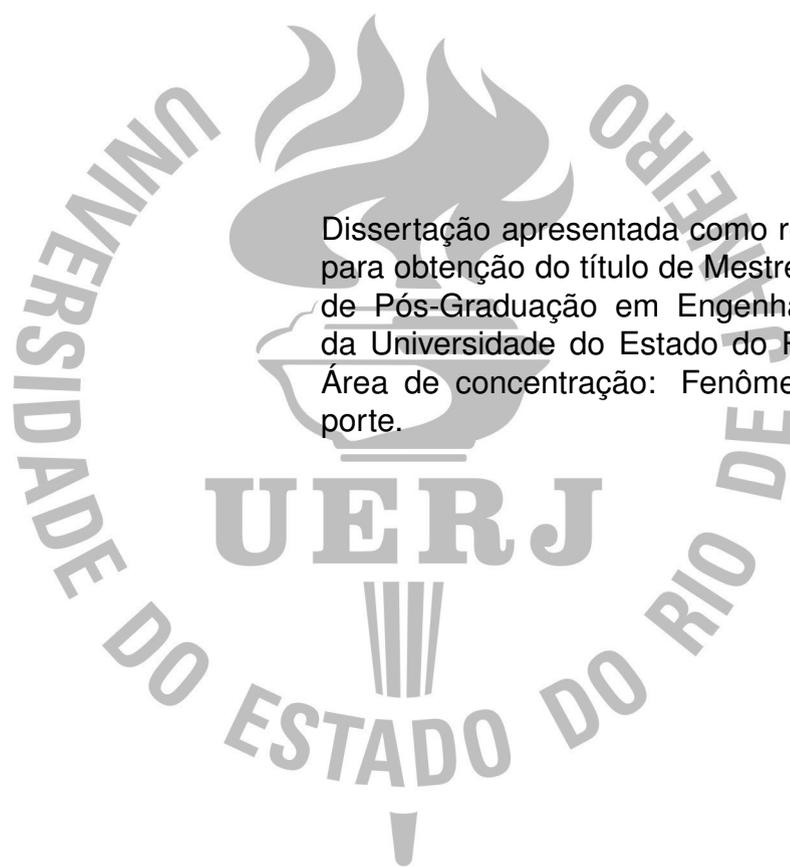
Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

Wagner Rodrigues Fortes

**Precondicionadores e Solucionadores para resolução
de sistemas lineares obtidos de simulação de
enchimento de reservatórios**



Dissertação apresentada como requisito parcial para obtenção do título de Mestre, ao Programa de Pós-Graduação em Engenharia Mecânica, da Universidade do Estado do Rio de Janeiro. Área de concentração: Fenômenos de Transporte.

Orientador: Prof. Dr. Luiz Mariano Paes de Carvalho Filho
Co-orientador: Prof. Dr. Carlos Antônio de Moura

Rio de Janeiro
2008

CATALOGAÇÃO NA FONTE
UERJ/REDE SIRIUS/BIBLIOTECA CTC/B

C837

Fortes, W. R.

Precondicionadores e Solucionadores para resolução de sistemas lineares obtidos de simulação de enchimento de reservatórios / Wagner Rodrigues Fortes. – 2008.

106 f.: il.

Orientador: Luiz Mariano Paes de Carvalho Filho.

Co-orientador: Carlos Antônio de Moura.

Dissertação (Mestrado) – Universidade do Estado do Rio de Janeiro, Faculdade de Engenharia Mecânica.

Bibliografia: f. 96-99.

1. Solucionadores de Krylov – Sistemas lineares não simétricos – Precondicionadores – Problema de ponto de sela generalizado – Teses. 2. Dinâmica dos fluidos – Reservatórios – Teses. I. Carvalho Filho, Luiz Mariano Paes de. II. Universidade do Estado do Rio de Janeiro. Faculdade de Engenharia. III. Título.

CDU 627.03

Autorizo, apenas para fins acadêmicos e científicos a reprodução total ou parcial desta dissertação.

Assinatura

Data

Wagner Rodrigues Fortes

**Precondicionadores e Solucionadores para resolução
de sistemas lineares obtidos de simulação de
enchimento de reservatórios**

Dissertação apresentada como requisito parcial
para obtenção do título de Mestre, ao Programa
de Pós-Graduação em Engenharia Mecânica,
da Universidade do Estado do Rio de Janeiro.
Área de concentração: Fenômenos de Trans-
porte.

Aprovado em _____

Banca Examinadora: _____

Prof. Dr. Luiz Mariano Paes de Carvalho Filho (Orientador)
Instituto de Matemática e Estatística da UERJ

Prof. Dr. Carlos Antônio de Moura (Co-orientador)
Instituto de Matemática e Estatística da UERJ

Prof. Dr. Norberto Mangiavacchi
Faculdade de Engenharia Mecânica da UERJ

Dr. Antonio Castelo Filho
Instituto de Ciências Matemáticas e de Computação da USP

AGRADECIMENTOS

Aos meus pais, Paulo Cesar Rodrigues Fortes e Angela Martins Rodrigues Fortes, por dedicar anos de suas vidas à minha educação acreditando no meu futuro. Ao meu irmão Wando Rodrigues Fortes e meus familiares que sempre estiveram por perto participando do meu crescimento pessoal e profissional.

Ao meu orientador, Prof. Luiz Mariano Paes de Carvalho e todos os meus professores, por terem me ajudado e fornecido meios para a conclusão da pós-graduação.

Aos colaboradores do Projeto GESAR.

Aos meus amigos que, tanto nos bons quanto nos maus momentos, colaboraram para tornar-me o que sou.

RESUMO

Fortes, W. R.. *Precondicionadores e Solucionadores para resolução de sistemas lineares obtidos de simulação de enchimento de reservatórios*. Brasil. 2008. 106 f. Dissertação (Mestrado em Engenharia Mecânica) – Faculdade de Engenharia Mecânica, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2008.

O presente trabalho consiste na comparação entre combinações de solucionadores, preconditionadores e outras técnicas utilizadas na resolução de sistemas lineares não simétricos, esparsos e com estrutura quase simétrica que são obtidos, em nosso caso, através da discretização das equações de Navier-Stokes que modelam o escoamento de um fluido Newtoniano e foram utilizadas para simular enchimento de reservatórios de hidrelétricas.

Após a discretização espacial e temporal, surgem grandes sistemas lineares cuja matriz dos coeficientes é uma matriz de ponto de sela generalizado. Necessitando, assim, de técnicas específicas para garantir uma convergência rápida de forma a tornar viável grandes simulações.

Assim, são apresentadas algumas formas tradicionais e alternativas de resolução desses sistemas de equações lineares variando métodos iterativos de Krylov, preconditionadores e reordenamentos. São apresentados, também, os modelos simulados de canal e de reservatório e em tabelas, os resultados comparativos de tempo e acurácia entre os métodos, preconditionadores e tamanho de malha.

Palavras-chave: Solucionadores de Krylov, Sistemas Lineares Não Simétricos, Preconditionadores, Problema de Ponto de Sela Generalizado, Dinâmica dos Fluidos.

ABSTRACT

This master thesis compares several solvers, preconditioners and others techniques in order to solve nonsymmetric sparse linear systems, with quasi-symmetric structure which are obtained, in our case, through discretization of the Navier-Stokes equations that model the flow of a Newtonian fluid and was used to simulate the flooding of hydroelectric power plant.

Before the spacial and time discretization, huge linear system arisen which the coefficient matrix is a generalized saddle point matrix. In this way, specific techniques are mandatory to ensure fast convergence and provide the feasibility of large simulations.

Therefore, some traditional and alternative resolutions are presented, to solve these algebraic linear systems of equations varying Krylov iterative solvers, preconditioners and reorderings. Simulated models of channel and reservoir are presented with comparative results about time, accuracy between methods, preconditioners and size of mesh.

Keywords: Krylov Solvers, Nonsymmetric linear systems, Preconditioners, Generalized Saddle Point Problem, Fluid Dynamics.

LISTA DE FIGURAS

1	Exemplo de estrutura de uma matriz de ponto de sela	15
2	Exemplo: Matriz original A , L e U , respectivamente	49
3	Exemplo: A matriz do complemento de Schur é densa.	60
4	Domínio para o escoamento em um canal.	69
5	Domínio para o escoamento em um degrau.	69
6	Exemplo de malha utilizada na simulação do compartimento de reservatório.	70
7	Decaimento do resíduo para alguns preconditionadores utilizando o PE-GMRes.	71
8	Decaimento do resíduo para alguns preconditionadores utilizando o PD-GMRes.	71
9	Decaimento do resíduo para alguns preconditionadores utilizando o Bi-CGStab.	72
10	Decaimento do resíduo para o preconditionador diagonal utilizando os três solucionadores.	72
11	Exemplo: Reordenamentos Cuthill-McKee reverso e grau mínimo.	75

LISTA DE TABELAS

1	Melhores tempos em cada problema tratado, CFL=1	76
2	Melhores tempos em cada problema tratado, CFL=5	76
3	Ranking segregado/acoplado canal, $20 \times 10 \times 3$ e CFL=1	83
4	Ranking canal acoplado, $20 \times 10 \times 3$ e CFL=1	83
5	Ranking segregado/acoplado canal, $50 \times 20 \times 3$ e CFL=1	83
6	Ranking canal acoplado, $50 \times 20 \times 3$ e CFL=1	84
7	Ranking segregado/acoplado canal, $20 \times 10 \times 3$ e CFL=5	84
8	Ranking canal acoplado, $20 \times 10 \times 3$ e CFL=5	84
9	Ranking segregado/acoplado canal, $50 \times 20 \times 3$ e CFL=5	84
10	Ranking canal acoplado, $50 \times 20 \times 3$ e CFL=5	84
11	Ranking segregado/acoplado degrau, $10 \times 5 \times 3$ e CFL=1	85
12	Ranking degrau acoplado, $10 \times 5 \times 3$ e CFL=1	85
13	Ranking segregado/acoplado degrau, $20 \times 10 \times 3$ e CFL=1	85
14	Ranking degrau acoplado, $20 \times 10 \times 3$ e CFL=1	86
15	Ranking segregado/acoplado degrau, $50 \times 20 \times 3$ e CFL=1	86
16	Ranking degrau acoplado, $50 \times 20 \times 3$ e CFL=1	86
17	Ranking segregado/acoplado degrau, $10 \times 5 \times 3$ e CFL=5	86
18	Ranking degrau acoplado, $10 \times 5 \times 3$ e CFL=5	86
19	Ranking segregado/acoplado degrau, $20 \times 10 \times 3$ e CFL=5	87
20	Ranking degrau acoplado, $20 \times 10 \times 3$ e CFL=5	87
21	Ranking segregado/acoplado degrau, $50 \times 20 \times 3$ e CFL=5	87
22	Ranking degrau acoplado, $50 \times 20 \times 3$ e CFL=5	87
23	Ranking segregado/acoplado Caledônia, sem refinamento e CFL=1	88
24	Ranking Caledônia, acoplado sem refinamento e CFL=1	88
25	Ranking segregado/acoplado Caledônia, com refinamento e CFL=1	88
26	Ranking Caledônia, acoplado com refinamento e CFL=1	89
27	Ranking segregado/acoplado Caledônia, sem refinamento e CFL=5	89
28	Ranking Caledônia, acoplado sem refinamento e CFL=5	89
29	Ranking segregado/acoplado Caledônia, com refinamento e CFL=5	89
30	Ranking Caledônia, acoplado com refinamento e CFL=5	89
31	Canal, segregado com malha $20 \times 10 \times 3$ e CFL=1	90

32	Canal, segregado com malha $50 \times 20 \times 3$ e CFL=1	90
33	Degrau, segregado com malha $10 \times 5 \times 3$ e CFL=1	90
34	Degrau, segregado com malha $20 \times 10 \times 3$ e CFL=1	91
35	Degrau, segregado com malha $50 \times 20 \times 3$ e CFL=1	91
36	Caledônia, segregado sem refinamento e CFL=1	91
37	Caledônia, segregado com refinamento e CFL=1	91
38	Canal, acoplado com malha $20 \times 10 \times 3$ e CFL=1	92
39	Canal, acoplado com malha $50 \times 20 \times 3$ e CFL=1	93
40	Degrau, acoplado com malha $10 \times 5 \times 3$ e CFL=1	94
41	Degrau, acoplado com malha $20 \times 10 \times 3$ e CFL=1	95
42	Degrau, acoplado com malha $50 \times 20 \times 3$ e CFL=1	96
43	Caledônia, acoplado sem refinamento e CFL=1	97
44	Caledônia, acoplado com refinamento e CFL=1	98
45	Canal, segregado com malha $20 \times 10 \times 3$ e CFL=5	99
46	Canal, segregado com malha $50 \times 20 \times 3$ e CFL=5	99
47	Degrau, segregado com malha $10 \times 5 \times 3$ e CFL=5	99
48	Degrau, segregado com malha $20 \times 10 \times 3$ e CFL=5	100
49	Degrau, segregado com malha $50 \times 20 \times 3$ e CFL=5	100
50	Caledônia, segregado sem refinamento e CFL=5	100
51	Caledônia, segregado com refinamento e CFL=5	100
52	Canal, acoplado com malha $20 \times 10 \times 3$ e CFL=5	101
53	Canal, acoplado com malha $50 \times 20 \times 3$ e CFL=5	102
54	Degrau, acoplado com malha $10 \times 5 \times 3$ e CFL=5	103
55	Degrau, acoplado com malha $20 \times 10 \times 3$ e CFL=5	104
56	Degrau, acoplado com malha $50 \times 20 \times 3$ e CFL=5	105
57	Caledônia, acoplado sem refinamento e CFL=5	106
58	Caledônia, acoplado com refinamento e CFL=5	107

SUMÁRIO

Introdução	6
Organização da Dissertação	6
Notas sobre os Apêndices	7
1 Problema e Conceitos Iniciais	8
1.1 Apresentação do Problema	8
1.2 Solução de Sistemas Lineares	8
1.3 Esparsidade e Métodos Iterativos	10
1.4 Precondicionamento	11
1.4.1 Métodos Diretos como Precondicionadores	12
1.4.2 Ordenamentos	13
1.5 Matrizes de Ponto de Sela	13
2 Dinâmica dos Fluidos	16
2.1 Descrição ALE	16
2.2 Equações de Conservação	18
2.2.1 Equação de Conservação de Massa	19
2.2.2 Equação de Conservação de Quantidade de Movimento Linear .	20
2.3 Método de Elementos Finitos	23
2.3.1 Conceitos Básicos	23
3 Método da Projeção	31
3.1 Método da Projeção Discreto	31
3.1.1 Aproximação Agrupada	33
4 Iterações de Krylov	35

4.1	O Funcionamento dos Métodos Iterativos	35
4.2	Gradiente Conjugado (CG)	38
4.3	GMRes	39
4.4	BiCGStab	40
4.5	Mínimos Quadrados e Equações Normais	41
4.5.1	Matriz de Ponto de Sela e CGLS	42
5	Precondicionamento	45
5.1	Precondicionando $\mathbf{Ax}=\mathbf{b}$	45
5.2	Diagonal	48
5.3	Fatorações Incompletas	48
5.3.1	ILU(0), ILUT, MILU	49
5.3.2	ICC	52
5.4	Ordenamentos	53
5.4.1	Grau Mínimo	55
5.4.2	Cuthill-McKee Reverso	56
5.5	Projeção	57
5.6	Diagonal em Blocos - MGW	58
5.6.0.1	Diagonal	59
5.6.0.2	Lumped	59
5.6.0.3	Matriz completa aproximada (GIRAUD, 2008)	59
5.6.0.4	CGLS	60
5.6.0.5	Probe	60
5.7	Probing-Sondagem	60
5.7.1	Matriz de Ponto de Sela	61
5.8	PCG	62
5.9	GMRes Precondicionado	63
5.9.1	Precondicionamento pela Esquerda	64
5.9.2	Precondicionamento pela Direita	64
5.10	Bi-CGStab Precondicionado	65
6	Resultados Numéricos	67

6.1 Canal 3D	68
6.2 Degrau 3D	68
6.3 Reservatório	70
6.4 Conclusões	73
REFERÊNCIAS	77
APÊNDICE A – Ranking	82
A.1 Canal 3D	83
A.2 Degrau 3D	85
A.3 Reservatório	88
APÊNDICE B – Tabelas de resultados com CFL=1	90
APÊNDICE C – Tabelas de resultados com CFL=5	99

INTRODUÇÃO

Esse trabalho foi desenvolvido no interior do GESAR (Grupo de estudos e simulações ambientais em reservatórios), um grupo interinstitucional sediado na UERJ cujo o maior projeto é o desenvolvimento de um simulador para análise da qualidade da água durante o enchimento de reservatórios de hidrelétricas.

É estimado, que de 80% a 90% do tempo de toda simulação seja despendido na resolução dos sistemas lineares de grande porte existentes. Assim, o objetivo principal é solucionar, de forma eficiente, sistemas de equações lineares provenientes da discretização via elementos finitos das equações de Navier-Stokes, apresentada no Capítulo 2. Apesar da existência de várias técnicas para este fim, nos concentramos na aplicação de preconditionadores.

Nossa contribuição é a comparação de diversos preconditionadores encontrados na literatura para a resolução desse problema. Utilizamos tanto preconditionadores tradicionais quanto outros que têm uma abordagem direcionada ao problema tratado.

Organização da Dissertação

- No Capítulo 1 apresentamos inicialmente o problema tratado, suas principais características e conceitos de algebra linear básica necessários para a compreensão e o desenvolvimento de tópicos apresentados nos outros capítulos.
- No Capítulo 2 são apresentadas equações de conservação que descrevem o comportamento de um fluido newtoniano. Além disso, são também descritos, a forma de discretização espacial das equações através do método de elementos finitos. Depois são apresentadas as equações geradas pela discretização temporal.
- No Capítulo 3 é apresentada uma estratégia para a resolução do problema de ponto de sela que se baseia na decomposição LU em blocos da matriz dos coeficientes. Esta estratégia tem como argumento principal, a capacidade de desacoplar as componentes de velocidade e pressão que facilitam a resolução do sistema tendo em contrapartida a necessidade do uso de alguns critérios de aproximação.
- No Capítulo 4 é apresentada a idéia do funcionamento dos métodos iterativos baseados em projeção no espaço de Krylov bem como a idéia de construção e os algoritmos de alguns destes métodos. Todos os solucionadores de Krylov apresentados foram utilizados para a resolução dos sistemas de equações

gerados.

- No Capítulo 5 é discutida a idéia do preconditionamento e sua forma de implementação nos métodos de Krylov. Além disso são apresentados alguns dos preconditionadores clássicos e reordenamentos, que precisam ser utilizados para alguns deles, bem como algumas formas alternativas de preconditionamentos baseadas na estrutura da matriz de ponto de sela. Todos os preconditionadores apresentados foram utilizados para testar sua eficiência.
- No Capítulo 6 são apresentados os problemas simulados e a abordagem utilizada para resolução dos sistemas lineares. Também está presente uma comparação detalhada do desempenho dos preconditionadores acompanhados dos métodos iterativos que foram utilizados. Além disso é feita uma análise das melhorias no número de condicionamento da matriz de coeficientes bem como a clusterização dos autovalores para cada preconditionador utilizado.

Notas sobre os Apêndices

- No Apêndice A se encontram tabelas que apresentam comparações entre as combinações de solucionadores, preconditionadores e aproximadores que, apresentaram os melhores resultados em termos de velocidade de convergência e acurácia do erro relativo, comparando os métodos acoplados e segregados.
Também se encontram tabelas que apresentam os melhores resultados nas três categoria citadas acima para cada solucionador utilizado, que são, GMRes preconditionado pela esquerda, GMRes preconditionado pela direita e Bi-CGStab.
- No Apêndice B pode-se encontrar as tabelas com todos os testes com CFL=1. Estas tabelas contêm resultados sobre tempo de construção dos preconditionadores, tempo de resolução dos sistemas lineares, erro relativo, e número de iterações.
- No Apêndice C pode-se encontrar as tabelas com todos os testes com CFL=5. Estas tabelas contêm resultados sobre tempo de construção dos preconditionadores, tempo de resolução dos sistemas lineares, erro relativo, e número de iterações.

1 PROBLEMA E CONCEITOS INICIAIS

Neste capítulo introduziremos alguns conceitos de álgebra linear computacional que serão aprofundados no decorrer dos próximos. Faremos uma breve descrição dos problemas tratados, falaremos sobre métodos diretos e iterativos para solução dos sistemas lineares que ocorrem nesses problemas e destacaremos alguns tópicos relevantes: preconditionadores, reordenamentos e espaços de Krylov. Por fim apresentaremos sistemas lineares do tipo ponto de sela.

1.1 Apresentação do Problema

O problema consiste em resolver as equações de Navier-Stokes (detalhamento no Capítulo 2) que modelam o deslocamento de água e outros componentes durante o enchimento de um reservatório. Após a discretização do domínio físico e das equações, o problema passa a ser a resolução de diversos sistemas lineares de grande porte. Sendo assim, esta monografia concentra o esforço e o ferramental para se resolver tais sistemas de forma eficiente. Matricialmente pode-se representar o sistema da seguinte forma

$$\mathcal{A}\mathbf{x} = \mathbf{b} \quad (1.1)$$

de onde deve-se determinar o vetor de incógnitas \mathbf{x} formado por componentes de velocidade e pressão em cada nó. O vetor \mathbf{b} é constante e a matriz dos coeficientes \mathcal{A} é quadrada de ordem n .

Devido à origem física e à forma de discretização, a ordem n da matriz \mathcal{A} é muito grande, o que exige uma análise mais profunda de suas características para poder escolher um método de resolução eficiente. Vamos apresentar duas classes de métodos para a solução de 1.1.

1.2 Solução de Sistemas Lineares

Métodos diretos baseados na fatoração da matriz \mathcal{A} em matrizes associadas a sistemas eficientemente solúveis são amplamente utilizados por serem robustos, ou seja, chegam a solução exata do problema, em tempo finito. O método direto mais popular é baseado em decomposição LU

Teorema 1.1 (GREENBAUM, 1997) (Decomposição LU) *Seja $\mathbf{A} \in \mathbb{R}^{n \times n}$ não singular. Então \mathbf{A} pode ser fatorada da seguinte forma*

$$\mathbf{A} = \mathbf{PLU}$$

onde \mathbf{P} é uma matriz permutação, \mathbf{L} é triangular inferior (todas as posições acima da diagonal principal são nulas), e \mathbf{U} é triangular superior (todas as posições abaixo da diagonal principal são nulas).

A decomposição LU é utilizada com um método direto padrão para resolver um sistema linear $\mathbf{Ax} = \mathbf{b}$. Ao fatorar \mathbf{A} desta maneira, resolve-se $\mathbf{Ly} = \mathbf{P}^T \mathbf{b}$ (pois $\mathbf{P}^T = \mathbf{P}^{-1}$), e então resolve-se $\mathbf{Ux} = \mathbf{y}$. O número de operações necessárias para calcular a decomposição LU , sem considerar a estrutura da matriz, é de $O(n^3)$ ¹ e para se resolver os sistemas para os fatores \mathbf{L} e \mathbf{U} de $O(n^2)$.

Um caso particular dessa decomposição ocorre quando $\mathbf{A} = \mathbf{A}^T$, onde \mathbf{A}^T é a transposta de \mathbf{A} , e ainda $\mathbf{x}^T \mathbf{Ax} > 0$, sempre que $\mathbf{x} \neq 0$, nesse caso dizemos que a matriz é simétrica e definida-positiva. Neste caso, não costuma ser necessário o uso da matriz permutação \mathbf{P} . Além disso, se as matrizes \mathbf{L} e \mathbf{U}^T têm as mesmas entradas da diagonal, então esta decomposição $\mathbf{A} = \mathbf{LL}^T$ é conhecida como *decomposição de Cholesky*. Porém, em outras circunstâncias pode ocorrer que algum elemento da diagonal seja nulo ou muito pequeno, o que implica em uma divisão por zero, ou quase zero, no algoritmo, tornando-o instável, sendo necessário o uso da matriz permutação.

Outro método direto para resolver sistemas lineares é a decomposição QR , que veremos após algumas definições.

Definição 1.1 (HORN; JOHNSON, 1985) O escalar $\mathbf{y}^T \mathbf{x}$ é um produto interno (produto escalar) de \mathbf{x} e $\mathbf{y} \in \mathbb{R}^n$ e é denotado por $\langle \mathbf{x}, \mathbf{y} \rangle \equiv \mathbf{y}^T \mathbf{x}$.

Definição 1.2 Define-se a norma Euclidiana ou norma 2 de um vetor $\mathbf{x} \in \mathbb{R}^n$ representada por $\|\mathbf{x}\|_2$, como $\langle \mathbf{x}, \mathbf{x} \rangle^{\frac{1}{2}}$.

Definição 1.3 (HORN; JOHNSON, 1985) Dois vetores $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ são ditos ortogonais se $\langle \mathbf{y}, \mathbf{x} \rangle = 0$. Estes mesmos vetores são chamados de ortonormais se além de ortogonais, a norma Euclidiana de cada um deles é unitária, isto é, $\langle \mathbf{x}, \mathbf{x} \rangle^{\frac{1}{2}} = \langle \mathbf{y}, \mathbf{y} \rangle^{\frac{1}{2}} = 1$.

Teorema 1.2 (GREENBAUM, 1997) (Decomposição QR) Seja $\mathbf{A} \in \mathbb{R}^{m \times n}$ com $m \geq n$. Existe uma matriz $\mathbf{Q} \in \mathbb{R}^{m \times n}$ com colunas ortonormais e uma matriz $\mathbf{R} \in \mathbb{R}^{n \times n}$ triangular superior tal que $\mathbf{A} = \mathbf{QR}$.

Porém o método de decomposição LU é mais barato que o QR por requerer metade de seu número de operações.

Ao lado dos métodos diretos surgem os métodos iterativos. Quando a resolução de grandes sistemas lineares tornaram-se importantes após a segunda guerra

¹Diremos que uma função $f(x) = O(g(x))$ quando existe uma constante real $c > 0$ tal que $\exists x_0$ tal que $|f(x)| \leq c|g(x)|$ para $x > x_0$.

mundial, os computadores eram muito limitados e os métodos diretos exigiam mais do que eles eram capazes de oferecer. Até vinte anos depois os métodos iterativos estacionários eram os mais utilizados até o surgimento dos métodos iterativos de Krylov (SAAD, 2003; VORST, 2003; SIMONCINI; SZYLD, 2007) (detalhamento no Capítulo 4). A diferença básica mais importante como critério de escolha entre os métodos diretos e iterativos é o número de operações, métodos diretos requerem $O(n^3)$ o que pode significar a impossibilidade física de solução, já para métodos iterativos o número de operações necessárias é $O(n^2)$, que é comparativamente pequeno, dependendo é claro de quão grande n é.

O conceito de magnitude para a ordem n das matrizes é relativo à época em que os problemas tentam ser resolvidos. A tabela abaixo, mostra aproximadamente o que era considerado “muito grande” para uma matriz densa ser utilizada em um computador:

1950:	$n = 20$	(Wilkinson)
1965:	$n = 200$	(Forsythe and Moler)
1980:	$n = 2.000$	(LINPACK)
1995:	$n = 20.000$	(LAPACK)
2008:	$n = 2.456.063$	(LINPACK) (DONGARRA, 2008)

Mesmo que seja fisicamente possível resolver um problema grande, ainda assim é necessário que ele seja rapidamente resolvido. A medida que a capacidade de armazenamento e processamento dos computadores aumenta, também aumentam as variedades de problemas e técnicas de simulação que fazem grandes exigências, não só da capacidade dos computadores mas também do desenvolvimento dos métodos numéricos.

1.3 Esparsidade e Métodos Iterativos

Em muitas aplicações práticas, assim como nos sistemas oriundos da discretização de equações diferenciais parciais (EDPs), as matrizes geradas apresentam elevado grau de esparsidade, isto, é, o número de elementos não nulos é pequeno comparado ao número total de elementos da matriz, em casos importantes chegam a 1% desse total. Esta característica pode ser bem aproveitada computacionalmente, como escolher uma boa forma de representação através de uma estrutura de armazenamento adequada e da utilização de propriedades da arquitetura da máquina para redução do número de operações.

Utilizando-se a fatoração LU de uma forma desavisada, pode ocorrer uma grande quantidade de elementos nos fatores \mathbf{L} e \mathbf{U} que não havia anteriormente na

matriz original, esse fenômeno é conhecido como preenchimento (fill in). Para algumas EDPs o número de elementos não nulos pode passar de $O(n)$ no sistema original para $O(n^{\frac{3}{2}})$ após efetuada a eliminação Gaussiana. Isto implica em um aumento do uso de memória além de um alto custo para obtenção dos fatores \mathbf{L} e \mathbf{U} , assim como da resolução dos sistemas triangulares gerados.

Apesar da estabilidade, por esses motivos, entre outros, muitas vezes, a solução direta não é indicada e a utilização de métodos iterativos baseados na projeção em espaços de Krylov é essencial. Escolher um método que seja eficiente é um problema delicado. Para sistemas lineares cuja matriz dos coeficientes é simétrica e positiva definida (SPD) a escolha é fácil, escolhe-se o Gradiente Conjugado (CG) (HESTENES; STIEFEL, 1952). Para outras matrizes a situação é mais difusa, para matrizes não simétricas, por exemplo, o GMRes (SAAD; SCHULTZ, 1986) é mais robusto porém é caro em termos de trabalho por iteração; já o Bi-CGStab (VORST, 1992) por outro lado é mais barato, no entanto pode apresentar problemas de convergência e estabilidade.

Nem sempre uma boa escolha do método é suficiente, para garantir uma convergência eficiente, várias estratégias devem ser levadas em consideração, como a construção de preconditionadores, reordenamentos, aumento da eficiência dos núcleos de computação intensiva, paralelização do código, dentre outras.

Apesar de ser não simétrica, a matriz \mathcal{A} do nosso problema apresenta uma estrutura simétrica, isto é, $a_{ij} \neq 0$ se e somente se $a_{ji} \neq 0$, ou *quase simétrica*, onde é vantajoso considerar o armazenamento de alguns elementos nulos para considerá-la simétrica por estrutura.

1.4 Precondicionamento

As técnicas de preconditionamento são não só essenciais para a melhoria da performance assim como para a viabilidade dos métodos de Krylov. É amplamente reconhecido que o preconditionamento é uma das partes críticas no desenvolvimento de formas eficientes de resolução de sistemas na computação científica. Nos últimos anos, muito esforço tem sido dedicado pela comunidade científica para o desenvolvimento de preconditionadores eficientes. Já que não é possível construir um preconditionador que resolva um problema de forma genérica, esta área de pesquisa está aberta à novos desenvolvimentos já que a cada momento podem surgir aplicações que necessitem de um preconditionamento mais eficiente para o tipo de problema gerado.

Qualquer transformação feita em um sistema pode ser chamada de preconditionamento no entanto, é comum só se referir ao termo preconditionamento quando o sistema é transformado em outro que tenha propriedades mais favoráveis que o original para uma solução por métodos iterativos. De forma geral, o preconditionador,

que é o meio de preconditionamento, deve melhorar, entre outras coisas, as propriedades espectrais da matriz dos coeficientes. Para o CG, a taxa de convergência depende da distribuição dos autovalores de \mathbf{A} . Assim espera-se que a matriz transformada (precondicionada) tenha o número de condicionamento pequeno ou autovalores aglomerados em torno de 1 (BENZI, 2002). Para problemas não simétricos a situação pode ser um pouco diferente (BENZI, 2002; GREENBAUM, 1997).

O critério de escolha de construção de um preconditionador baseada em seu custo depende, entre outros fatores, se ele será reutilizado ou não. É comum se deparar com problemas de onde surge uma sequência de sistemas em que a matriz dos coeficientes permanece constante (ou quase) enquanto o vetor do lado direito muda. Assim é vantajoso gastar algum tempo para construir um preconditionador de qualidade pois o tempo de construção será amortizado ao longo da sequência de sistemas.

1.4.1 Métodos Diretos como Precondicionadores

Normalmente, a aplicação do preconditionador é feita de forma implícita, quer dizer, ao aplicar o método iterativo, a aplicação do preconditionador não irá alterar de fato o sistema, apenas implica na resolução de um sistema linear $\mathbf{Mz} = \mathbf{r}$ para o resíduo \mathbf{r} onde \mathbf{M} é a matriz conhecida como preconditionador.

Devido ao alto custo dos métodos diretos eles podem ser substituídos por métodos iterativos, no entanto para a resolução do sistema oriundo do preconditionamento, pode ser interessante usar a reconhecida eficiência dos métodos diretos. Como o preconditionador pode ser uma aproximação do problema original, a solução do sistema baseado no preconditionador não precisa ser de alta qualidade para promover uma melhoria na performance das iterações. Costuma-se obter a matriz \mathbf{M} como produto implícito dos fatores aproximados da matriz original \mathbf{A} .

Apesar disto, apenas esta técnica pode não garantir a qualidade necessária para a construção de um preconditionador. A esparsidade da matriz ainda é uma peça fundamental para os métodos iterativos mas, através das fatorações, os fatores podem tornar-se densos. Ao tentar descartar alguns elementos, para garantir esparsidade, no processo de aproximação dos fatores, pode se perde a qualidade.

É possível, porém, a utilização de técnicas que, através de simples trocas de colunas e linhas, garantam às fatorações um menor grau de preenchimento ao mesmo tempo garantir uma melhoria na aproximação, bem como em velocidade e armazenamento.

1.4.2 Ordenamentos

Desde os últimos anos da década de 60, surgiu um intenso interesse neste ramo que tem como objetivo melhorar o desempenho do processo de eliminação gaussiana (BENZI, 2002). Ao se trocar linhas e colunas de um sistema, é necessária a utilização de algum critério para que ao realizar-se a eliminação de Gauss seus fatores se tornem tão pouco densos quanto possível.

Se \mathbf{P} é uma matriz de permutação, o número de enchimentos no processo de eliminação gaussiana das matrizes de estrutura simétrica \mathbf{A} e \mathbf{PAP}^T pode ser drasticamente diferente. Coloca-se então o problema de encontrar uma matriz de permutação \mathbf{P}^* tal que a matriz $\mathbf{P}^* \mathbf{A} \mathbf{P}^{*T}$ sofra o menor número de enchimentos possível em seus fatores \mathbf{L} e \mathbf{U} . Tal problema de minimização de enchimentos no processo de eliminação gaussiana pertence a uma classe de problemas extremamente complexa, os chamados problemas NP-completos, para o qual não se conhece nenhum algoritmo eficiente de resolução (YANNAKAKIS, 1981; BILLIONNET, 1986). No entanto, vários algoritmos heurísticos têm sido propostos para a determinação de um ordenamento de equações e variáveis de modo a reduzir os enchimentos no processo de eliminação.

É conveniente que a estrutura simétrica seja preservada, tanto para garantir a utilização de técnicas de armazenamento quanto para manter matriz diagonalmente dominante, pois matrizes oriundas de EDPs são, em casos importantes, diagonal dominantes. Uma matriz é dita diagonal dominante por linhas (colunas) se o valor absoluto de cada entrada da diagonal é maior que a soma do valor absoluto dos elementos fora da diagonal de suas linhas (colunas) (caso essa soma seja maior ou igual, alguns autores usam o termo fracamente diagonal dominante, ver (GREENBAUM, 1997)). No caso de diagonal dominância a matriz é não singular devido ao teorema de Gershgorin em (GREENBAUM, 1997).

Ao aplicar permutações em sua linhas e colunas ela pode perder a diagonal dominância, e, se for o caso preservá-la, toda permutação feita em suas colunas deverá também ser feita em suas linhas, mantendo sua estrutura simétrica e os elementos da diagonal original na diagonal da matriz permutada.

1.5 Matrizes de Ponto de Sela

Grandes sistemas lineares do tipo ponto de sela surgem em uma ampla variedade de aplicações através da ciência da computação e engenharia. Devido a propriedades espectrais pobres das matrizes associadas esses sistemas representam um grande desafio. Em (BENZI; GOLUB; LIESEN, 2005) é feita a apresentação de uma variedade de aplicações e formas de resolução deste tipo de sistema.

O problema de ponto de sela se caracteriza por sua matriz de coeficientes em blocos 2×2 apresentada abaixo

$$\begin{bmatrix} \mathbf{A} & \mathbf{G} \\ \mathbf{D} & -\mathbf{C} \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix} \quad \text{ou } \mathcal{A}\mathbf{x} = \mathbf{b}, \quad (1.2)$$

$$\mathbf{A} \in \mathbb{R}^{n \times n}, \quad \mathbf{G}^T, \mathbf{D} \in \mathbb{R}^{m \times n}, \quad \mathbf{C} \in \mathbb{R}^{m \times m} \quad \text{com } m \geq n. \quad (1.3)$$

Em nosso caso, valem as seguintes condições

1. \mathbf{A} é simétrica: $\mathbf{A} = \mathbf{A}^T$;
2. \mathbf{A} é positiva definida: $\mathbf{x}^T \mathbf{A} \mathbf{x} > 0 \quad \forall \mathbf{x} \neq 0$;
3. $\mathbf{G}^T \neq \mathbf{D}$;
4. \mathbf{G}^T e \mathbf{D} têm a mesma estrutura de elementos não nulos, ou quase a mesma;
5. $\mathbf{C} = \mathbf{0}$ (\mathbf{C} é uma matriz identicamente nula);
6. \mathbf{A} , \mathbf{G}^T e \mathbf{D} têm posto completo.

Assim temos o seguinte sistema de ponto de sela não simétrico

$$\begin{bmatrix} \mathbf{A} & \mathbf{G} \\ \mathbf{D} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix} \quad (1.4)$$

que pode ser observado no exemplo na Figura 1, onde a matriz \mathcal{A} tem ordem de cerca de 8500 e o percentual de elementos não nulos é de aproximadamente 0,25%.

Repare que a característica número 6. não garante a invertibilidade de \mathcal{A} , apesar disso, ela será invertível caso o complemento de Schur $\mathbf{S} = -\mathbf{D}\mathbf{A}^{-1}\mathbf{G}$ o seja.

Ao fatorar a matriz \mathcal{A} em LU em blocos da seguinte forma

$$\begin{bmatrix} \mathbf{A} & \mathbf{G} \\ \mathbf{D} & \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{D} & -\mathbf{D}\mathbf{A}^{-1}\mathbf{G} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{A}^{-1}\mathbf{G} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \quad (1.5)$$

surge o termo $\mathbf{S} = -\mathbf{D}\mathbf{A}^{-1}\mathbf{G}$ que aparece no fator \mathbf{L} é conhecido como complemento de Schur de \mathcal{A} em \mathbf{A} .

A inversa de \mathcal{A} pode ser escrita explicitamente como segue

$$\mathcal{A}^{-1} = \begin{bmatrix} \mathbf{A} & \mathbf{G} \\ \mathbf{D} & \mathbf{0} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{G}\mathbf{S}^{-1}\mathbf{D}\mathbf{A}^{-1} & -\mathbf{A}^{-1}\mathbf{G}\mathbf{S}^{-1} \\ -\mathbf{S}^{-1}\mathbf{D}\mathbf{A}^{-1} & \mathbf{S}^{-1} \end{bmatrix}, \quad (1.6)$$

que exige que \mathbf{S}^{-1} exista para que \mathcal{A}^{-1} também exista.

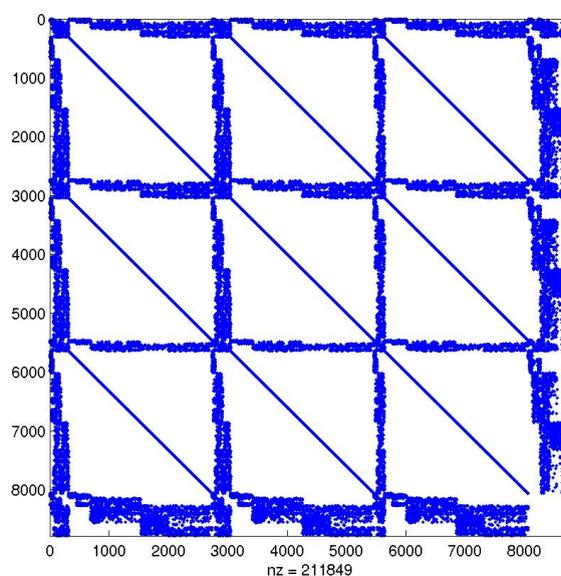


Figura 1: Exemplo de estrutura de uma matriz de ponto de sela

Devido ao tipo de estrutura e características destas matrizes de ponto de sela, objetivamos a utilização tanto dos métodos tradicionais de resolução como de métodos que sejam capazes de obter alguma melhoria a partir do aproveitamento desta estrutura de ponto de sela.

2 DINÂMICA DOS FLUIDOS

O objetivo deste capítulo é apresentar o processo de discretização das equações de *Navier-Stokes* que, descritas dentro da formulação Lagrangeana-Euleriana Arbitrária, descrevem o escoamento de um fluido. Como consequência desta formulação obtemos os sistemas lineares descritos no capítulo Capítulo 1, que surgem também em diversos outros contextos.

Na seção 2.1, encontra-se um resumo da descrição Lagrangeana-Euleriana Arbitrária; na seção 2.2, encontram-se as equações de conservação de massa e quantidade de movimento linear e na seção 2.3, podem ser vistos conceitos básicos de elementos finitos e a discretização das equações de *Navier-Stokes*.

2.1 Descrição ALE

Para se especificar o movimento de um fluido em uma dada região do espaço, existem três principais tipos de abordagens: a descrição *Euleriana*, a descrição *Lagrangeana* e a descrição *Lagrangeana-Euleriana Arbitrária*.

Na formulação Euleriana, define-se uma região fixa no espaço, que não deforma com relação ao tempo, onde o comportamento do fluido será estudado. Neste caso, há um fluxo de massa pelas faces desse volume de controle, e as equações para o escoamento são determinadas a partir do balanço de fluxo neste volume de controle.

Na formulação Lagrangeana, define-se uma região material formada por um conjunto de partículas de fluido, denominado de volume de controle Lagrangeano, ou ainda região material. Conforme as partículas se movimentam no escoamento, esta região se deforma, e não há fluxo de massa através de suas fronteiras. Nesta formulação, as grandezas do escoamento são especificadas como função do tempo e da partícula de fluido. A terceira abordagem, a formulação Lagrangeana-Euleriana Arbitrária (ou ALE, isto é, Arbitrary Lagrangian-Eulerian), é uma forma mais geral de se descrever o movimento de um fluido. Nesta, as equações são desenvolvidas em um referencial se movendo no espaço com velocidade arbitrária. Quando este referencial se move com a mesma velocidade do fluido, obtém-se a descrição Lagrangeana, e quando o referencial está parado, obtém-se a descrição Euleriana. Segundo (SIMEONI, 2005), essa descrição é muito útil para o caso de métodos com malhas dinâmicas, onde as equações são escritas no referencial da malha que se move com velocidade arbitrária. Assim, baseando-se em (SIMEONI, 2005), será descrita a variação temporal das propriedades macroscópicas do fluido em um referencial se movendo com velocidade arbitrária, a fim de se determinar, posteriormente, as equações

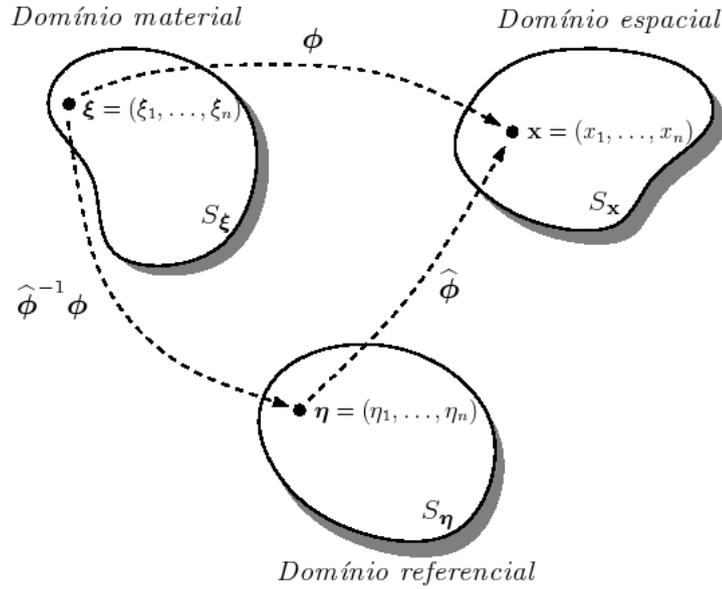


Figura 2.1: Domínios utilizados na formulação ALE.

de conservação utilizando-se a descrição Lagrangeana-Euleriana arbitrária. Sendo assim, considerando-se os domínios material (S_ξ), referencial ou computacional (S_η) e espacial (S_x), mostrados na Figura 2.1, na descrição Lagrangeana-Euleriana arbitrária, os domínios S_ξ e S_η estão se movendo enquanto S_x está parado. Considerando-se que uma partícula ξ de fluido que ocupa o domínio S_ξ se move para um ponto x em um dado instante t no domínio espacial, o movimento de um fluido pode ser descrito por uma transformação ϕ tal que $x = \phi(\xi, t)$. Assim, define-se *velocidade* da partícula ξ no domínio espacial como

$$\mathbf{u} = \left. \frac{\partial \mathbf{x}}{\partial t} \right|_{\xi} = \frac{\partial}{\partial t} \phi(\xi, t).$$

Da mesma forma, seja $\hat{\phi}$ a transformação que descreve o movimento de um ponto η no domínio S_η para um ponto x no domínio S_x , então define-se velocidade do ponto η no domínio espacial como

$$\hat{\mathbf{u}} = \left. \frac{\partial \mathbf{x}}{\partial t} \right|_{\eta} = \frac{\partial}{\partial t} \hat{\phi}(\eta, t).$$

Seja f uma propriedade expressa como função de (ξ, t) ou (x, t) , ou seja, expressa através de uma descrição material (Lagrangeana) ou espacial (Euleriana). Em uma formulação Lagrangeana-Euleriana arbitrária, a expressão de f como função de (η, t) é chamada de descrição referencial. Considerando-se as seguintes derivadas

temporais

$$\left. \frac{\partial}{\partial t} f(\mathbf{x}, t) \right|_{\xi} = \left. \frac{\partial}{\partial t} f(\phi(\xi, t), t) \right|_{\xi} = \left. \frac{\partial f}{\partial t} \right|_{\mathbf{x}} + \frac{\partial}{\partial t} \phi(\xi, t) \cdot \nabla f = \left. \frac{\partial f}{\partial t} \right|_{\mathbf{x}} + (\mathbf{u} \cdot \nabla) f, \quad (2.1)$$

$$\left. \frac{\partial}{\partial t} f(\mathbf{x}, t) \right|_{\eta} = \left. \frac{\partial}{\partial t} f(\widehat{\phi}(\eta, t), t) \right|_{\eta} = \left. \frac{\partial f}{\partial t} \right|_{\mathbf{x}} + \frac{\partial}{\partial t} \widehat{\phi}(\eta, t) \cdot \nabla f = \left. \frac{\partial f}{\partial t} \right|_{\mathbf{x}} + (\widehat{\mathbf{u}} \cdot \nabla) f, \quad (2.2)$$

subtraindo-se (2.2) de (2.1), tem-se

$$\frac{Df}{Dt} = \left. \frac{\partial f}{\partial t} \right|_{\xi} = \left. \frac{\partial f}{\partial t} \right|_{\eta} + ((\mathbf{u} - \widehat{\mathbf{u}}) \cdot \nabla) f$$

que é a *derivada material* ou *substantiva* da propriedade f em um referencial η que se move com velocidade $\widehat{\mathbf{u}}$.

Note que, em uma descrição Lagrangeana, o domínio referencial se desloca com a mesma velocidade do domínio material, ou seja, $\mathbf{u} = \widehat{\mathbf{u}}$, e portanto a variação temporal da propriedade f nos domínios material e referencial é a mesma.

Já em uma descrição Euleriana, o domínio referencial está parado em relação ao domínio espacial, ou seja, $\widehat{\mathbf{u}} = 0$, e portanto a expressão da variação temporal de f no domínio material é dada por

$$\frac{Df}{Dt} = \left. \frac{\partial f}{\partial t} \right|_{\xi} = \left. \frac{\partial f}{\partial t} \right|_{\eta} + (\mathbf{u} \cdot \nabla) f$$

que corresponde à derivada material de f utilizando-se uma descrição Euleriana.

2.2 Equações de Conservação

A partir da seção 2.1, pode-se alcançar as equações de conservação (de *massa* e *quantidade de movimento linear*), utilizando-se a formulação Lagrangeana-Euleriana arbitrária. Porém, é necessário, antes de tudo, enunciar dois teoremas importantes para o desenvolvimento desta seção.

Teorema 2.1 (Divergência de Gauss) *Seja $\Omega \subset \mathbb{R}^m$, $\Gamma = \partial\Omega$ o bordo de Ω e $\mathbf{u} : U \subset \mathbb{R}^m \rightarrow \mathbb{R}^m$ um campo vetorial. Então*

$$\int_{\Omega} \nabla \cdot \mathbf{u} \, d\Omega = \int_{\Gamma} \mathbf{u} \cdot \mathbf{n} \, d\Gamma$$

onde \mathbf{n} é o vetor normal à Γ .

Teorema 2.2 (Teorema de Leibnitz) *Sejam $R(t) \subset \mathbb{R}^m$ uma região movendo-se no espaço com velocidade \mathbf{w} , $S(t) = \partial R(t)$ o bordo de $R(t)$ e T uma grandeza de*

dimensão $k < \infty$. Então

$$\frac{d}{dt} \int_{R(t)} \mathbf{T}(\mathbf{x}, t) dV = \int_R \frac{\partial}{\partial t} \mathbf{T}(\mathbf{x}, t) \Big|_{\mathbf{x}} dV + \int_S (\mathbf{n} \cdot \mathbf{w}) \mathbf{T} dS.$$

2.2.1 Equação de Conservação de Massa

Sabendo-se que a massa total numa região material no espaço $V(t) \subset \mathbb{R}^m$ é dada por

$$m(V(t), t) = \int_{V(t)} \rho(\mathbf{x}, t) dV,$$

onde $\rho(\mathbf{x}, t)$ é a massa por unidade de volume (*massa específica*).

Sabendo-se que, pelo princípio da conservação de massa, na ausência de fontes ou sorvedouros, a variação temporal de massa na região $V(t)$ é nula, ou seja,

$$\frac{d}{dt} m(V(t), t) = \frac{d}{dt} \int_{V(t)} \rho(\mathbf{x}, t) dV = 0. \quad (2.3)$$

Aplicando-se o Teorema 2.2 em (2.3), encontra-se

$$\int_V \frac{\partial}{\partial t} \rho(\mathbf{x}, t) \Big|_{\mathbf{x}} dV + \int_S (\mathbf{n} \cdot \mathbf{u}) \rho dS = 0,$$

onde $S(t) = \partial V(t)$ é o bordo e \mathbf{u} a velocidade da região material $V(t)$, e \mathbf{n} é um vetor unitário normal a $S(t)$. Aplicando-se o Teorema 2.1 na equação anterior, tem-se

$$\int_V \left(\frac{\partial}{\partial t} \rho(\mathbf{x}, t) \Big|_{\mathbf{x}} + \nabla \cdot (\rho \mathbf{u}) \right) dV = 0,$$

e, uma vez que essa equação deve ser verificada para qualquer região fixa de V , é necessário que a expressão entre parênteses seja identicamente nula (GAMA, 2005). Sendo assim,

$$\frac{\partial}{\partial t} \rho(\mathbf{x}, t) \Big|_{\mathbf{x}} + \nabla \cdot (\rho \mathbf{u}) = 0$$

ou,

$$\underbrace{\frac{\partial \rho}{\partial t} \Big|_{\eta} + ((\mathbf{u} - \hat{\mathbf{u}}) \cdot \nabla) \rho}_{\text{Derivada Material, seção 2.1, com } f = \rho} + \rho \nabla \cdot \mathbf{u} = 0,$$

Derivada Material, seção 2.1, com $f = \rho$.

substituindo-se por (2.2) e utilizando-se a identidade

$$\nabla \cdot (\rho \mathbf{u}) = \rho \nabla \cdot \mathbf{u} + (\mathbf{u} \cdot \nabla) \rho,$$

ou, ainda,

$$\frac{D\rho}{Dt} + \rho \nabla \cdot \mathbf{u} = 0, \quad (2.4)$$

utilizando-se a notação de derivada material expressa na seção 2.1. No caso de uma simulação de escoamentos multifásicos, cada fluido envolvido determina uma fase. Assim, foi considerado, no trabalho desenvolvido em (SIMEONI, 2005), o fluido em todas as fases como um contínuo, onde as propriedades mudam de acordo com a posição de partículas marcadoras de interface em cada ponto do escoamento como um todo. Desta forma, as equações de conservação de massa e de quantidade de movimento foram desenvolvidas com propriedades, como massa específica (ρ) e viscosidade (μ), variáveis para o escoamento como um todo, mas constantes no interior de cada uma das fases separadamente. Devido ao fato de este trabalho se basear nesta referência, considera-se a mesma abordagem com relação as fases do fluido.

Sendo assim, seja $\Omega = \cup_i \Omega_i$, onde Ω_i é a região ocupada pelo fluido i , tem-se

$$\begin{aligned} \rho(\mathbf{x}) &= \rho_i, \text{ se } \mathbf{x} \in \Omega_i, \\ \mu(\mathbf{x}) &= \mu_i, \text{ se } \mathbf{x} \in \Omega_i. \end{aligned}$$

Desta forma, tem-se que ρ e μ variam de acordo com a posição, mas são constantes no interior de cada subdomínio Ω_i e não se alteram em relação ao tempo, o que permite escrever que

$$\frac{D\rho}{Dt} = 0,$$

e, substituindo em (2.4), tem-se

$$\rho \nabla \cdot \mathbf{u} = 0.$$

Como $\rho \neq 0$ para todas as fases, pode-se afirmar que

$$\nabla \cdot \mathbf{u} = 0. \quad (2.5)$$

A Equação (2.5) é conhecida como condição de incompressibilidade, e será mantida em todo o domínio. Os fluidos que obedecem esta condição são chamados *incompressíveis*.

2.2.2 Equação de Conservação de Quantidade de Movimento Linear

A equação de conservação da quantidade de movimento linear é obtida aplicando-se a segunda lei de Newton, a qual afirma que a taxa de variação temporal da quantidade de movimento de uma partícula é igual a resultante das forças que agem sobre essa partícula. Considere novamente uma região material arbitrária $V(t) \in \Omega$

movendo-se no espaço, limitada por uma superfície $S(t) = \partial V(t)$.

O balanço integral da quantidade de movimento pode ser expresso pela equação

$$\frac{d}{dt} \int_{V(t)} \rho \mathbf{u}(\mathbf{x}, t) dV = \int_S \boldsymbol{\sigma} \cdot \mathbf{n} dS + \int_{V(t)} \rho \mathbf{b}(\mathbf{x}, t) dV$$

Na equação acima, o lado esquerdo representa a taxa de variação de quantidade de movimento linear (produto da massa pela aceleração), a primeira integral do lado direito representa a resultante das tensões agindo sobre a superfície S , e a segunda integral do lado direito representa a resultante das forças de campo.

Aplicando-se o Teorema 2.2 na equação acima tem-se

$$\int_V \frac{\partial(\rho \mathbf{u})}{\partial t} \Big|_{\mathbf{x}} dV + \int_S (\mathbf{n} \cdot \mathbf{u}) \rho \mathbf{u} dS = \int_S \boldsymbol{\sigma} \cdot \mathbf{n} dS + \int_V \rho \mathbf{b} dV$$

e utilizando o Teorema 2.1 para transformar as integrais de superfície em integrais de volume, tem-se

$$\int_V \frac{\partial(\rho \mathbf{u})}{\partial t} \Big|_{\mathbf{x}} dV + \int_V \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) dV = \int_V \nabla \cdot \boldsymbol{\sigma} dV + \int_V \rho \mathbf{b} dV,$$

onde \otimes representa o produto diádico entre dois vetores dado por

$$\begin{aligned} \otimes : \mathbb{R}^m \times \mathbb{R}^m &\rightarrow \mathbb{R}^{m^2} \\ (\mathbf{u}, \mathbf{v}) &\mapsto \mathbf{u} \otimes \mathbf{v} = \mathbf{A} = (u_i v_j) \end{aligned}$$

Assim

$$\int_V \left(\frac{\partial(\rho \mathbf{u})}{\partial t} \Big|_{\mathbf{x}} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) - \nabla \cdot \boldsymbol{\sigma} - \rho \mathbf{b} \right) dV = 0,$$

de onde vem

$$\frac{\partial(\rho \mathbf{u})}{\partial t} \Big|_{\mathbf{x}} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) - \nabla \cdot \boldsymbol{\sigma} - \rho \mathbf{b} = 0.$$

Utilizando-se a Equação (2.2) para substituir a derivada temporal, e distribuindo o operador divergente do segundo termo, obtém-se

$$\frac{\partial(\rho \mathbf{u})}{\partial t} \Big|_{\boldsymbol{\eta}} - (\hat{\mathbf{u}} \cdot \nabla)(\rho \mathbf{u}) + (\mathbf{u} \cdot \nabla)(\rho \mathbf{u}) + \rho \mathbf{u}(\nabla \cdot \mathbf{u}) - \nabla \cdot \boldsymbol{\sigma} - \rho \mathbf{b} = 0,$$

que, pela Equação (2.5), resulta em

$$\frac{\partial(\rho \mathbf{u})}{\partial t} \Big|_{\boldsymbol{\eta}} + ((\mathbf{u} - \hat{\mathbf{u}}) \cdot \nabla) \rho \mathbf{u} - \nabla \cdot \boldsymbol{\sigma} - \rho \mathbf{b} = 0,$$

e finalmente, com a notação de derivada material (seção 2.1) tem-se

$$\frac{D(\rho \mathbf{u})}{Dt} = \nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{b}. \quad (2.6)$$

Na equação (2.6), $\boldsymbol{\sigma}$ representa o tensor de tensões totais do escoamento. Utilizando um modelo *Newtoniano* isotrópico, este tensor pode ser escrito como

$$\boldsymbol{\sigma} = -p \mathbf{I} + \boldsymbol{\tau}$$

onde $p = p(\mathbf{x}, t)$ representa o campo de pressão do escoamento, e $\boldsymbol{\tau}$ é o tensor de tensões viscosas dado por

$$\boldsymbol{\tau} = 2\mu \mathbf{D} + \left(\lambda - \frac{2}{3}\mu \right) (\nabla \cdot \mathbf{u}) \mathbf{I}, \quad (2.7)$$

sendo μ o coeficiente de viscosidade dinâmica e λ o coeficiente de viscosidade volumétrica do fluido. Devido a condição de incompressibilidade (2.5), pode-se escrever simplesmente

$$\boldsymbol{\tau} = 2\mu \mathbf{D}. \quad (2.8)$$

O tensor \mathbf{D} que aparece nas equações (2.7) e (2.8), é denominado *tensor deformação*, e é dado por

$$\mathbf{D} = \frac{1}{2} \dot{\boldsymbol{\gamma}}(\mathbf{u}) = \frac{1}{2} (\nabla \mathbf{u} + \nabla \mathbf{u}^T),$$

onde $\dot{\boldsymbol{\gamma}}(\mathbf{u})$ é denominado de *tensor taxa de deformação*.

Deste modo, utilizando o tensor de tensões *Newtoniano* isotrópico, pode-se calcular o divergente do tensor de tensões totais:

$$\nabla \cdot \boldsymbol{\sigma} = \nabla \cdot [-p \mathbf{I} + \mu (\nabla \mathbf{u} + \nabla \mathbf{u}^T)] = -\nabla p + \nabla \cdot [\mu (\nabla \mathbf{u} + \nabla \mathbf{u}^T)].$$

O vetor \mathbf{b} é uma força por unidade de massa que age sobre o fluido na região Ω . Para o tipo de escoamento que é objeto de interesse neste trabalho, somente o campo gravitacional é considerado, ou seja, $\mathbf{b} = \mathbf{g}$. Finalmente, a equação de conservação de quantidade de movimento pode ser expressa como

$$\frac{D(\rho \mathbf{u})}{Dt} = -\nabla p + \nabla \cdot [\mu (\nabla \mathbf{u} + \nabla \mathbf{u}^T)] + \rho \mathbf{g}. \quad (2.9)$$

As equações da continuidade (2.5) e da conservação de quantidade de movimento linear (2.9) são as equações de conservação que modelam escoamentos incompres-

síveis, e são também chamadas de equações de *Navier-Stokes*, que, na forma adimensional (veja (SIMEONI, 2005), para maiores detalhes), podem ser escrita como

$$\frac{\partial(\rho\mathbf{u})}{\partial t} + ((\mathbf{u} - \hat{\mathbf{u}}) \cdot \nabla)\rho\mathbf{u} = -\nabla p + \frac{1}{N^{\frac{1}{2}}}\nabla \cdot [\mu(\nabla\mathbf{u} + \nabla\mathbf{u}^T)] + \rho\mathbf{g} + \frac{1}{E_0}\mathbf{f} \quad (2.10)$$

$$\nabla \cdot \mathbf{u} = 0, \quad (2.11)$$

onde

$$N = \frac{\rho_0^2 D^3 g}{\mu_0^2} \quad \text{e} \quad E_0 = \frac{\rho_0 g D^2}{\sigma_0},$$

são o *Número de Galileu* e o *Número de Eötvös* respectivamente, e onde ρ_0 , μ_0 e σ_0 são, respectivamente, valores de referência para massa específica, viscosidade e coeficiente de tensão superficial, D é o diâmetro (aparece em escoamento com bolhas) e g é a força gravitacional. A força \mathbf{f} definida por

$$\mathbf{f} = -\sigma\kappa\nabla H,$$

onde $\sigma = \sigma(\mathbf{x}, t) \in \mathbb{R}$ é o coeficiente de tensão interfacial entre os fluidos, $\kappa = \kappa(\mathbf{x}, t) \in \mathbb{R}$ denota a curvatura local da superfície que define a interface e $H : \mathbb{R}^m \rightarrow \{0, 1\}$ é uma função de *Heaviside* que pode ser definida como 1 onde existe um determinado fluido ou 0 caso contrário, representa, no caso de escoamentos multifásicos, uma força campo que está ligada à tensão interfacial, e portanto, deve agir diretamente na interface entre dois fluidos

2.3 Método de Elementos Finitos

Nesta seção, será feita uma breve descrição do *Método de Elementos Finitos*, a fim de se gerar uma base teórica para a obtenção dos sistemas de equações aqui trabalhados. Para maiores detalhes sobre o assunto, veja (SIMEONI, 2005; ANJOS, 2007; ZIENKIEWICZ; TAYLOR, 2000a; ZIENKIEWICZ; TAYLOR, 2000b).

2.3.1 Conceitos Básicos

O método de elementos finitos é uma eficiente ferramenta numérica de resolução de problemas de meio contínuo que consiste na divisão deste meio, isto é, deste domínio Ω , em subdomínios Ω_i conhecidos como elementos que, por sua vez, podem ser obtidos através de técnicas de geração de malha como, por exemplo, a triangulação de Delaunay para malhas 2D ou, a tetraedralização Delaunay para malhas 3D.

Para se ter uma idéia básica de como é este método, considere uma função

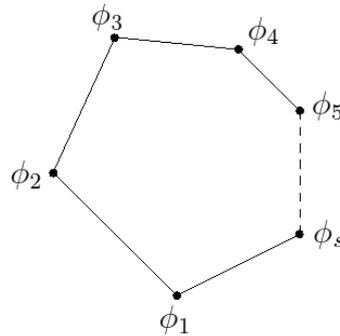


Figura 2.2: Elemento arbitrário com s lados.

$\phi : \mathbb{R}^m \longrightarrow \mathbb{R}$, válida no domínio Ω com condições de contorno estabelecidas como sendo a solução exata do problema

$$\mathcal{L}\phi = f, \quad (2.12)$$

onde \mathcal{L} é um operador diferencial. O método de elementos finitos busca uma aproximação $\Phi(\mathbf{x})$ da solução exata $\phi(\mathbf{x})$ de maneira particionada em cada elemento Ω^e do domínio, ou seja, a aproximação de $\phi(\mathbf{x})$ restrita a cada elemento Ω^e é tal que $\phi^e(\mathbf{x}) = 0$ para qualquer ponto \mathbf{x} fora de Ω^e . Assim, a aproximação de $\phi(\mathbf{x})$ pode ser escrita como

$$\Phi(\mathbf{x}) = \sum_e \phi^e(\mathbf{x}).$$

O valor de ϕ avaliado nos vértices da malha que discretiza Ω é chamado de valor nodal de ϕ , denotado por ϕ_i , onde i é o número do nó correspondente, isto é,

$$\phi_i = \phi(\mathbf{x}_i).$$

Define-se graus de liberdade de um vértice (ou nó) como sendo a quantidade de variáveis associadas com este mesmo nó, e graus de liberdade de um elemento como sendo a somatória dos graus de liberdade dos nós que definem tal elemento. Considere um elemento poligonal com s lados, como mostrado na Figura 2.2, e considere que, associado a cada vértice do elemento está o valor de ϕ avaliado naquele nó, digamos ϕ_i , $i = 1, \dots, s$. A cada elemento pode ser associada uma família de funções de interpolação, também chamadas de funções de forma, representada ge-

nericamente pela matriz

$$\mathbf{N}^e(\mathbf{x}) = \begin{bmatrix} N_{1,1}^e(\mathbf{x}) & \cdots & N_{1,s}^e(\mathbf{x}) \\ \vdots & \ddots & \vdots \\ N_{r,1}^e(\mathbf{x}) & \cdots & N_{r,s}^e(\mathbf{x}) \end{bmatrix}$$

onde r representa o grau de liberdade de cada nó, e s representa a quantidade de nós em cada elemento. Portanto, o número de graus de liberdade no elemento é dado por rs . Assumindo que cada nó tenha somente um grau de liberdade, tem-se

$$\mathbf{N}^e(\mathbf{x}) = [N_1^e(\mathbf{x}), \dots, N_s^e(\mathbf{x})] .$$

Assim, sendo

$$\boldsymbol{\delta}^e = [\phi_1, \dots, \phi_s]^T ,$$

o que é chamado de vetor deslocamento na literatura de elementos finitos, tem-se que a aproximação para ϕ em cada elemento é dada por

$$\phi^e(\mathbf{x}) = \mathbf{N}^e \boldsymbol{\delta}^e \quad (2.13)$$

e a solução aproximada de (2.12) pode ser escrita como

$$\Phi(\mathbf{x}) = \sum_{e=1}^{NE} \phi^e(\mathbf{x}) = \sum_{e=1}^{NE} \mathbf{N}^e \boldsymbol{\delta}^e , \quad (2.14)$$

onde NE é o número total de elementos. Tendo sido decidido o tipo de discretização do domínio, o próximo passo é escolher como são as funções de interpolação. A escolha mais comum é optar por funções polinomiais, pois além de serem mais fáceis de manipular, tanto algebricamente quanto computacionalmente, aproximações polinomiais possuem propriedades garantidas pelo teorema da aproximação de *Weierstrass*, o qual especifica que qualquer função contínua pode ser aproximada por um polinômio, tão próximo quanto se deseje.

A escolha de tais aproximações polinomiais é feita seguindo-se algumas observações:

1. O número total de termos no polinômio deve ser igual ao número total de graus de liberdade do elemento. Isto garante a unicidade da aproximação;
2. A aproximação não deve ter uma direção preferencial, ou seja, deve ter invariância geométrica;
3. Para garantir convergência, as incógnitas devem ser contínuas, e a aproximação

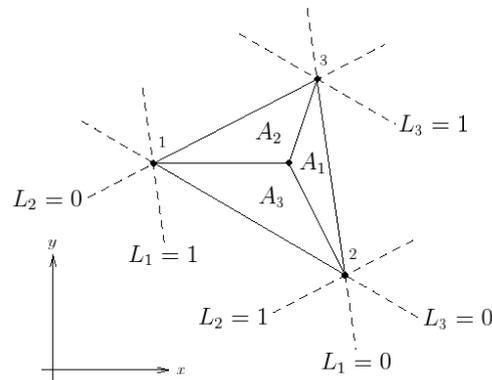


Figura 2.3: Determinação de funções de interpolação para um elemento triangular.

polinomial tal que permita com que elas assumam qualquer forma linear arbitrária.

Apesar disso, é difícil estabelecer regras gerais que possam ser aplicadas a todos os casos. Geralmente não é ideal acrescentar na aproximação polinomial termos de alta ordem ao custo da exclusão de outros de baixa ordem, o que pode causar oscilações na interpolação. Assim, o uso de polinômios completos é mais recomendável.

O problema de se determinar as funções de interpolação através do uso de aproximações polinomiais leva à inversão de matrizes mal-condicionadas. Por isso, a melhor forma de se obter tais funções é através de interpolação direta no elemento. Isto é feito escolhendo-se convenientemente um conjunto de polinômios de interpolação $N_i^e(\mathbf{x})$ de maneira que, se \mathbf{x}_j representa as coordenadas do nó j , então

$$N_i^e(\mathbf{x}_j) = \delta_{ij}, \quad i, j = 1, \dots, s \quad (2.15)$$

onde δ_{ij} representa o delta de *Kroneck*, dado por

$$\delta_{ij} = \begin{cases} 1, & \text{se } i = j, \\ 0, & \text{se } i \neq j. \end{cases} \quad (2.16)$$

É claro que as funções de forma obtidas desta maneira devem satisfazer as condições 1, 2 e 3 vistas anteriormente. Em particular, se as funções de forma satisfazem a condição 3, então

$$\sum_{i \in e} N_i^e = 1, \quad \sum_{i \in e} N_i^e \mathbf{x}_i = \mathbf{x}.$$

Para exemplificar o uso dessas funções, considera-se as as funções de interpolação para o elemento triangular. Apesar de ser possível a determinação de funções de interpolação em coordenadas globais para o triângulo, o uso de coordenadas locais simplifica e padroniza os cálculos na hora de se resolver as integrais do modelo variacional. Considere então um sistema de coordenadas locais no triângulo (L_1, L_2, L_3) , que na Figura 2.3 é dado por

$$L_1 = \frac{A_1}{A}, \quad L_2 = \frac{A_2}{A}, \quad L_3 = \frac{A_3}{A}, \quad (2.17)$$

onde $A = A_1 + A_2 + A_3$ é a área do triângulo e A_1 , A_2 e A_3 são as áreas mostradas na figura Figura 2.3. Note que, por exemplo, qualquer ponto sobre a aresta entre os vértices 1 e 2 tem coordenada $L_3 = 0$, e se o ponto coincide com o vértice 3, então $L_3 = 1$, uma vez que $A_3 = A$. Desta forma, a posição de qualquer ponto dentro do triângulo pode ser representada pelas coordenadas (L_1, L_2, L_3) , também chamadas de coordenadas baricêntricas. Note ainda que

$$L_1 + L_2 + L_3 = 1. \quad (2.18)$$

A relação entre o sistema de coordenadas global (x, y) e o sistema de coordenadas local (L_1, L_2, L_3) é dada pelas equações

$$x = L_1x_1 + L_2x_2 + L_3x_3, \quad (2.19)$$

$$y = L_1y_1 + L_2y_2 + L_3y_3, \quad (2.20)$$

que podem ser resolvidas para obter L_i em termos de x e y como

$$L_i(x, y) = \frac{a_i + b_i x + c_i y}{2A}, \quad (2.21)$$

onde a área A é dada por

$$A = \frac{1}{2} \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix}. \quad (2.22)$$

As constantes a_i , b_i e c_i são dadas em termos das coordenadas nodais

$$a_i = x_j y_k - x_k y_j, \quad (2.23)$$

$$b_i = y_j - y_k, \quad (2.24)$$

$$c_i = x_k - x_j, \quad (2.25)$$

onde j e k são permutações cíclicas de i . De (2.21) as seguintes relações para as

derivadas podem ser obtidas

$$\frac{\partial L_i}{\partial x} = \frac{b_i}{2A}, \quad \frac{\partial L_i}{\partial y} = \frac{c_i}{2A}. \quad (2.26)$$

Finalmente, um resultado sobre integração em coordenadas de área, importante para a discretização por elementos triangulares, é dado por

$$\int_A L_1^m L_2^n L_3^p dx dy = \frac{2Am!n!p!}{(m+n+p+2)!}, \quad (2.27)$$

cuja prova pode ser encontrada em (DAVIES, 1980).

Considerando um elemento triangular com um grau de liberdade em cada nó, a aproximação da variável ϕ no elemento segue a seguinte forma

$$\phi^e(x, y) = \alpha_0 + \alpha_1 x + \alpha_2 y, \quad (2.28)$$

e utilizando as coordenadas baricêntricas do triângulo e como funções de interpolação, isto é,

$$\mathbf{N}^e = \begin{bmatrix} L_1 & L_2 & L_3 \end{bmatrix}, \quad (2.29)$$

pode-se escrever

$$\phi^e(x, y) = \mathbf{N}^e(x, y) \boldsymbol{\delta}^e = \begin{bmatrix} L_1(x, y) & L_2(x, y) & L_3(x, y) \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix}. \quad (2.30)$$

Funções de interpolação para elementos de mais alta ordem podem ser determinadas, mas são facilmente encontradas em qualquer livro de elementos finitos, como (DAVIES, 1980) ou para mais detalhes (ZIENKIEWICZ; TAYLOR, 2000a).

Utilizando-se a teoria descrita acima e discretizando-se as equações de *Navier-Stokes*, em sua formulação variacional, no espaço (através do método de Taylor-Galerkin) e em relação ao tempo por um sistema semi-implícito (veja (SIMEONI, 2005) para maiores detalhes), têm-se

$$\mathbf{M}_\rho \left(\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} \right) + \mathbf{A} \mathbf{u}^n + \frac{1}{Re} \mathbf{K} \mathbf{u}^{n+1} - \mathbf{G} \mathbf{p}^{n+1} - \frac{1}{Fr^2} \mathbf{M}_\rho \mathbf{g} - \frac{1}{We} \mathbf{M} \mathbf{f} = \mathbf{0} \quad (2.31)$$

$$\mathbf{D} \mathbf{u}^{n+1} = \mathbf{0} \quad (2.32)$$

Os números Re , Fr e We são, respectivamente, os números de *Reynold*, *Froude* e

Weber, dados por

$$Re = \frac{\rho_0 LU}{\mu_0} = \frac{LU}{\nu_0},$$

$$Fr = \frac{U}{\sqrt{gL}} \mathbf{e}$$

$$We = \frac{\rho_0 LU^2}{\sigma_0}$$

onde ρ_0 , L , U , μ_0 , g e σ_0 são os valores de referência para massa específica, comprimento, velocidade, viscosidade, força gravitacional e coeficiente de tensão superficial. O valor ν_0 representa a viscosidade cinemática e é a razão entre a viscosidade aparente ou dinâmica μ_0 e a massa específica. As matrizes de (2.31) são dadas por

$$\mathbf{M}_\rho = \begin{bmatrix} \mathbf{M}_{\rho,x} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_{\rho,y} \end{bmatrix}_{2NV \times 2NV} \quad \mathbf{M} = \begin{bmatrix} \mathbf{M}_x & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_y \end{bmatrix}_{2NV \times 2NV}$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_x & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_y \end{bmatrix}_{2NV \times 2NV} \quad \mathbf{K} = \begin{bmatrix} \mathbf{K}_{xx} & \mathbf{K}_{xy} \\ \mathbf{K}_{yx} & \mathbf{K}_{yy} \end{bmatrix}_{2NV \times 2NV}$$

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_x \\ \mathbf{G}_y \end{bmatrix}_{2NV \times NP} \quad \mathbf{D} = \begin{bmatrix} \mathbf{D}_x & \mathbf{D}_y \end{bmatrix}_{NP \times 2NV}.$$

onde

$$\begin{aligned} \mathbf{M}_{\rho,x} &= \mathcal{I}_x(\mathbf{m}_\rho^e), & \mathbf{M}_{\rho,y} &= \mathcal{I}_y(\mathbf{m}_\rho^e), & \mathbf{M}_x &= \mathcal{I}_x(\mathbf{m}^e), & \mathbf{M}_y &= \mathcal{I}_y(\mathbf{m}^e), \\ \mathbf{K}_{xx} &= \mathcal{I}_x(\mathbf{k}_{xx}^e), & \mathbf{K}_{xy} &= \mathcal{I}_x(\mathbf{k}_{xy}^e), & \mathbf{K}_{yx} &= \mathcal{I}_y(\mathbf{k}_{yx}^e), & \mathbf{K}_{yy} &= \mathcal{I}_y(\mathbf{k}_{yy}^e), \\ \mathbf{A}_x &= \mathcal{I}_x(\mathbf{a}^e), & \mathbf{A}_y &= \mathcal{I}_y(\mathbf{a}^e), & \mathbf{G}_x &= \mathcal{I}_x(\mathbf{g}_x^e), & \mathbf{G}_y &= \mathcal{I}_y(\mathbf{g}_y^e), \\ & & \mathbf{D}_x &= \mathcal{I}_x(\mathbf{d}_x^e), & \mathbf{D}_y &= \mathcal{I}_y(\mathbf{d}_y^e), \end{aligned}$$

tal que as submatrizes \mathbf{m}_ρ^e , \mathbf{m}^e , \mathbf{a}^e , \mathbf{k}_{xx}^e , \mathbf{k}_{xy}^e , \mathbf{k}_{yx}^e , \mathbf{k}_{yy}^e , \mathbf{g}_x^e , \mathbf{g}_y^e , \mathbf{d}_x^e e \mathbf{d}_y^e , que são matrizes

definidas localmente para cada elemento, são dadas por

$$m_{\rho,ij}^e = \int_{\Omega^e} \rho^e N_i^e N_j^e d\Omega \quad (2.33)$$

$$m_{ij}^e = \int_{\Omega^e} N_i^e N_j^e d\Omega \quad (2.34)$$

$$a_{ij}^e = \int_{\Omega^e} \rho^e N_i^e \left((u^e - \hat{u}^e) \frac{\partial N_j^e}{\partial x} + (v^e - \hat{v}^e) \frac{\partial N_j^e}{\partial y} \right) d\Omega \quad (2.35)$$

$$k_{xx,ij}^e = \int_{\Omega^e} \mu^e \left(\frac{\partial N_i^e}{\partial x} \frac{\partial N_j^e}{\partial x} \right) d\Omega \quad (2.36)$$

$$k_{xy,ij}^e = \int_{\Omega^e} \mu^e \left(\frac{\partial N_i^e}{\partial y} \frac{\partial N_j^e}{\partial x} \right) d\Omega \quad (2.37)$$

$$k_{yx,ij}^e = \int_{\Omega^e} \mu^e \left(\frac{\partial N_i^e}{\partial x} \frac{\partial N_j^e}{\partial y} \right) d\Omega \quad (2.38)$$

$$k_{yy,ij}^e = \int_{\Omega^e} \mu^e \left(\frac{\partial N_i^e}{\partial y} \frac{\partial N_j^e}{\partial y} \right) d\Omega \quad (2.39)$$

$$g_{x,ik}^e = \int_{\Omega^e} \frac{\partial N_i^e}{\partial x} P_k^e d\Omega \quad (2.40)$$

$$g_{y,ik}^e = \int_{\Omega^e} \frac{\partial N_i^e}{\partial y} P_k^e d\Omega \quad (2.41)$$

$$d_{x,kj}^e = \int_{\Omega^e} \frac{\partial N_j^e}{\partial x} P_k^e d\Omega \quad (2.42)$$

$$d_{y,kj}^e = \int_{\Omega^e} \frac{\partial N_j^e}{\partial y} P_k^e d\Omega \quad (2.43)$$

O operador \mathcal{T} que aparece acima é um operador que monta as submatrizes de elemento nas submatrizes das matrizes de (2.31).

3 MÉTODO DA PROJEÇÃO

Os métodos para solução das equações de *Navier-Stokes* para escoamentos de fluidos podem ser, de maneira geral, classificados em métodos *acoplados* e métodos *segregados*.

Métodos acoplados (ZIENKIEWICZ; TAYLOR, 2000a; FORTUNA, 2000) buscam resolver o sistema completo de equações a cada ciclo computacional. Devido ao forte acoplamento entre velocidade e pressão, esta é a maneira mais imediata de se resolver estas equações. Tal abordagem, juntamente com as fortes não-linearidades vindas do termo convectivo, torna o problema acoplado computacionalmente difícil e muito caro de se resolver. No caso tridimensional, por exemplo, o sistema não-linear acoplado é composto de quatro equações (três de quantidade de movimento e uma de continuidade) para cada ponto computacional na malha que discretiza o domínio de solução.

Os métodos segregados buscam o desacoplamento entre as equações, separando o sistema não-linear multidimensional em problemas mais simples, que podem ser resolvidos sequencialmente. Dentre os métodos segregados, os que mais se destacaram foram os denominados métodos de projeção. Tal família de métodos foi primeiramente introduzida por Chorin (CHORIN, 1968), seguido por muitos outros autores, como por exemplo (HARLOW; WELCH, 1965) com o método MAC, (TOMÉ; MCKEE, 1994) com método GENSMAC e (PATANKAR, 1980) com o método SIMPLE.

Os métodos de projeção podem ainda ser classificados em métodos contínuos, métodos semi-discretos, ou de passo fracionário (GRESHO, 1990; GRESHO; CHAN, 1990), e métodos discretos, baseados em decomposição *LU* (PEROT, 1993; NI; KOMORI; MORLEY, 2003; LEE; OH; KIM, 2001).

3.1 Método da Projeção Discreto

O método da projeção discreto baseado em decomposição *LU* é obtido através da fatoração *LU* em blocos do sistema linear resultante, onde é feita a separação entre as incógnitas de velocidade e pressão após a discretização espacial e temporal das equações de governo. Tais equações, discretizadas no tempo por um esquema semi-implícito, são dadas por

$$\mathbf{M}_\rho \left(\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} \right) + \mathbf{A}\mathbf{u}^n + \frac{1}{Re} \mathbf{K}\mathbf{u}^{n+1} - \mathbf{G}\mathbf{p}^{n+1} - \frac{1}{Fr^2} \mathbf{M}_\rho \mathbf{g} - \frac{1}{We} \mathbf{M}\mathbf{f} = \mathbf{0} \quad (3.1)$$

$$\mathbf{D}\mathbf{u}^{n+1} = \mathbf{0} \quad (3.2)$$

A equação (2.31) pode ainda ser reescrita como

$$\left(\mathbf{M}_\rho + \frac{\Delta t}{Re} \mathbf{K} \right) \mathbf{u}^{n+1} = -\Delta t \left(\mathbf{A} \mathbf{u}^n - \mathbf{G} \mathbf{p}^{n+1} - \frac{1}{Fr^2} \mathbf{M}_\rho \mathbf{g} - \frac{1}{We} \mathbf{M} \mathbf{f} \right) + \mathbf{M}_\rho \mathbf{u}^n \quad (3.3)$$

que juntamente com (2.32), formam um sistema de equações que pode ser representado da seguinte forma

$$\begin{bmatrix} \mathbf{B} & -\Delta t \mathbf{G} \\ \mathbf{D} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u}^{n+1} \\ \mathbf{p}^{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{r}^n \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{bc}_1 \\ \mathbf{bc}_2 \end{bmatrix} \quad (3.4)$$

onde agora o sistema é escrito apenas para as incógnitas do problema, ou seja, $\mathbf{u}^{n+1} = [u_1^{n+1}, \dots, u_{N_u}^{n+1}, v_1^{n+1}, \dots, v_{N_v}^{n+1}]^T$, $\mathbf{p}^{n+1} = [p_1^{n+1}, \dots, p_{N_p}^{n+1}]^T$, sendo N_u , N_v e N_p o número de incógnitas (nós livres) para velocidade na direção x , velocidade na direção y e pressão respectivamente. A notação para as matrizes e vetores foi mantida a mesma por simplicidade. A matriz \mathbf{B} é dada por

$$\mathbf{B} = \mathbf{M}_\rho + \frac{\Delta t}{Re} \mathbf{K} \quad (3.5)$$

e o lado direito representa as grandezas conhecidas no tempo n ,

$$\mathbf{r}^n = -\Delta t \left(\mathbf{A} \mathbf{u}^n - \frac{1}{Fr^2} \mathbf{M}_\rho \mathbf{g} - \frac{1}{We} \mathbf{M} \mathbf{f} \right) + \mathbf{M}_\rho \mathbf{u}^n, \quad (3.6)$$

mais as condições de contorno que nada mais são do que as contribuições dos valores conhecidos de velocidade e pressão no lado direito do sistema.

O método consiste em decompor a matriz do sistema (3.4) através de uma fatoração por blocos. Em (LEE; OH; KIM, 2001) são apresentadas várias formas de se fatorar exatamente esta matriz, cada forma dando origem a uma família de métodos diferentes (nem todos práticos ou mesmo possíveis). Utilizando uma fatoração canônica LU por blocos, tem-se o seguinte sistema

$$\begin{bmatrix} \mathbf{B} & \mathbf{0} \\ \mathbf{D} & \Delta t \mathbf{D} \mathbf{B}^{-1} \mathbf{G} \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\Delta t \mathbf{B}^{-1} \mathbf{G} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{u}^{n+1} \\ \mathbf{p}^{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{r}^n \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{bc}_1 \\ \mathbf{bc}_2 \end{bmatrix} \quad (3.7)$$

O sistema dado em (3.7), se resolvido exatamente, dá origem a um método conhecido como método de *Uzawa* (CHANG; GIRALDO; PEROT, 2002). Contudo, tal solução é muito cara computacionalmente, pois envolve a inversão da matriz \mathbf{B} a cada iteração. Métodos práticos para a solução de (3.7) aproximam a inversa de \mathbf{B} pelas

matrizes \mathbf{B}_1 e \mathbf{B}_2 de modo que o sistema pode ser escrito como

$$\begin{bmatrix} \mathbf{B} & \mathbf{0} \\ \mathbf{D} & \Delta t \mathbf{D} \mathbf{B}_1 \mathbf{G} \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\Delta t \mathbf{B}_2 \mathbf{G} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{u}^{n+1} \\ \mathbf{p}^{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{r}^n \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{bc}_1 \\ \mathbf{bc}_2 \end{bmatrix} \quad (3.8)$$

Segundo (CHANG; GIRALDO; PEROT, 2002), diferentes aproximações para as matrizes \mathbf{B}_1 e \mathbf{B}_2 podem ser feitas de modo a controlar o erro feito na separação das equações. Se $\mathbf{B}_1 = \mathbf{B}_2$, então a equação da continuidade (2.32) é satisfeita exatamente, e todo o erro das aproximações aparece na equação de conservação de quantidade de movimento. Caso as aproximações sejam diferentes, o erro cometido na separação pode ser movido inteiramente da equação de conservação de quantidade de movimento para a equação da continuidade. Ainda não é claro qual tipo de erro é melhor, sendo a resposta para esta questão dependente do problema a ser resolvido. Contudo, é desejável, principalmente em problemas envolvendo escoamentos incompressíveis, que a equação da continuidade seja satisfeita exatamente (ainda que a nível discreto), para que haja conservação de massa no problema.

Uma aproximação comum para a inversa \mathbf{B}^{-1} para este problema é considerar $\mathbf{B}_1 = \mathbf{B}_2 = \mathbf{M}_\rho^{-1}$ (CHANG; GIRALDO; PEROT, 2002), e portanto, o sistema (3.8) fica

$$\mathbf{B} \tilde{\mathbf{u}} = \mathbf{r}^n + \mathbf{bc}_1 \quad (3.9)$$

$$\Delta t \mathbf{D} \mathbf{M}_\rho^{-1} \mathbf{G} \mathbf{p}^{n+1} = -\mathbf{D} \tilde{\mathbf{u}} + \mathbf{bc}_2 \quad (3.10)$$

$$\mathbf{u}^{n+1} = \tilde{\mathbf{u}} + \Delta t \mathbf{M}_\rho^{-1} \mathbf{G} \mathbf{p}^{n+1} \quad (3.11)$$

O procedimento para a solução das equações é dado na seguinte ordem:

1. Resolve-se $\tilde{\mathbf{u}}$ de (3.9);
2. Resolve-se \mathbf{p}^{n+1} de (3.10);
3. Encontra-se a velocidade final \mathbf{u}^{n+1} usando (3.11);

Métodos de ordem mais altas podem ser obtidos melhorando a aproximação da matriz inversa \mathbf{B}^{-1} . Outros métodos podem ser obtidos mudando-se a forma de decompor a matriz do sistema. Por exemplo, em (NI; KOMORI; MORLEY, 2003) é descrito um esquema de segunda ordem, obtido pela adição do termo $\mathbf{G} \mathbf{p}^n$ no lado direito do sistema \mathbf{r}^n .

3.1.1 Aproximação Agrupada

A aplicação do método descrito nesta seção envolve a inversão de uma matriz de massa \mathbf{M}_ρ . Para se obter a matriz do sistema do passo 2 do algoritmo acima (3.10),

deve-se inverter \mathbf{M}_ρ a cada iteração, não sendo uma opção computacionalmente atrativa. Uma alternativa, o que é conhecido na literatura por “*lumping*” consiste em aproximar \mathbf{M}_ρ por uma matriz diagonal $\mathbf{M}_{L\rho}$, definida por

$$m_{L\rho,ii} = \sum_{j=1}^N m_{\rho,ij} \quad (3.12)$$

onde N é a dimensão da matriz \mathbf{M}_ρ . Tal procedimento soma as contribuições referente às relações de vizinhança entre os vértices do elemento na diagonal da matriz. Para não afetar a restrição de divergência nula a nível discreto, esta aproximação é utilizada para os passos 2 e 3 do algoritmo. O algoritmo final é dado por

1. Resolve-se $\tilde{\mathbf{u}}$ de

$$\mathbf{B}\tilde{\mathbf{u}} = \mathbf{r}^n + \mathbf{bc}_1 \quad (3.13)$$

2. Resolve-se \mathbf{p}^{n+1} de

$$\Delta t \mathbf{D} \mathbf{M}_{L\rho}^{-1} \mathbf{G} \mathbf{p}^{n+1} = -\mathbf{D}\tilde{\mathbf{u}} + \mathbf{bc}_2 \quad (3.14)$$

3. Encontra-se a velocidade final \mathbf{u}^{n+1} usando

$$\mathbf{u}^{n+1} = \tilde{\mathbf{u}} + \Delta t \mathbf{M}_{L\rho}^{-1} \mathbf{G} \mathbf{p}^{n+1} \quad (3.15)$$

Observação 3.1 *Com o intuito de simplificar a notação, a partir de agora a matriz*

$$\begin{bmatrix} \mathbf{B} & -\Delta t \mathbf{G} \\ \mathbf{D} & \mathbf{0} \end{bmatrix}$$

passará a ser representada por

$$\begin{bmatrix} \mathbf{A} & \mathbf{G} \\ \mathbf{D} & \mathbf{0} \end{bmatrix}.$$

4 ITERAÇÕES DE KRYLOV

De forma prática, pode-se dizer que existem duas maneiras de resolver o problema $\mathbf{Ax} = \mathbf{b}$, em que a matriz \mathbf{A} é de grande porte e esparsa. Uma delas é a utilização de métodos diretos que de maneira geral se torna impraticável devido ao alto custo e por seus fatores se tornarem densos. Porém ainda pode-se utilizar alguma estratégia para explorar a esparsidade de \mathbf{A} , como o uso inteligente da estrutura de armazenamento ou a utilização de pivoteamentos especiais que minimizam preenchimento.

Além do método direto existem os métodos iterativos. Estes métodos geram uma sequência de soluções aproximadas $\{\mathbf{x}_{(k)}\}$ e somente utiliza a matriz \mathbf{A} em produtos matriz-vetor. A avaliação da qualidade de um método iterativo é medida por quanto rápido $\mathbf{x}_{(k)}$ converge, isto é, segundo o número de iterações e segundo a velocidade da redução do erro.

Há algumas dezenas de métodos de Krylov, como por exemplo (FREUND; GOLUB; NACHTIGAL, 1992; SAAD; VORST, 2000; SIMONCINI; SZYLD, 2007), porém, apresentaremos apenas três deles, pois foram utilizados no programa de testes do simulador.

4.1 O Funcionamento dos Métodos Iterativos

A idéia por trás dos métodos iterativos é substituir o sistema original por um sistema que se aproxima do original mas que pode ser facilmente resolvido e então aperfeiçoar o resultado. Isto é, ao invés de resolver $\mathbf{Ax} = \mathbf{b}$ pode-se resolver facilmente o sistema $\mathbf{M}\mathbf{x}_{(0)} = \mathbf{b}$ onde $\mathbf{x}_{(0)}$ é uma aproximação da solução exata \mathbf{x}^* . As iterações surgem quando pretende-se melhorar a solução encontrada.

Suponha que $\mathbf{x}^* = \mathbf{x}_{(0)} + \mathbf{e}$, onde o vetor $\mathbf{e} = \mathbf{x}^* - \mathbf{x}_{(0)}$ é o erro da solução $\mathbf{x}_{(0)}$, então

$$\mathbf{A}(\mathbf{x}_{(0)} + \mathbf{e}) = \mathbf{b} \implies \mathbf{A}\mathbf{e} = \mathbf{b} - \mathbf{A}\mathbf{x}_{(0)}$$

Assim, deve-se resolver um novo sistema, que ao ser resolvido aproximadamente, utilizando novamente a matriz \mathbf{M} , obtemos

$$\mathbf{M}\mathbf{e}_{(0)} = \mathbf{b} - \mathbf{A}\mathbf{x}_{(0)}$$

obtendo desta forma, uma nova aproximação $\mathbf{x}_{(1)} = \mathbf{x}_{(0)} + \mathbf{e}_{(0)}$. O procedimento de correção pode agora ser repetido para $\mathbf{x}_{(1)}$. Seguindo desta maneira constrói-se um método iterativo onde a solução da iteração $(i + 1)$ é

$$\begin{aligned}
\mathbf{x}_{(i+1)} &= \mathbf{x}_{(i)} + \mathbf{e}_{(i)} \\
&= \mathbf{x}_{(i)} + \mathbf{M}^{-1}(\mathbf{b} - \mathbf{A}\mathbf{x}_{(i)}) \\
&= \mathbf{x}_{(i)} + \tilde{\mathbf{b}} - \tilde{\mathbf{A}}\mathbf{x}_{(i)}.
\end{aligned}$$

É claro que nunca calcula-se \mathbf{M}^{-1} , mas sim, resolve-se o sistema $\mathbf{M}\tilde{\mathbf{b}} = \mathbf{b}$ para encontrar $\tilde{\mathbf{b}}$, e através do mesmo raciocínio encontra-se $\tilde{\mathbf{A}}$.

O método iterativo apresentado acima pode ser representado pelos seguintes passos

1. Calcular $\mathbf{r}_{(i)} = \mathbf{b} - \mathbf{A}\mathbf{x}_{(i)}$
2. Obter $\mathbf{e}_{(i)}$ de $\mathbf{M}\mathbf{e}_{(i)} = \mathbf{r}_{(i)}$
3. Calcular $\mathbf{x}_{(i)} = \mathbf{x}_{(i-1)} + \mathbf{e}_{(i-1)}$

Esta forma tem uma característica interessante pois apresenta \mathbf{M} como preconditionador como será visto adiante.

Diferentes escolhas da matriz \mathbf{M} geram diferentes métodos iterativos como Jacobi, Gauss-Seidel ou SOR dentre outros. Métodos iterativos construídos desta forma, são conhecidos como métodos iterativos estacionários.

Em (GREENBAUM, 1997) é feita uma análise de convergência do termo do erro que tende a zero, se e somente se $\lim_{i \rightarrow \infty} \|(\mathbf{I} - \mathbf{M}^{-1}\mathbf{A})^i\| = 0$, onde $\|\cdot\|$ é uma norma matricial.

Se escolhermos $\mathbf{M} = \mathbf{I}$, obtemos as iterações de Richardson:

$$\mathbf{x}_{(i+1)} = \mathbf{b} + (\mathbf{I} - \mathbf{A})\mathbf{x}_{(i)} = \mathbf{x}_{(i)} + \mathbf{r}_{(i)} \quad (4.1)$$

onde $\mathbf{r}_{(i)} = \mathbf{b} - \mathbf{A}\mathbf{x}_{(i)}$ é o resíduo da iteração i .

Através desta identidade, é possível relacionar o resíduo $\mathbf{r}_{(i+1)}$ com o resíduo inicial $\mathbf{r}_{(0)}$, ao multiplica-la por $-\mathbf{A}$ e depois adicionar o vetor \mathbf{b}

$$\begin{aligned}
\mathbf{b} - \mathbf{A}\mathbf{x}_{(i+1)} &= \mathbf{b} - \mathbf{A}\mathbf{x}_{(i)} - \mathbf{A}\mathbf{r}_{(i)} \\
\mathbf{r}_{(i+1)} &= (\mathbf{I} - \mathbf{A})\mathbf{r}_{(i)} \\
&= (\mathbf{I} - \mathbf{A})^{i+1}\mathbf{r}_{(0)} \\
&= \mathcal{P}_{i+1}(\mathbf{A})\mathbf{r}_{(0)}.
\end{aligned} \quad (4.2)$$

Assim $\mathcal{P}_{i+1}(\mathbf{A})$ representa um polinômio de potências de \mathbf{A} .

A partir das equações (4.1) e (4.2) podemos obter

$$\begin{aligned}\mathbf{x}_{(i+1)} &= \mathbf{x}_{(0)} + \mathbf{r}_{(0)} + \mathbf{r}_{(1)} + \mathbf{r}_{(2)} + \dots + \mathbf{r}_{(i)} \\ &= \mathbf{x}_{(0)} + \sum_{j=0}^i \mathcal{P}_j(\mathbf{A})\mathbf{r}_{(0)},\end{aligned}\quad (4.3)$$

que como vimos $\mathcal{P}_j(\mathbf{A})$ acima é um polinômio de potências de \mathbf{A} , e assim pode-se concluir que $\mathbf{x}_{i+1} \in \mathbf{x}_{(0)} + \text{span}\{\mathbf{r}_{(0)}, \mathbf{A}\mathbf{r}_{(0)}, \mathbf{A}^2\mathbf{r}_{(0)}, \dots, \mathbf{A}^i\mathbf{r}_{(0)}\}$. Este espaço gerado por \mathbf{A} e $\mathbf{r}_{(0)}$ é conhecido como subespaço de Krylov e é representado por $\mathcal{K}^m(\mathbf{A}; \mathbf{r}_{(0)})$, onde m representa a dimensão deste espaço.

Se $\mathbf{x}_{(0)} = \mathbf{0}$ então podemos ver, facilmente, que $\mathbf{x}_{i+1} \in \mathcal{K}^m(\mathbf{A}; \mathbf{b})$, pois $\mathbf{r}_{(0)} = \mathbf{b}$ na equação (4.3).

Em outras palavras, podemos enunciar o seguinte teorema

Teorema 4.1 (Existência da solução de Krylov) *Um sistema linear quadrado $\mathbf{Ax} = \mathbf{b}$ tem uma solução de Krylov se e somente se $\mathbf{b} \in \text{Im}(\mathbf{A}^i)$, onde i é o índice¹ do autovalor zero de \mathbf{A} .*

Para a construção dos algoritmos dos métodos iterativos de Krylov é necessário construir o espaço de Krylov e em alguns casos apenas alguns vetores deste espaço são suficientes.

Teorema 4.2 *Se o polinômio minimal (ver (5.3)) de uma matriz não singular \mathbf{A} tem grau m , então a solução para $\mathbf{Ax} = \mathbf{b}$ pertence ao espaço $\mathcal{K}^m(\mathbf{A}, \mathbf{b})$.*

Na tentativa de construir uma base para este espaço, pode-se optar por uma base óbvia $\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \dots, \mathbf{A}^{i-1}\mathbf{r}_0$. No entanto, esta base não é muito atrativa do ponto de vista numérico, desde que os vetores $\mathbf{A}^j\mathbf{r}_0$, de acordo com o crescimento de j , apontam cada vez mais em direção ao autovetor dominante, isto é, ao autovetor correspondente ao maior autovalor. Esta forma de construção é conhecida como *método das potências*, e o problema com este método é que os vetores da base tornam-se linearmente dependentes em aritmética finita.

A seguir descreveremos alguns métodos baseados em projeções em espaços de Krylov. Primeiro, o gradiente conjugado, que constrói o espaço de Krylov a partir do *método de Lanczos* (LANCZOS, 1950; LANCZOS, 1952). Em seguida, o GMRes e o Bi-CGStab, ambos constroem o espaço de Krylov a partir do *método de Arnoldi* (ARNOLDI, 1951).

¹Para definição de índice do autovalor ver (MEYER, 2000) (p.394)

4.2 Gradiente Conjugado (CG)

Este algoritmo é, primeiramente, derivado de outro método iterativo mais simples, o *Steepest Descent* (SHEWCHUK, 1994). Diferentemente do *Steepest Descent*, o CG trabalha para não seguir mais de uma vez em uma mesma direção. Este procedimento é feito através da idéia do método das direções conjugadas, de forma que elas são construídas pela conjugação dos resíduos. No entanto, o grande ganho está em reconhecê-lo como um método que tenta gerar uma boa aproximação para o espaço de Krylov.

Devido ao vasto material já existente na literatura, iremos apenas apresentar, aqui, o algoritmo. Para conhecer detalhes deste método, as seguintes referências podem ser consultadas: (GOLUB; O'LEARY, 1989), (GOLUB; LOAN, 1996; HESTENES; STIEFEL, 1952; MEURANT, 2006), (REID, 1971; SHEWCHUK, 1994).

Assim, segue o algoritmo do método do gradiente conjugado

$$\begin{aligned}
 d_{(0)} &= r_{(0)} = b - Ax_{(0)}, \\
 \alpha_{(i)} &= \frac{r_{(i)}^T r_{(i)}}{d_{(i)}^T A d_{(i)}}, \\
 x_{(i+1)} &= x_{(i)} + \alpha_{(i)} d_{(i)}, \\
 r_{(i+1)} &= r_{(i)} + \alpha_{(i)} A d_{(i)}, \\
 \beta_{(i+1)} &= \frac{r_{(i+1)}^T r_{(i+1)}}{r_{(i)}^T r_{(i)}}, \\
 d_{(i+1)} &= r_{(i+1)} + \beta_{(i+1)} d_{(i)}.
 \end{aligned}$$

que pode ser melhor apresentado por

Algoritmo do CG

1. Escolha $\mathbf{x}_{(0)}$, atribua $i = 0$ e calcule $\mathbf{d}_{(0)} = \mathbf{r}_{(0)} = \mathbf{b} - \mathbf{A}\mathbf{x}_{(0)}$ e $res_{(0)} = \|\mathbf{r}_{(0)}\|^{\frac{1}{2}}$.
 2. *While* ($i < i_{max}$) e ($res_{(i)} > tol$) *Do*
 3. $\mathbf{q}_{(i)} = \mathbf{A}\mathbf{d}_{(i)}$;
 4. $\alpha_{(i)} = \frac{res_{(i)}}{\mathbf{d}_{(i)}^T \mathbf{q}_{(i)}}$;
 5. $\mathbf{x}_{(i+1)} = \mathbf{x}_{(i)} + \alpha_{(i)} \mathbf{d}_{(i)}$;
 6. $\mathbf{r}_{(i+1)} = \mathbf{r}_{(i)} - \alpha_{(i)} \mathbf{q}_{(i)}$;
 7. $res_{(i+1)} = \mathbf{r}_{(i+1)}^T \mathbf{r}_{(i+1)}$;
 8. $\beta = \frac{res_{(i+1)}}{res_{(i)}}$;
 9. $\mathbf{d}_{(i+1)} = \mathbf{r}_{(i+1)} + \beta \mathbf{d}_{(i)}$;
 10. $i \leftarrow i + 1$;
 11. *End.*
-

4.3 GMRes

O método do Resíduo Mínimo Generalizado (GMRes) foi proposto por Saad (SAAD; SCHULTZ, 1986). Este é um método iterativo, para resolução de sistemas lineares não simétricos, onde a cada passo a norma do resíduo é minimizada sobre um subespaço de Krylov escolhido. Este método requer grandes recorrências e atende a certas condições de otimalidade. Tal método é derivado do método de Arnoldi, no qual se obtém uma base ortonormal para um espaço de Krylov escolhido. Além disso, pode-se optar pelo uso de "restart" para controlar as necessidades de armazenamento.

Para resolver o sistema $\mathbf{Ax} = \mathbf{b}$, dada uma estimativa inicial $\mathbf{x}_{(0)}$, o algoritmo constroi sucessivas aproximações $\mathbf{x}_{(k)}$, com $k = 1, 2, \dots$, do espaço afim

$$\mathbf{x}_{(0)} + \text{span}\{\mathbf{r}_{(0)}, \mathbf{A}\mathbf{r}_{(0)}, \dots, \mathbf{A}^{k-1}\mathbf{r}_{(0)}\},$$

onde $\mathbf{r}_{(0)} \equiv \mathbf{b} - \mathbf{A}\mathbf{x}_{(0)}$ é o resíduo inicial.

Tal algoritmo, usa o algoritmo de Arnoldi (ARNOLDI, 1951), determinando, assim, uma base ortonormal \mathbf{V}_m e a matriz de Hessenberg \mathbf{H}_m . Assim, tem-se que a igualdade abaixo

$$\mathbf{V}_{i+1}^H \mathbf{A} \mathbf{V}_i = \mathbf{H}_{i+1,i}$$

é válida e pode ser reescrita como

$$\mathbf{A} \mathbf{V}_i = \mathbf{V}_{i+1} \mathbf{H}_{i+1,i}.$$

O GMRES tem como ponto principal minimizar a norma do resíduo sobre \mathcal{K}^i

(representando $\mathcal{K}^i(\mathbf{A}; \mathbf{r}_{(0)})$), ou seja, o que se quer é resolver o seguinte problema

$$\min_{\delta_i \in \mathcal{K}^i} \|b - A(x_{(0)} + \delta_i)\| = \min_{\delta_i \in \mathcal{K}^i} \|r_{(0)} - A\delta_i\|. \quad (4.4)$$

Algoritmo do GMRes

1. Escolha $\mathbf{x}_{(0)}$ e calcule $\mathbf{r}_{(0)} = \mathbf{b} - \mathbf{A}\mathbf{x}_{(0)}$ e $\mathbf{v}_{(1)} = \frac{\mathbf{r}_{(0)}}{\|\mathbf{r}_{(0)}\|}$.
 2. *for* $j = 1, \dots, m$
 3. $h_{ij} = \langle \mathbf{A}\mathbf{v}_{(j)}, \mathbf{v}_{(i)} \rangle, i = 1, \dots, j$;
 4. $\hat{\mathbf{v}}_{(j+1)} = \mathbf{A}\mathbf{v}_{(j)} - \sum_{i=1}^j h_{i,j} \mathbf{v}_{(i)}$;
 5. $h_{j+1,j} = \|\hat{\mathbf{v}}_{(j+1)}\|$; e
 6. $\mathbf{v}_{(j+1)} = \frac{\hat{\mathbf{v}}_{(j+1)}}{h_{j+1,j}}$;
 7. *end*.
 8. Forme a solução: $\mathbf{x}_{(m)} = \mathbf{x}_{(0)} + \mathbf{V}_m \mathbf{y}_m$ onde \mathbf{y}_m minimiza $J(\mathbf{y}_{(i)}) = \|\beta \mathbf{e}_1 - \mathbf{H}_m \mathbf{y}_m\|$, onde $\beta = \|\mathbf{r}_{(0)}\|$.
-

É natural que seja levantada a hipótese de o algoritmo falhar. Portanto, é importante observar os resultados a seguir, em que suas demonstrações podem ser encontradas em (SAAD; SCHULTZ, 1986) e (SAAD, 2003).

Teorema 4.3 *A solução $\mathbf{x}_{(j)}$ produzida pelo GMRes no passo (j) é exata se e somente se as seguintes condições equivalentes ocorrem:*

1. *O algoritmo falha no passo (j) .*
2. $\hat{\mathbf{v}}_{(j+1)} = 0$.
3. $h_{j+1,j} = 0$.
4. *O grau do polinômio mínimo do resíduo inicial $\mathbf{r}_{(0)}$ é igual a j .*

Corolário 4.3.1 *Para uma matriz de ordem $n \times n$ o GMRes termina em no máximo n passos.*

Teorema 4.4 *Seja \mathbf{A} uma matriz não-singular. Então o algoritmo do GMRes falha no passo j , isto é, $h_{j+1,j} = 0$, se a solução aproximada, $\mathbf{x}_{(j)}$, é a solução exata.*

4.4 BiCGStab

O método do Gradiente Bi-Conjugado Estabilizado (VORST, 1992) aplica-se para resolução de sistemas lineares com matrizes não-simétricas. Em contraste com

o GMRes, não satisfaz certas condições de otimalidade mas tem custo computacional fixo a cada passo.

O Bi-CGStab pode ser obtido a partir do Bi-CG (FLETCHER, 1976) e, como o CGS (SONNEVELD, 1989), este não usa a matriz conjugada para operações do tipo matriz-vetor.

Abaixo, encontra-se o algoritmo do método do Bi-CGStab.

Algoritmo do Bi-CGStab

1. Escolha $\mathbf{x}_{(0)}$. Calcule $\mathbf{r}_{(0)} = \mathbf{b} - \mathbf{A}\mathbf{x}_{(0)}$
 2. Escolha $\hat{\mathbf{r}}_{(0)}$ tal que $\langle \mathbf{r}_{(0)}, \hat{\mathbf{r}}_{(0)} \rangle \neq 0$, por exemplo, $\mathbf{r}_{(0)} = \hat{\mathbf{r}}_{(0)}$
 3. $\rho_{(0)} = \alpha = \omega_{(0)} = 1$
 4. $\mathbf{v}_{(0)} = \mathbf{p}_{(0)} = \mathbf{0}$
 5. Para $i = 1, 2, 3, \dots$. **Do**
 6. $\rho_{(i)} = \langle \mathbf{r}_{(0)}, \hat{\mathbf{r}}_{(0)} \rangle; \beta = \left(\frac{\rho_{(i)}}{\rho_{(i-1)}} \right) \left(\frac{\alpha}{\omega_{(i-1)}} \right)$
 7. $\mathbf{p}_{(i)} = \mathbf{r}_{(i-1)} + \beta(\mathbf{p}_{(i-1)} - \omega_{(i-1)}\mathbf{v}_{(i-1)})$
 8. $\mathbf{v}_{(i)} = \mathbf{A}\mathbf{p}_{(i)}$
 9. $\alpha = \frac{\rho_{(i)}}{\langle \hat{\mathbf{r}}_{(0)}, \mathbf{v}_{(i)} \rangle}$
 10. $\mathbf{s} = \mathbf{r}_{(i-1)} - \alpha\mathbf{v}_{(i)}$
 11. $\mathbf{t} = \mathbf{A}\mathbf{s}$
 12. $\omega_{(i)} = \frac{\langle \mathbf{t}, \mathbf{s} \rangle}{\langle \mathbf{t}, \mathbf{t} \rangle}$
 13. $\mathbf{x}_{(i)} = \mathbf{x}_{(i-1)} + \alpha\mathbf{p}_{(i)} + \omega_{(i)}\mathbf{s}$
 14. se $\mathbf{x}_{(i)}$ "is accurate enough" **EndDo**
 15. $\mathbf{r}_{(i)} = \mathbf{s} - \omega_{(i)}\mathbf{t}$
 16. Fim
-

4.5 Mínimos Quadrados e Equações Normais

Considere um sistema de equações lineares com n equações e m incógnitas onde $n > m$. Desta forma, deseja-se encontrar um vetor $\mathbf{x} \in \mathbb{R}^m$ que satisfaz $\mathbf{A}\mathbf{x} = \mathbf{b}$, onde $\mathbf{A} \in \mathbb{R}^{n \times m}$ e $\mathbf{b} \in \mathbb{R}^n$. É possível que este seja um problema sem solução a menos que $\mathbf{b} \in \text{Im}(\mathbf{A})$. Diz-se que o sistema de equações com $n > m$ é *sobredeterminado*.

O vetor resíduo $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x} \in \mathbb{R}^n$ pode ser bem pequeno dependendo do vetor \mathbf{x} , porém é possível que não possa ser nulo.

Assim, o objetivo é minimizar o resíduo tanto quanto possível. Escolhendo a norma 2 para medir o resíduo o problema toma a seguinte forma

Dado $\mathbf{A} \in \mathbb{R}^{n \times m}$, $n \geq m$, $\mathbf{b} \in \mathbb{R}^n$,
encontrar $\mathbf{x} \in \mathbb{R}^m$ tal que $\|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2$ seja minimizada.

Teorema 4.5 (TREFETHEN; Bau, III, 1997) *Seja $\mathbf{A} \in \mathbb{R}^{n \times m}$ ($n \geq m$) e $\mathbf{b} \in \mathbb{R}^n$ dado. Um vetor $\mathbf{x} \in \mathbb{R}^m$ minimiza a norma do resíduo $\|\mathbf{r}\|_2 = \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2$, ao resolver o*

problema de mínimos quadrados (4.5), se e somente se $\mathbf{r} \perp \text{Im}(A)$, isto é, $\mathbf{A}^T \mathbf{r} = 0$, ou equivalentemente,

$$\mathbf{A}^T \mathbf{A} \mathbf{x} = \mathbf{A}^T \mathbf{b} \quad (4.5)$$

O sistema de equações (4.5) de dimensões $m \times m$ é conhecido como equações normais, e é não singular se e somente se \mathbf{A} tem posto completo. Consequentemente a solução \mathbf{x} é única se \mathbf{A} tem posto completo.

A forma clássica de resolver um problema de mínimos quadrados é resolver o sistema de equações normais associado a ele. Se \mathbf{A} tem posto completo, a matriz $\mathbf{A}^T \mathbf{A}$ é quadrada, simétrica e positiva definida, permitindo a utilização do PCG com Cholesky.

Para sistemas lineares cuja matriz $\mathbf{A} \in \mathbb{R}^{m \times n}$ onde $n > m$, isto é, tem mais equações do que incógnitas, consideraremos a matriz \mathbf{A} como sendo de posto completo. Um sistema linear composto desta forma é chamado de sistema *subdeterminado* e apresenta infinitas soluções. O sistema de equações normais associado a ele tem a seguinte forma

$$\mathbf{A} \mathbf{A}^T \mathbf{u} = \mathbf{b}, \quad \mathbf{x} = \mathbf{A}^T \mathbf{u}. \quad (4.6)$$

Considere \mathbf{x}^* como qualquer solução do sistema subdeterminado $\mathbf{A} \mathbf{x} = \mathbf{b}$. Então (4.5) representa a equação normal associada ao seguinte problema de mínimos quadrados

$$\text{minimizar } \|\mathbf{x}^* - \mathbf{A}^T \mathbf{u}\|_2,$$

que representa a minimização do erro.

Utilizar equação normais para resolver problemas de mínimos quadrados podem gerar problemas significativos como a obtenção de matrizes $\mathbf{A}^T \mathbf{A}$ ou $\mathbf{A} \mathbf{A}^T$ densas e o aumento do número de condicionamento da matriz do sistema

$$\text{Cond}_2(\mathbf{A}^T \mathbf{A}) = \|\mathbf{A}\|_2^2 \|\mathbf{A}^{-1}\|_2^2 = \text{Cond}_2^2(A).$$

Para tentar contornar estes problemas, pode-se utilizar o método do gradiente conjugado adaptado ao problema de mínimos quadrados que não montam a matriz dos coeficientes explicitamente e a utilização de preconditionamento para reduzir o número de condicionamento.

4.5.1 Matriz de Ponto de Sela e CGLS

Uma estratégia alternativa para a resolução do sistema fatorado (1.5) consiste em não aproximar \mathbf{A}^{-1} para não perder qualidade de aproximação da solução. No

entanto, é necessária a resolução de sistemas lineares retangulares em

$$-\mathbf{DA}^{-1}\mathbf{G}\mathbf{y}_2 = \bar{\mathbf{b}}, \quad (4.7)$$

onde $\bar{\mathbf{b}} = -\mathbf{b}_2 + \mathbf{D}\mathbf{y}_1$. Já que não se conhece \mathbf{A}^{-1} , faz-se

$$\mathbf{D}\mathbf{z} = \bar{\mathbf{b}} \quad (4.8)$$

onde $\mathbf{z} = \mathbf{A}^{-1}\mathbf{G}\mathbf{y}_2$ e \mathbf{D} tem dimensão $m \times n$ em que $n > m$.

Após resolver este sistema subdeterminado, deve-se resolver $\mathbf{A}^{-1}\mathbf{G}\mathbf{y}_2 = \mathbf{z}$, e como não é interessante o uso de \mathbf{A}^{-1} , utiliza-se o sistema

$$\mathbf{G}\mathbf{y}_2 = \mathbf{A}\mathbf{z}, \quad (4.9)$$

onde \mathbf{G} tem dimensão $n \times m$ tendo, portanto, que resolver um sistema sobredeterminado para obter \mathbf{y}_2 .

Além deste desenvolvimento para a resolução da equação (4.7), precisa-se de outra forma de resolução para

$$\mathbf{x}_1 = \mathbf{y}_1 - \mathbf{A}^{-1}\mathbf{G}\mathbf{y}_2$$

já que o intuito é não utilizar \mathbf{A}^{-1} . Então pode-se proceder como já mostrado anteriormente, utiliza-se o gradiente conjugado preconditionado para resolver

$$\mathbf{A}\mathbf{x}_1 = \mathbf{b}_1 - \mathbf{G}\mathbf{y}_2.$$

Dois momentos importantes são as resoluções dos sistemas subdeterminado e sobredeterminado. Uma forma de resolução para (4.8) é feita através do seguinte sistema equivalente

$$\mathbf{DD}^T\bar{\mathbf{z}} = \bar{\mathbf{b}} \quad (4.10)$$

em que $\mathbf{z} = \mathbf{D}^T\bar{\mathbf{z}}$.

E para o sistema (4.9) sua resolução é feita através do seguinte sistema de equações

$$\mathbf{G}^T\mathbf{G}\mathbf{y}_2 = \bar{\bar{\mathbf{z}}} \quad (4.11)$$

onde $\bar{\bar{\mathbf{z}}} = \mathbf{G}^T\mathbf{A}\mathbf{z}$.

Uma outra maneira de se resolver os sistemas (4.8) e (4.9) é utilizar uma adaptação do gradiente conjugado para sistemas retangulares.

O sistema (4.10) e métodos derivados dele são associados à sigla NE (N

de "normal" e E de "error") por minimizar a norma do erro. Enquanto que (4.11) e métodos derivados dele são associados à sigla NR (N de "normal" e R de "residual") por minimizar a norma do resíduo. Assim, os métodos de gradiente conjugado associados a estes sistemas são chamados, respectivamente, CGNE e CGNR, podendo ainda serem chamados de CGLS (L de "least" e S de "square") e podem ser encontrados em (BARRETT et al., 1994; SAAD, 2003).

5 PRECONDICIONAMENTO

Os métodos baseados em espaço de Krylov, descritos no capítulo anterior, convergem rapidamente se a matriz dos coeficientes \mathbf{A} se aproxima, de alguma forma, da matriz identidade \mathbf{I} , porém isto não é comum. A taxa de convergência de métodos iterativos depende, em muito casos importantes, de propriedades espectrais da matriz de coeficientes, que pode até mesmo não permitir que as iterações convirjam como pode ser visto em (BENZI, 2002; GREENBAUM, 1997; GOLUB; LOAN, 1996; SAAD, 2003; VORST, 2003).

Então, ao invés de resolver este problema, que é difícil, podemos transformar tal sistema em um outro sistema que é equivalente ao original, isto é, com a mesma solução, porém com melhores propriedades espectrais ou, pensando de forma mais simples, transformando a matriz original em uma nova matriz que de alguma maneira se aproxima da matriz identidade.

O preconditionamento, que é o processo de transformação do sistema original em outro equivalente porém mais simples, é essencial para a grande maioria das aplicações de métodos iterativos, devendo ser utilizado para garantir a viabilidade dos métodos iterativos de Krylov aplicados aos problemas reais.

Nesse capítulo vamos discutir de vários preconditionadores. Na seção 5.1

5.1 Precondicionando $\mathbf{Ax}=\mathbf{b}$

A idéia do processo de preconditionamento é elementar. Suponha que queiramos resolver o seguinte sistema não singular de n equações e n incógnitas

$$\mathbf{Ax} = \mathbf{b} \quad (5.1)$$

Para qualquer matriz não singular \mathbf{M} de dimensões $n \times n$, os sistemas

$$\mathbf{M}^{-1}\mathbf{Ax} = \mathbf{M}^{-1}\mathbf{b} \quad (5.2)$$

e

$$\mathbf{AM}^{-1}\hat{\mathbf{x}} = \mathbf{b} \quad \text{com} \quad \mathbf{x} = \mathbf{M}^{-1}\hat{\mathbf{x}}. \quad (5.3)$$

têm a mesma solução que o original (5.1).

Diz-se que as equações (5.2) e (5.3) estão preconditionadas pela esquerda e pela direita, respectivamente.

Ao aplicar o condicionamento, espera-se que a matriz \mathbf{M} chamada *precondicionador*, seja uma boa aproximação para \mathbf{A} e que obtenha-se, desta forma, as propriedades favoráveis desejadas para a matriz deste novo sistema $\mathbf{M}^{-1}\mathbf{A}$ que é chamada de matriz condicionada pela esquerda, neste caso, fazendo assim referência a matriz original do sistema e à forma com que ela foi condicionada.

Em alguns casos, como no Gradiente Conjugado, o condicionamento deve ser feito de forma que a matriz condicionada seja simétrica, então deve-se utilizar a seguinte forma de aplicação do condicionador conhecida como *split*

$$\mathbf{M}_1^{-1}\mathbf{A}\mathbf{M}_2^{-1}\hat{\mathbf{x}} = \mathbf{M}_1^{-1}\mathbf{b} \quad \text{com} \quad \mathbf{x} = \mathbf{M}_2^{-1}\hat{\mathbf{x}}. \quad (5.4)$$

onde $\mathbf{M} = \mathbf{M}_1\mathbf{M}_2$. A matriz \mathbf{M}_1 pode ser o fator triangular inferior da decomposição de Cholesky de \mathbf{M} , por exemplo.

As três formas de condicionamento apresentadas podem ser consideradas equivalentes, sob algum ponto de vista, pois garantem as mesmas propriedades espectrais à matriz condicionada.

Definição 5.1 *Uma matriz $\mathbf{A} \in \mathbb{R}^{n \times n}$ é dita ser similar a uma matriz $\mathbf{B} \in \mathbb{R}^{n \times n}$ se existe uma matriz $\mathbf{S} \in \mathbb{R}^{n \times n}$ tal que*

$$\mathbf{B} = \mathbf{S}^{-1}\mathbf{A}\mathbf{S}.$$

Teorema 5.1 *Sejam $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$. Se \mathbf{A} e \mathbf{B} são similares, então seus polinômios característicos são os mesmos. E com isso possuem os mesmos autovalores com suas respectivas multiplicidades.*

As matrizes $\mathbf{A}\mathbf{M}^{-1}$, $\mathbf{M}^{-1}\mathbf{A}$ e $\mathbf{M}_1^{-1}\mathbf{A}\mathbf{M}_2^{-1}$ são similares e assim têm os mesmos autovalores.

Para que a idéia de condicionamento seja viável deve-se ser capaz de efetuar a multiplicação $\mathbf{M}^{-1}\mathbf{A}$, $\mathbf{A}\mathbf{M}^{-1}$ ou $\mathbf{M}_1^{-1}\mathbf{A}\mathbf{M}_2^{-1}$ mesmo que de forma implícita. Estes fatores não são construídos explicitamente, pois a matriz condicionada não seria esparsa e além disso, inverter \mathbf{M} não costuma ser viável. Mais vantajoso que uma inversão é resolver o sistema $\mathbf{M}\mathbf{y} = \mathbf{c}$. Este sistema surge naturalmente como consequência da aplicação do condicionador como será visto nas seções 5.8, 5.9 e 5.10 sobre aplicação dos condicionadores nos métodos iterativos.

Assim a escolha do condicionador deve ser feita de forma que

1. O sistema condicionado seja fácil de ser resolvido.
2. O condicionador seja fácil de se construir e aplicar.

A primeira propriedade quer dizer que, após o sistema ser preconditionado, os métodos devem convergir rapidamente, enquanto a segunda afirma que cada iteração não deve ser muito cara. Na realidade, deve-se conseguir que o processo de construção e aplicação do preconditionador seja compensado na resolução do sistema preconditionado em relação ao original. Repare que os dois requisitos acima competem um com o outro.

Vejamos os casos extremos. Se $\mathbf{M} = \mathbf{A}$ então resolver $\mathbf{M}\mathbf{y} = \mathbf{c}$ é tão difícil quanto resolver $\mathbf{A}\mathbf{x} = \mathbf{b}$ tornando a aplicação do preconditionador tão difícil quanto resolver o problema original. Se $\mathbf{M} = \mathbf{I}$ então aplicar o preconditionador é trivial porém a dificuldade original permanece.

Então, como avaliar de antemão se determinado preconditionador é bom o suficiente? Existem diversas características que podem dizer se determinado preconditionador é suficientemente bom, mas uma boa e simples resposta para esta pergunta é:

Um preconditionador é bom se $\mathbf{M}^{-1}\mathbf{A}$ não está muito distante de ser normal¹ e seus autovalores são agrupados (TREFETHEN; Bau, III, 1997).

Observação 5.1 *Também pode ser visto em (TREFETHEN; Bau, III, 1997) que os autovalores de $\mathbf{M}^{-1}\mathbf{A}$ podem estar aglomerados em torno de outro valor que não o 1, e podem existir autovalores bem afastados dos demais, para que \mathbf{M} seja um bom preconditionador. Para alguns métodos, por exemplo, é suficiente que os valores singulares de $\mathbf{M}^{-1}\mathbf{A}$ estejam aglomerados e não os autovalores. Para outras características ver (GREENBAUM, 1997) (p.120).*

Vamos introduzir mais algumas definições de álgebra linear que permitam avançar na análise dos preconditionadores. Nesse contexto estamos considerando que o corpo de escalares sobre o qual as matrizes estão operando é o corpo dos complexos.

Definição 5.2 *O polinômio característico de $\mathbf{A} \in \mathbb{R}^{n \times n}$ é definido por*

$$p_{\mathbf{A}}(t) \equiv \det(t\mathbf{I} - \mathbf{A}).$$

Assim, o conjunto das raízes deste polinômio forma o conjunto dos autovalores de \mathbf{A} .

Teorema 5.2 *Seja $\mathbf{A} \in \mathbb{R}^{n \times n}$ dada. Existe um único polinômio mônico $q_{\mathbf{A}}(t)$ de grau mínimo tal que $q_{\mathbf{A}}(\mathbf{A}) = \mathbf{0}$. O grau deste polinômio é de no máximo n . Se $p(t)$ é qualquer polinômio tal que $p(\mathbf{A}) = \mathbf{0}$, então $q_{\mathbf{A}}(t)$ divide $p(t)$.*

¹Uma matriz normal comuta com a sua hermitiana, ou seja: $\mathbf{A}^H \mathbf{A} = \mathbf{A} \mathbf{A}^H$, onde \mathbf{A}^H é a conjugada transposta de \mathbf{A} .

Definição 5.3 Seja $\mathbf{A} \in \mathbb{R}^{n \times n}$ dada. O único polinômio mônico $q_{\mathbf{A}}(t)$ de grau mínimo tal que $q_{\mathbf{A}}(\mathbf{A}) = \mathbf{0}$ é chamado de polinômio minimal de \mathbf{A} .

Corolário 5.2.1 Matrizes similares têm o mesmo polinômio minimal.

Corolário 5.2.2 Para toda matriz $\mathbf{A} \in \mathbb{R}^{n \times n}$, o polinômio minimal $p_{\mathbf{A}}(t)$ divide o polinômio característico $p_{\mathbf{A}}(t)$. Além disso, $q_{\mathbf{A}}(\lambda) = 0$ se e somente se λ é um autovalor de \mathbf{A} . Então, toda raiz de $p_{\mathbf{A}}(t) = 0$ é raiz de $q_{\mathbf{A}}(t) = 0$.

Uma outra alternativa encontrada na literatura (BENZI; GOLUB; LIESEN, 2005) para se obter um bom preconditionador \mathbf{M} que aproxima \mathbf{A} seria obter diretamente uma matriz \mathbf{M} que aproxime \mathbf{A}^{-1} pois, neste caso, precisaria apenas acrescentar mais uma multiplicação ao algoritmo e não uma inversão ou fatoração. Os preconditionadores dessa forma são denominados “aproximantes da inversa” ou “explícitos”. Não trataremos desses preconditionadores no restante desse trabalho.

5.2 Diagonal

Talvez um dos mais simples seja o preconditionador *diagonal* ou *preconditionador de Jacobi* dado por $\mathbf{M} = \text{diag}(\mathbf{A})$, o que significa que, \mathbf{M} é uma matriz diagonal que é composta exatamente da diagonal da matriz \mathbf{A} , para \mathbf{A} não singular, assim

$$m_{ij} = \begin{cases} a_{ij} & \text{se } i = j \\ 0 & \text{caso contrário} \end{cases}$$

representando um simples escalamento.

Para muitos problemas esta transformação é suficiente para tornar rápida uma convergência lenta.

Preconditionadores de Jacobi exigem muito pouco armazenamento e são fáceis de implementar mesmo em paralelo. Por outro lado, em geral, preconditionadores mais sofisticados e caros apresentam melhores resultados.

5.3 Fatorações Incompletas

O processo de eliminação Gaussiana é equivalente a fatoração da matriz \mathbf{A} na forma $\mathbf{PA} = \mathbf{LU}$, onde \mathbf{L} é triangular inferior com diagonal unitária e \mathbf{U} é triangular superior e \mathbf{P} é uma matriz de permutação. No texto a seguir não usaremos a matriz \mathbf{P} , a não ser em casos relevantes. Estes fatores surgem por simples aplicações de transformações lineares pela esquerda. Na construção desses fatores, ao trabalhar-

se com matrizes esparsas, eles podem se tornar menos esparsos do que \mathbf{A} , o que torna este processo computacionalmente muito caro.

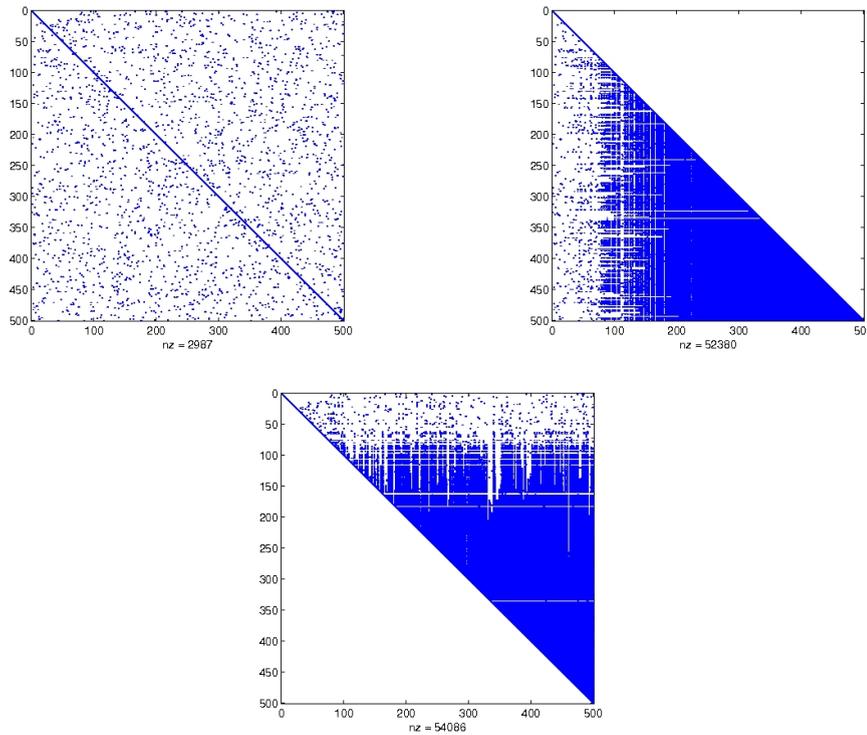


Figura 2: Exemplo: Matriz original \mathbf{A} , \mathbf{L} e \mathbf{U} , respectivamente

Em muitos casos não é necessária a fatoração completa da matriz, apenas uma fatoração aproximada é suficiente; nesse quadro surgem as fatorações incompletas (GOLUB; LOAN, 1996; SAAD, 2003). Diz-se que uma fatoração é incompleta se durante o processo de fatoração alguns elementos não nulos dos fatores são anulados.

Mesmo assim, uma fatoração incompleta pode ser computacionalmente muito cara. Uma das utilizações dessa fatoração ocorre quando devem-se resolver muitos sistemas com a mesma matriz \mathbf{A} e vetor do lado direito variável, assim aplica-se os fatores a muitos sistemas com custo de apenas uma fatoração.

5.3.1 ILU(0), ILUT, MILU

Ao aproximar a fatoração de uma matriz na forma citada acima, isto é, $\mathbf{A} = \mathbf{L}\mathbf{U}$, utiliza-se como forma de construção da fatoração *ILU* (*Incomplete LU*) uma modificação na eliminação Gaussiana permitindo preenchimento em apenas um conjunto de

posições nos fatores LU .

Suas posições, em que são permitidos preenchimentos são dadas em função de um conjunto de índices \mathbf{S} predefinido:

$$\begin{aligned} l_{ij} &= 0 & \text{se } j > i & \text{ ou } (i, j) \notin \mathbf{S} \\ u_{ij} &= 0 & \text{se } i > j & \text{ ou } (i, j) \notin \mathbf{S}. \end{aligned}$$

Ao construir uma fatoração LU incompleta, estamos construindo fatores $\tilde{\mathbf{L}}$ e $\tilde{\mathbf{U}}$ que aproximam os fatores \mathbf{L} e \mathbf{U} , respectivamente, ou equivalentemente $\mathbf{A} = \mathbf{LU} - \mathbf{R}$ onde \mathbf{R} é o resíduo ou erro da fatoração. Esta fatoração incompleta de \mathbf{A} representa uma fatoração completa de $\tilde{\mathbf{A}}$, quer dizer, $\tilde{\mathbf{A}} = \tilde{\mathbf{L}}\tilde{\mathbf{U}}$ onde o preconditionador \mathbf{M} é dado por $\mathbf{M} = \tilde{\mathbf{L}}\tilde{\mathbf{U}}$. De forma que não reste dúvidas, os fatores \mathbf{L} e \mathbf{U} representarão os fatores aproximados sempre que necessário.

Para que \mathbf{M} seja um bom preconditionador, é necessário que se aproxime de \mathbf{A} através de alguma medida. Uma estratégia típica é que seja exigido que as entradas \mathbf{M} coincidam com as de \mathbf{A} no conjunto \mathbf{S} :

$$m_{ij} = a_{ij} \quad \text{se } (i, j) \in \mathbf{S}. \quad (5.5)$$

Uma estratégia comumente utilizada para definir o conjunto \mathbf{S} é:

$$\mathbf{S} = \{(i, j) | a_{ij} \neq 0\}.$$

Isto quer dizer que os únicos elementos diferentes de zero permitidos nos fatores LU são aqueles que correspondem aos elementos diferentes de zero de \mathbf{A} . Este tipo de fatoração incompleta é chamada de ILU(0): fatoração LU incompleta de nível zero. Esta fatoração é simples e tem baixo custo computacional, mas pode acarretar, em muitos casos, em baixa efetividade na aceleração da convergência dos métodos iterativos.

Para superar este problema, diversas outras opções de fatorações incompletas foram desenvolvidas para permitir maior quantidade de preenchimento e assim prover uma melhoria na aproximação de forma a tornar o preconditionador mais eficiente, porém tornando o processo de fatoração mais custoso computacionalmente.

O conjunto \mathbf{S} apresentado anteriormente representa uma alternativa pertencente a um padrão estático, onde as posições que serão anuladas nos fatores são previamente definidas. Neste caso, é muito importante que não anule-se nenhum elemento da diagonal para não tornar a fatoração instável.

Fatorações que se baseiam na estrutura da matriz perdem muita informação

ao não considerar os valores numéricos dos elementos. Existem alguns métodos disponíveis que descartam elementos baseados em suas magnitudes. Este tipo de decomposição incompleta baseia-se em um padrão dinâmico.

A fatoração *LU incompleta com limitante (threshold)* ou *ILUT* baseia-se em um padrão dinâmico cuja característica é anular dos fatores os elementos que, em módulo, são menores que um determinado parâmetro limite τ pré-estabelecido, obtendo uma melhor aproximação com custo computacional mais elevado. Este critério pode funcionar mal caso a matriz apresente problemas de escalabilidade, neste caso é melhor utilizar um parâmetro τ relativo. Por exemplo, quando for eliminar a linha i , um novo preenchimento é aceitável somente se, em valor absoluto, é maior do que $\tau \|\mathbf{a}_{i*}\|_2$, onde \mathbf{a}_{i*} denota a i -ésima linha de \mathbf{A} . Outros critérios podem ser utilizados. O grande problema é conseguir determinar o valor ideal para parâmetro τ pois varia de um problema para outro, em nossos experimentos que serão reportados no Capítulo 6 utilizamos valores entre 10^{-4} e 10^{-2} .

Também é possível enfrentar problemas de armazenamento quando não podemos prever o quanto pode ser preenchido por este critério. Pode-se então utilizar uma estratégia de *limitante duplo* (SAAD, 2003): fixa-se uma tolerância τ e um valor inteiro positivo p que representa quantas entradas não nulas são permitidas em cada linha, que serão aquelas de maior magnitude, dos fatores \mathbf{L} e \mathbf{U} .

Em todas as técnicas apresentadas, os elementos que devem ser retirados dos fatores são simplesmente descartados. Porém existem técnicas que tentam compensar de alguma forma os elementos eliminados. Como por exemplo, uma estratégia popular é retirar de cada elemento da diagonal de \mathbf{U} a soma dos elementos de cada linha respectivamente. Esta técnica que visa uma compensação diagonal origina o chamado *Modified ILU* ou *MILU*. De forma mais simples, pode-se dizer que a condição (5.5) é removida para $m_{ii} = a_{ii}$ e uma nova condição para a soma das linhas é adicionada. Assim pode-se substituir (5.5) por

$$\sum_{j=1}^n m_{ij} = \sum_{j=1}^n a_{ij} \quad \forall i \quad \text{e} \quad m_{ij} = a_{ij} \quad \text{se} \quad i \neq j \quad \text{e} \quad (i, j) \in \mathbf{S}. \quad (5.6)$$

Denotemos a i -ésima linha de \mathbf{R} por \mathbf{r}_{i*} . Assim, a modificação proposta apresenta

$$u_{ii} := u_{ii} - (\mathbf{r}_{i*} \mathbf{e})$$

onde $\mathbf{e} \equiv (1, 1, \dots, 1)^T$. Note que $\mathbf{r}_{i*} \mathbf{e}$ é a soma dos elementos da linha \mathbf{r}_{i*} . A expressão

acima pode ser reescrita como $\mathbf{u}_{i*} := \mathbf{u}_{i*} - (\mathbf{r}_{i*} \mathbf{e}) \mathbf{e}_i^T$ e assim pode-se escrever

$$\mathbf{a}_{i*} = \sum_{k=1}^i l_{ik} \mathbf{u}_{k*} + (\mathbf{r}_{i*} \mathbf{e}) \mathbf{e}_i^T - \mathbf{r}_{i*}$$

Se multiplicarmos à esquerda por \mathbf{e} , obteremos

$$\mathbf{a}_{i*} \mathbf{e} = \sum_{k=1}^{i-1} l_{ik} \mathbf{u}_{k*} \mathbf{e} = \mathbf{L} \mathbf{U} \mathbf{e}.$$

Isto garante que a soma das linhas de \mathbf{A} é igual a soma das linhas de \mathbf{LU} . Este processo pode ser utilizado para qualquer fatoração \mathbf{LU} incompleta.

A abordagem de aproximação lumped de \mathbf{A} pode ser vista como um caso particular do MILU, onde todos os elementos fora da diagonal são anulados e compensados na diagonal.

5.3.2 ICC

Matrizes simétricas e positivas definidas (**spd**) podem ser decompostas em fatores triangulares mais rapidamente do que matrizes sem essas propriedades. O algoritmo padrão para esta operação é a *fatoração de Cholesky*, que é uma variante da eliminação gaussiana que opera em ambos os lados, esquerdo e direito, da matriz de uma vez, preservando e explorando sua simetria.

Relembrando, uma matriz $\mathbf{A} \in \mathbb{R}^{n \times n}$ é simétrica se $\mathbf{A} = \mathbf{A}^T$, e é positiva definida se $\mathbf{x}^T \mathbf{A} \mathbf{x} > 0$.

Suponha que \mathbf{A} seja uma matriz **spd** e $\mathbf{X} \in \mathbb{R}^{m \times n}$ de posto completo com $m \geq n$, então a matriz $\mathbf{X}^T \mathbf{A} \mathbf{X}$ também é simétrica e positiva definida. É simétrica porque $(\mathbf{X}^T \mathbf{A} \mathbf{X})^T = \mathbf{X}^T \mathbf{A}^T \mathbf{X} = \mathbf{X}^T \mathbf{A} \mathbf{X}$, e positiva definida porque, para qualquer vetor $\mathbf{x} \neq 0$, temos $\mathbf{X} \mathbf{x} \neq 0$ e então $\mathbf{x}^T (\mathbf{X}^T \mathbf{A} \mathbf{X}) \mathbf{x} = (\mathbf{X} \mathbf{x})^T \mathbf{A} (\mathbf{X} \mathbf{x}) > 0$.

Além disso, todos os autovalores de uma matriz simétrica positiva definida são reais e positivos pois, se $\mathbf{A} \mathbf{x} = \lambda \mathbf{x}$ para $\mathbf{x} \neq 0$, então $\mathbf{x}^T \mathbf{A} \mathbf{x} = \lambda \mathbf{x}^T \mathbf{x} > 0$ de onde pode-se concluir que $\lambda > 0$. Por outro lado, se uma matriz simétrica e tem todos os autovalores positivos, podemos concluir que ela é positiva definida.

Ao aplicar uma simples transformação linear pela esquerda da matriz a ser fatorada \mathbf{A} , visamos anular todos os elementos abaixo da primeira entrada da diagonal. Pela simetria da matriz, este mesmo procedimento pode ser feito pela direita para anular todos os elementos a direita da primeira entrada da diagonal, onde a matriz que representa a transformação linear pela direita é a transposta daquela aplica a esquerda. Ao proceder desta forma para o restante dos elementos, estamos transfor-

mando a matriz \mathbf{A} em uma matriz identidade e obtemos

$$\mathbf{A} = \underbrace{\mathbf{C}_1^T \mathbf{C}_2^T \dots \mathbf{C}_m^T}_{\mathbf{C}^T} \underbrace{\mathbf{C}_m \dots \mathbf{C}_2 \mathbf{C}_1}_{\mathbf{C}},$$

isto é

$$\mathbf{A} = \mathbf{C}^T \mathbf{C}.$$

Uma redução deste tipo é conhecida como *fatoração de Cholesky*.

Quando da implementação desta fatoração, somente metade da matriz a ser fatorada precisa ser representada explicitamente e assim somente metade das operações precisa ser feita. Ao final da fatoração só é necessário armazenar um fator, já que o outro é facilmente calculado. Esta fatoração, como era de se esperar, é realizada com apenas metade das operações necessárias para a decomposição LU . E sua forma incompleta é construída da mesma forma que a apresentada para a decomposição LU , passando a se chamar *decomposição de Cholesky incompleta*, representada por ICC .

5.4 Ordenamentos

Como foi dito anteriormente, o processo de decomposição, além de computacionalmente caro, pode gerar fatores menos esparsos que a matriz original, e para não perder a esparsidade existem várias técnicas de fatoração incompleta que podem ser utilizados para este fim. No entanto, decomposições incompletas geram fatores aproximados, devido ao descarte de entradas, que podem ter baixa qualidade, então apresentaremos nesta seção um meio que, dentre outras funções, pode minimizar tal problema.

O *ordenamento* ou *reordenamento* das matrizes pode ser, em alguns casos, um procedimento muito simples e computacionalmente barato, que visa diminuir os preenchimentos dos fatores através de matrizes de permutação, apesar de ser, em geral, um problema difícil (YANNAKAKIS, 1981; BILLIONNET, 1986).

Permutar linhas ou colunas das matrizes constitui uma simples operação, do ponto de vista matemático e computacional, porém uma importantíssima operação para o uso de implementações em paralelo bem como para técnicas de resolução de sistemas lineares, tanto diretas quanto iterativas. Além disso, apresentam significativo ganho quando utilizadas para garantir a esparsidade dos fatores de uma decomposição, como já mencionado.

Definição 5.4 (SAAD, 2003) *Seja \mathbf{A} uma matriz e $\pi = \{i_1, i_2, \dots, i_n\}$ uma permutação*

de um conjunto $\{1, 2, \dots, n\}$. Então as matrizes

$$\begin{aligned}\mathbf{A}_{\pi,*} &= \{a_{\pi(i),j}\}_{i=1,\dots,n;j=1,\dots,m}, \\ \mathbf{A}_{*,\pi} &= \{a_{i,\pi(j)}\}_{i=1,\dots,n;j=1,\dots,m}\end{aligned}$$

são chamadas *permutação- π linha* e *permutação- π coluna* de \mathbf{A} , respectivamente.

Uma matriz de *permutação simples* é a matriz identidade com duas de suas linhas permutadas. Denotemos estas matrizes por $\mathbf{X}_{i,j}$ sendo i e j o número das linhas trocadas. Para que troquemos linhas de uma matriz \mathbf{A} basta que a multipliquemos pela esquerda pela matriz $\mathbf{X}_{i,j}$. Seja π uma permutação arbitrária. Esta permutação é o produto de uma seqüência de n trocas consecutivas $\sigma(i_k, j_k), k = 1, \dots, n$. Cada uma destas trocas é feita através de uma multiplicação à esquerda por \mathbf{X}_{i_k, j_k} . O mesmo pode ser feito pra trocas de coluna ao multiplicar a matriz \mathbf{A} pela direita por \mathbf{X}_{i_k, j_k} . Claramente pode-se ver que $\mathbf{X}_{i,j}^2 = \mathbf{I}$ que é equivalente a $\mathbf{X}_{i,j} \mathbf{X}_{i,j}^{-1}$.

A seguinte proposição apresenta estes fatos de forma mais clara

Proposição 5.1 (SAAD, 2003) *Seja π uma permutação resultante de um produto de permutação simples $\sigma(i_k, j_k), k = 1, \dots, n$. Então*

$$\mathbf{A}_{\pi,*} = \mathbf{P}_\pi \mathbf{A}, \quad \mathbf{A}_{*,\pi} = \mathbf{A} \mathbf{Q}_\pi,$$

onde

$$\begin{aligned}\mathbf{P}_\pi &= \mathbf{X}_{i_n, j_n} \mathbf{X}_{i_{n-1}, j_{n-1}} \cdots \mathbf{X}_{i_1, j_1}, \\ \mathbf{Q}_\pi &= \mathbf{X}_{i_1, j_1} \mathbf{X}_{i_2, j_2} \cdots \mathbf{X}_{i_n, j_n}.\end{aligned}$$

Produtos de permutação simples de linhas ou colunas de matrizes são chamados de *matrizes permutação*. É importante observar que \mathbf{P}_π e \mathbf{Q}_π são inversas uma da outra, e utilizando a mesma permutação para permutar linhas e colunas é equivalente a fazer uma transformação similar, que propriedades importantes da matriz original como, por exemplo, seus autovalores. Desde que as matrizes elementares \mathbf{X}_{i_k, j_k} são simétricas, pode-se concluir que

$$\mathbf{Q}_\pi = \mathbf{P}_\pi^T = \mathbf{P}_\pi^{-1}.$$

No contexto de resolução de sistemas lineares, quando as linhas de uma matriz são permutadas, a ordem em que as equações são escritas é alterada. Enquanto que quando as colunas são permutadas, as incógnitas são reordenadas.

Tais sistemas, quando obtidos de EDPs muitas vezes podem gerar matrizes com diagonal dominância. Com o intuito de manter esta característica é comum

utilizarem-se permutações simétricas para preservar os elementos da diagonal, da seguinte maneira

$$\mathbf{A}_{\pi,\pi} = \mathbf{P}_{\pi}^T \mathbf{A} \mathbf{P}_{\pi}.$$

5.4.1 Grau Mínimo

Um dos algoritmos mais eficientes e utilizados, para a diminuição de preenchimento nas fatorações LU , é o *algoritmo do grau mínimo*, que apesar de suas várias versões não teve seu nome muito alterado. Este algoritmo não utiliza os valores dos elementos da matriz como critério, levando em consideração apenas sua estrutura, pois ao considerar a matriz diagonal dominante devido à sua origem e fazer sempre uma permutação simétrica não temos problemas com instabilidade numérica na eliminação gaussiana assim, podemos descrevê-lo de maneira geral como segue.

Ao trabalhar somente com a estrutura de \mathbf{A} , é simulado de alguma forma, os n passos da eliminação simétrica gaussiana. A cada passo, uma linha e sua coluna correspondente, da submatriz ainda por ser fatorada, são deslocadas de sua posição original de forma que o número de elementos não nulos da linha ou coluna pivô seja mínimo. Após os n passos desta simulação o novo posicionamento dos pivôs determina o novo ordenamento.

Diversas vezes, o processo de determinar qual linha deve ser movida de forma a minimizar o número de elementos não nulos encontra empates segundo o critério original. Então, algum tipo de critério de desempate deve ser utilizado, tornando-se fundamental para garantir qualidade à este ordenamento. Diversas técnicas para melhoria deste processo bem como critérios de desempate foram desenvolvidos e uma grande variedade pode ser encontrada, juntamente com exemplos numéricos, em (GEORGE; LIU, 1989).

A forma como a simulação da eliminação gaussiana é feita, é através da teoria de grafos (BONDY; MURTY, 1976). Sendo assim, considere $\mathcal{G}(\mathbf{A})$, ou simplesmente \mathcal{G} quando não restar dúvidas, um grafo não direcionado, pois este representa a estrutura de uma matriz simétrica por estrutura, considere também v como um nó de \mathcal{G} . Denotaremos por $\text{Adj}_{\mathcal{G}}(v)$ o conjunto de nós adjacentes ao nó v , e o grau de v por $\text{grau}_{\mathcal{G}}(v)$ que representa o número de lados conectados ao nó v .

Quando permutamos as linhas e colunas de \mathbf{A} usando uma matriz permutação \mathbf{P} geramos uma matriz \mathbf{PAP}^T , cujo grafo é $\mathcal{G}(\mathbf{PAP}^T)$. Este grafo tem estrutura idêntica à do grafo $\mathcal{G}(\mathbf{A})$ porém com os nós renomeados ou reordenados, de acordo com a matriz permutação \mathbf{P} .

Após o primeiro passo da eliminação gaussiana, onde sem perda de genera-

lidade foi eliminado o nó v , surge um novo grafo \mathcal{G}_v que é obtido pela eliminação do nó v bem como os lados conectados a ele e, adicionando novos lados que passarão a conectar os nós adjacentes a v , isto é, todo o conjunto $\text{Adj}_{\mathcal{G}}(v)$. O grau de um nó pode variar ao longo do processo de eliminação pois se v é adjacente a y então

$$\text{grau}_{\mathcal{G}_y}(v) \geq \text{grau}_{\mathcal{G}}(v) - 1 \geq \text{grau}_{\mathcal{G}}(y) - 1.$$

Assim elimina-se apenas um nó de cada vez, o que pode tornar o algoritmo lento. Formas de melhoria deste algoritmo são também apresentadas na referência (GEORGE; LIU, 1989) já citada anteriormente.

Este é um procedimento que reduz preenchimento de forma local, com isto o nó de grau mínimo não é necessariamente o que provoca o menor número de enchiamentos. Para melhorar este desenvolvimento é inevitável o aumento do custo computacional.

5.4.2 Cuthill-McKee Reverso

É fato conhecido que matrizes com banda ou envoltória pequenas mantêm esta propriedade ao se calcular seus fatores triangulares e, portanto, não sofrem muitos preenchimentos no processo de eliminação gaussiana. Reduzir a banda de uma matriz é um critério para redução de preenchimento de um ponto de vista global.

Originalmente, esta idéia surgiu em (CUTHILL; MCKEE, 1969) e ficou conhecido como *algoritmo de Cuthill-McKee (CM)*, cuja idéia é reduzir a banda de uma matriz de estrutura simétrica. Esta estratégia visa posicionar os elementos da matriz o mais próximo possível da diagonal, assim a largura de banda é definida como sendo o maior valor de $|i - j|$ para $a_{ij} \neq 0$.

Assim a idéia é, primeiro escolher um nó qualquer e nomea-lo 1. Depois, nomeia-se os nós adjacentes ao nó 1 em ordem crescente de seus graus. Este conjunto de nós é dito ser de primeiro nível. Agora, este procedimento deve continuar de acordo com a ordem em que foram nomeados os nós deste nível. Os novos nós são de segundo nível. Repete-se este procedimento até esgotarem-se os nós. Assim, esta forma de ordenamento gera uma estrutura de níveis.

Obviamente é possível fazer algumas melhorias como determinação do nó inicial segundo algum critério e a utilização de critérios de desempate, por exemplo. Um processo iterativo para melhorar este algoritmo utilizando características da estrutura de níveis é:

Procurar nós iniciais que gerem estruturas de níveis de largura pequenos, isto

é, pequena quantidade de nós por nível. Como observado em (GIBBS; JR.; STOCKMEYER, 1976), aumentando-se a profundidade, ou quantidade de níveis, da estrutura L, deve-se esperar uma diminuição da largura de seus níveis. Em (GEORGE; LIU, 1979), é proposto um algoritmo heurístico que procura determinar nós que gerem estrutura de níveis de profundidade elevada da seguinte maneira. Construa uma estrutura de níveis gerada por um nó r , escolha um nó de grau mínimo x do último nível da estrutura e gere uma nova estrutura a partir dele. Se sua estrutura é mais profunda que a gerada por r então repita o processo.

Apesar de o algoritmo CM ser eficiente, o de maior sucesso na redução do tamanho da envoltória de uma matriz esparsa é construído por uma pequena modificação do CM proposta por George, em sua tese de doutorado (GEORGE, 1971), e que consiste em inverter a numeração obtida pelo ordenamento CM, assim este algoritmo ficou conhecido como *algoritmo de Cuthill-McKee reverso* ou *CMR*.

Em termos matriciais temos:

$$\mathbf{A}_r = \mathbf{P}\mathbf{A}_c\mathbf{P}^T$$

onde

$$\mathbf{P} = \begin{bmatrix} & & & 1 \\ & & 1 & \\ & \dots & & \\ & 1 & & \\ 1 & & & \end{bmatrix} \quad \text{e} \quad \mathbf{P}^T = \mathbf{P},$$

onde \mathbf{A}_c representa a permutação da matriz \mathbf{A} pelo algoritmo CM e \mathbf{A}_r representa a permutação da matriz \mathbf{A} pelo algoritmo CMR.

Em (LIU; SHERMAN, 1976) é demonstrado que CMR necessita, no máximo, do mesmo número de operações e espaço de memória utilizado para realização da eliminação gaussiana.

5.5 Projeção

A utilização do método da projeção com aproximação agrupada para a matriz \mathbf{A} em (3.9), proposto na seção 3.1, nos fornece uma solução aproximada e de forma bem mais rápida que através da resolução do problema de ponto de sela $\mathcal{A}\mathbf{x} = \mathbf{b}$ (1.1) sem a fatoração LU em blocos.

No entanto, esta aproximação não é muito sofisticada e assim pode-se perder

em qualidade de solução. A utilização da solução do sistema através do método da projeção como estimativa inicial para a resolução do problema original $\mathcal{A}\mathbf{x} = \mathbf{b}$, nos conduz à solução exata mais rapidamente já que inicia-se o algoritmo com um ponto de partida mais próximo da solução.

Uma alternativa de melhoria, é a utilização do método da projeção como preconditionador. Neste caso, o preconditionamento é feito sem um preconditionador \mathbf{M} , mas sim com a utilização de uma forma de resolução aproximada cuja solução representa o efeito do uso do preconditionador. Esta forma de utilização do método da projeção deve ser aplicada ao ter que solucionar o sistema $\mathbf{M}\mathbf{z} = \mathbf{r}$ que surge ao utilizar o GMRes ou Bi-CGStab preconditionados.

Neste caso, teremos de resolver dois sistemas lineares, um para a matriz \mathcal{A} **spd**, onde utilizamos o gradiente conjugado com preconditionador Cholesky incompleto, e outro sistema linear para o complemento de Schur, onde utilizamos o GMRes (poderia ser o Bi-CGStab) preconditionado pela esquerda (poderia ser pela direita) com ILUT.

5.6 Diagonal em Blocos - MGW

Uma condição suficiente para que um preconditionador seja eficiente é que a matriz preconditionada $\mathbf{T} = \mathbf{M}^{-1}\mathbf{A}$ tenha polinômio mínimo de baixo grau, o que significa que \mathbf{T} tem poucos autovalores distintos.

Em (MURPHY; GOLUB; WATHEN, 2000) é proposto o seguinte preconditionador para o problema de ponto de sela apresentado

$$\mathcal{P} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}\mathbf{A}^{-1}\mathbf{G} \end{bmatrix},$$

que chamamos aqui, de MGW 1 (Murphy, Golub e Wathen). Assim, a matriz preconditionada $\mathbf{T} = \mathcal{P}^{-1}\mathcal{A}$ satisfaz $\mathbf{T}(\mathbf{T} - \mathbf{I})(\mathbf{T}^2 - \mathbf{T} - \mathbf{I}) = \mathbf{0}$, polinômio múltiplo do polinômio mínimo, o que implica que a matriz \mathbf{T} (não singular) tem três autovalores distintos: 1 , $\frac{1 \pm \sqrt{5}}{2}$.

Isto quer dizer que qualquer método iterativo, que atenda a condição de não simetria, baseado em subespaços de Krylov convergirá em no máximo 3 iterações, em aritmética exata, com o uso deste preconditionador para o problema proposto.

Além deste preconditionador, também foi proposto

$$\mathcal{P} = \begin{bmatrix} \mathbf{A} & \mathbf{G} \\ \mathbf{0} & \mathbf{D}\mathbf{A}^{-1}\mathbf{G} \end{bmatrix},$$

que chamamos de MGW 2. Onde a matriz preconditionada $\mathcal{P}^{-1}\mathcal{A}$ por ele tem apenas os autovalores ± 1 .

Para ambos os preconditionadores apresentados, é necessário resolver dois sistemas lineares, um para a matriz A e outro para o complemento de Schur, em que utilizamos, respectivamente, o CG preconditionado com ICC e o GMRes preconditionado à esquerda com ILUT.

Porém, problema que permanece é: como calcular $(\mathbf{DA}^{-1}\mathbf{G})^{-1}$ ou como resolver o sistema $\mathbf{DA}^{-1}\mathbf{G}\mathbf{z}_2 = \mathbf{r}_2$, já que não podemos inverter \mathbf{A} ?

Descreveremos agora algumas aproximações que para \mathbf{A}^{-1} e algumas formas que aproximam o complemento de Schur diretamente.

5.6.0.1 Diagonal

Analogamente ao preconditionador diagonal, monta-se a matriz diagonal que aproxima \mathbf{A} e a invertemos de forma trivial.

5.6.0.2 Lumped

Esta aproximação já foi descrita no capítulo 3 Método da projeção, e pode também ser utilizada aqui. Esta aproximação gera uma matriz diagonal cuja inversa não representa nenhum custo computacional significativo.

5.6.0.3 Matriz completa aproximada (GIRAUD, 2008)

Esta abordagem visa aproximar o complemento de Schur. Para isto tentaremos encontrar uma matriz $\mathbf{F} \in \mathbb{R}^{n \times m}$ que pode ser obtida do seguinte problema de múltiplos lados direitos

$$\mathbf{AF} = \mathbf{G}.$$

Este sistema de múltiplos lados direitos implica na resolução de m sistemas lineares. Utilizamos então, a fatoração de Cholesky de \mathbf{A} , já calculada para resolver o sistema $\mathbf{Az}_1 = \mathbf{r}_1$, que exige a resolução de dois sistemas triangulares.

A construção do complemento de Schur desta forma gera uma matriz densa, como pode ser visto na Figura 3, o que acarreta em uma redução na eficiência da resolução destes sistemas e aumento de gasto em armazenamento de memória. Para tentar diminuir o número de preenchimento do complemento de Schur, pode ser feita uma esparsificação, isto é, utilizar a idéia de redução de preenchimento utilizada nos métodos de fatoração incompleta.

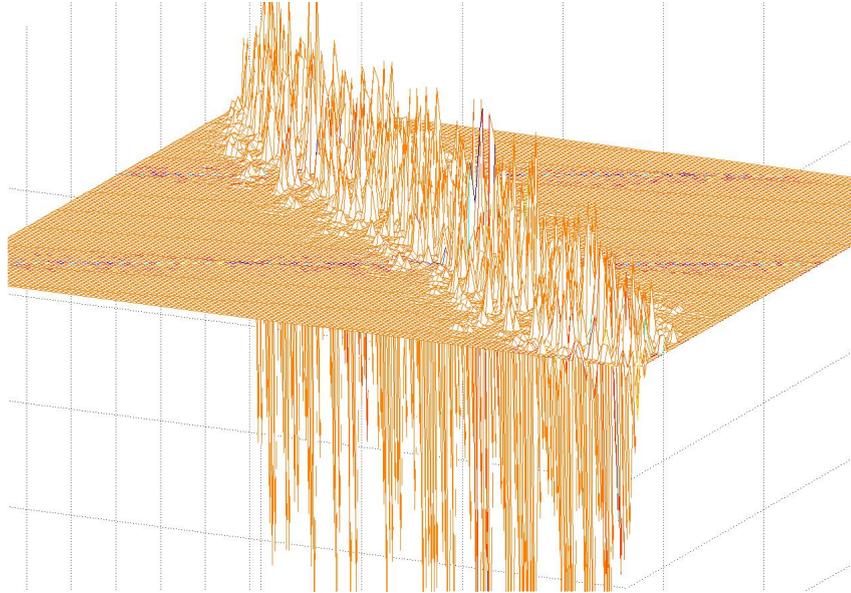


Figura 3: Exemplo: A matriz do complemento de Schur é densa.

Esta idéia também pode ser utilizada para o método da projeção.

5.6.0.4 CGLS

No Capítulo 4 iterações de Krylov foram apresentados os método CGLS para resolver problemas de mínimos quadrados, bem como a forma de utiliza-los para resolver o sistema apresentado para o complemento de Schur.

Para a resolução do complemento de Schur desta forma, é necessário, na relidade, resolver dois sistemas lineares: $\mathbf{D}\mathbf{D}^T\bar{\mathbf{z}} = \bar{\mathbf{b}}$ e $\mathbf{G}^T\mathbf{G}\mathbf{y}_2 = \bar{\bar{\mathbf{z}}}$ em são utilizados preconditionadores do tipo ICC para cada um deles.

5.6.0.5 Probe

Na seção 5.7 será apresentada uma técnica de sondagem que captura os valores da matriz a ser aproximada e então os distribui para gerar a aproximação.

5.7 Probing-Sondagem

Esta é uma técnica de aproximação da matriz do sistema, que pode não estar disponível explicitamente. Para construir esta aproximação são utilizados vetores de sondagem que capturam os elementos da matriz a ser aproximada. As informações adquiridas por estes vetores devem então ser reposicionadas para constituir a matriz aproxiada.

Assuma que \mathbf{A} é tridiagonal, isto é, os elementos fora das três diagonais principais são nulos

$$\mathbf{A} = \begin{pmatrix} a_1 & b_2 & & & & & \\ c_2 & a_2 & b_3 & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & c_{m-1} & a_{m-1} & b_m & & \\ & & & c_m & a_m & & \end{pmatrix}.$$

É simples recuperar \mathbf{A} utilizando-se três vetores bem escolhidos

$$\begin{aligned} \mathbf{w}_1 &= (1, 0, 0, 1, 0, 0, 1, \dots)^T, \\ \mathbf{w}_2 &= (0, 1, 0, 0, 1, 0, 0, \dots)^T, \\ \mathbf{w}_3 &= (0, 0, 1, 0, 0, 1, 0, \dots)^T. \end{aligned}$$

Então temos

$$\begin{aligned} \mathbf{A}\mathbf{w}_1 &= (a_1, c_2, b_4, a_4, c_5, b_7, \dots, b_{3i+1}, a_{3i+1}, c_{3i+2}, \dots)^T, \\ \mathbf{A}\mathbf{w}_2 &= (b_2, a_2, c_3, b_5, a_5, c_6, \dots, b_{3i+2}, a_{3i+2}, c_{3i+3}, \dots)^T, \\ \mathbf{A}\mathbf{w}_3 &= (0, b_3, a_3, c_4, b_6, a_6, c_7, \dots, b_{3i}, a_{3i}, c_{3i+1}, \dots)^T. \end{aligned}$$

Isto mostra que todos os coeficientes da matriz \mathbf{A} estão representados nesses três vetores. Assim facilmente podemos calcular $\mathbf{A}\mathbf{W}$, onde \mathbf{W} é uma matriz cujas colunas são compostas pelos vetores \mathbf{w}_i , e obter a própria matriz \mathbf{A} .

No entanto para matrizes não estruturadas pode haver perdas de elementos ou então soma de elementos que irão compor a nova matriz, que se tornará distinta da original. Para tentar melhorar a aproximação é necessário então aumentar o número de vetores de sonda, o que nem sempre é conveniente.

Ou então, podemos utilizar estratégias mais refinadas atuando diretamente sobre o grafo associado á matriz (SIEFERT; STURLER, 2006).

5.7.1 Matriz de Ponto de Sela

Em nosso caso, a matriz a ser aproximada é o complemento de Schur que tem a forma $\mathbf{S} = -\mathbf{D}\mathbf{A}^{-1}\mathbf{G}$ onde $\mathbf{D}, \mathbf{G}^T \in \mathbb{R}^{m \times n}$, $\mathbf{A} \in \mathbb{R}^{n \times n}$ e $n > m$.

Para aproximar \mathbf{S} é necessário obter o resultado da multiplicação $\mathbf{G}\mathbf{W}$ onde \mathbf{W} é a matriz formada pelos vetores de sonda. Como não é possível obter \mathbf{A}^{-1} , resolve-se o sistema linear $\mathbf{A}\mathbf{F} = \mathbf{G}\mathbf{W}$ utilizando PCG ou a os fatores \mathbf{L} e \mathbf{U} de \mathbf{A} , que já foram calculados anteriormente, e então resolve-se dois sistemas lineares triangulares. No entanto, este é um problema de múltiplos lados direitos que exige que tenhamos de resolver k sistemas lineares, um para cada vetor que compõe as colunas de $\mathbf{G}\mathbf{W}$,

isto é, k é o número de vetores de sonda escolhido. Assim, obtemos explicitamente, $\tilde{\mathbf{F}} \approx \mathbf{A}^{-1}\mathbf{G}\mathbf{W}$.

Faltando então, somente multiplicar por $-\mathbf{D}$ à esquerda para obter uma aproximação explícita para o complemento de Schur \mathbf{S} . O grande problema desta abordagem é que a matriz \mathbf{S} é densa, e mesmo que fosse feita uma esparsificação cuidadosa ela seria não estruturada, caso em que esta técnica não é capaz de tirar bom proveito.

5.8 PCG

Como visto na seção 4.2, o gradiente conjugado é um eficiente método que resolve sistemas lineares cuja matriz dos coeficientes é simétrica e positiva definida. Para preconditionar o gradiente conjugado, isto é, preconditionar o sistema que será resolvido utilizando-se o gradiente conjugado, precisamos manter sua matriz simétrica após a aplicação do preconditionador. Mesmo que \mathbf{M} seja simétrica, $\mathbf{M}^{-1}\mathbf{A}$ ou $\mathbf{A}\mathbf{M}^{-1}$ não são necessariamente simétricas.

Para contornar esta dificuldade, pode-se fatorar \mathbf{M} simétrica e positiva definida como $\mathbf{M} = \mathbf{E}\mathbf{E}^T$ (onde \mathbf{E} pode ser o fator de Cholesky por exemplo).

Assim transformarmos o sistema $\mathbf{A}\mathbf{x} = \mathbf{b}$ em

$$\mathbf{E}^{-1}\mathbf{A}\mathbf{E}^{-T}\hat{\mathbf{x}} = \mathbf{E}^{-1}\mathbf{b}, \quad \hat{\mathbf{x}} = \mathbf{E}^T\mathbf{x}, \quad (5.7)$$

onde resolve-se primeiro o sistema para \hat{x} e depois para x .

Como $\mathbf{M}^{-1} = \mathbf{E}^{-1}\mathbf{A}\mathbf{E}^{-T}$ é simétrica e positiva definida então pode-se utilizar o CG para resolver este sistema, porém agora iremos chamá-lo de *gradiente conjugado preconditionado* ou *PCG*. No entanto, não é necessário modificar todo algoritmo do gradiente conjugado para resolver o problema preconditionado. Toda transformação que simplifica a aplicação do preconditionador no algoritmo de gradiente conjugado pode ser encontrada em detalhes em (SHEWCHUK, 1994). Apresentamos abaixo o algoritmo do gradiente conjugado preconditionado onde o preconditionador somente precisa ser aplicado no resíduo onde, de forma equivalente, pode ser aplicado pela esquerda ou pela direita.

Algoritmo do PCG

1. Escolha $\mathbf{x}_{(0)}$, atribua $i = 0$ e calcule
 $\mathbf{r}_{(0)} = \mathbf{b} - \mathbf{A}\mathbf{x}_{(0)}$,
 $\mathbf{d}_{(0)} = \mathbf{M}^{-1}\mathbf{r}_{(0)}$ e
 $res_{(0)} = \mathbf{r}_{(0)}^T \mathbf{d}_{(0)}$.
 2. *While* ($i < i_{max}$) e ($res_{(i)} > tol$) *Do*
 3. $\mathbf{q}_{(i)} = \mathbf{A}\mathbf{d}_{(i)}$;
 $res = \frac{res_{(i)}}{\mathbf{d}_{(i)}^T \mathbf{q}_{(i)}}$;
 4. $\alpha_{(i)} = \frac{res_{(i)}}{\mathbf{d}_{(i)}^T \mathbf{q}_{(i)}}$;
 5. $\mathbf{x}_{(i+1)} = \mathbf{x}_{(i)} + \alpha_{(i)} \mathbf{d}_{(i)}$;
 6. $\mathbf{r}_{(i+1)} = \mathbf{r}_{(i)} - \alpha_{(i)} \mathbf{q}_{(i)}$;
 7. $\mathbf{s}_{(i+1)} = \mathbf{M}^{-1}\mathbf{r}_{(i+1)}$;
 8. $res_{(i+1)} = \mathbf{r}_{(i+1)}^T \mathbf{s}_{(i+1)}$;
 9. $\beta = \frac{res_{(i+1)}}{res_{(i)}}$;
 10. $\mathbf{d}_{(i+1)} = \mathbf{s}_{(i+1)} + \beta \mathbf{d}_{(i)}$;
 11. $i \leftarrow i + 1$;
 12. *End.*
-

Note que, apenas é necessário que seja resolvido um sistema $\mathbf{M}^{-1}\mathbf{x} = \mathbf{r}$, para o resíduo \mathbf{r} como lado direito, a cada iteração. Aliás, é possível armazenar o vetor obtido de $\mathbf{s}_{(i)} = \mathbf{M}^{-1}\mathbf{r}_{(i)}$ pois é utilizado várias vezes. Assim, é necessário resolver apenas um sistema linear para o preconditionador por iteração, garantindo uma melhoria significativa em quantidade de operação e conseqüentemente, em tempo.

É claro que, se tiver sido necessário fatorar \mathbf{M} , como no caso de utilizar-se a fatoração de Cholesky, então utiliza-se os fatores triangulares \mathbf{E} e \mathbf{E}^T resolvendo dois sistemas retangulares, para resolver este novo sistema.

Nem sempre é necessário calcular \mathbf{M} ou \mathbf{M}^{-1} explicitamente, somente é necessário que se calcule o efeito da aplicação de \mathbf{M}^{-1} em um vetor. Isto quer dizer que não é preciso que se tenha um preconditionador, é possível ter uma forma ou método de preconditionamento.

5.9 GMRes Precondicionado

No caso de preconditionar o GMRes, podem ser utilizadas as mesmas três opções de aplicação de preconditionamento que estão disponíveis para o gradiente conjugado (preconditionamento pela esquerda, direita e split). No entanto, para o GMRes a forma de preconditionamento modifica a forma de medida do resíduo proporcionando alguma diferença na velocidade de convergência do algoritmo.

5.9.1 Precondicionamento pela Esquerda

Exatamente como apresentado anteriormente, o precondicionamento do GMRes pela esquerda é feito através da seguinte alteração do sistema

$$\mathbf{M}^{-1}\mathbf{A}\mathbf{x} = \mathbf{M}^{-1}\mathbf{b}$$

o que implica na seguinte alteração no algoritmo

Algoritmo do GMRes precondicionado pela esquerda

1. Escolha $\mathbf{x}_{(0)}$ e calcule $\mathbf{r}_{(0)} = \mathbf{M}^{-1}(\mathbf{b} - \mathbf{A}\mathbf{x}_{(0)})$ e $\mathbf{v}_{(1)} = \frac{\mathbf{r}_{(0)}}{\|\mathbf{r}_{(0)}\|}$.
 2. *for* $j = 1, \dots, m$
 3. $\hat{\mathbf{v}}_{(j)} = \mathbf{M}^{-1}\mathbf{A}\mathbf{v}_{(j)}$;
 4. $h_{ij} = \langle \hat{\mathbf{v}}_{(j)}, \mathbf{v}_{(i)} \rangle, i = 1, \dots, j$;
 5. $\hat{\mathbf{v}}_{(j+1)} = \mathbf{A}\mathbf{v}_{(j)} - \sum_{i=1}^j h_{i,j} \mathbf{v}_{(i)}$;
 6. $h_{j+1,j} = \|\hat{\mathbf{v}}_{(j+1)}\|$; e
 7. $\mathbf{v}_{(j+1)} = \frac{\hat{\mathbf{v}}_{(j+1)}}{h_{j+1,j}}$;
 8. *end*.
 9. Forme a solução: $\mathbf{x}_{(m)} = \mathbf{x}_{(0)} + \mathbf{V}_m \mathbf{y}_m$ onde \mathbf{y}_m minimiza $J(\mathbf{y}_m) = \|\beta \mathbf{e}_1 - \mathbf{H}_m \mathbf{y}_m\|$, onde $\beta = \|\mathbf{r}_{(0)}\|$.
-

Todos os resíduos e normas dos resíduos que são calculadas neste algoritmo correspondem ao resíduo precondicionado $\mathbf{M}^{-1}(\mathbf{b} - \mathbf{A}\mathbf{x}_{(m)})$, ao invés do resíduo original (não precondicionado) $\mathbf{b} - \mathbf{A}\mathbf{x}_{(m)}$. Não há como medir o resíduo não precondicionado a menos que o calcule explicitamente através de uma multiplicação por \mathbf{M} . O que proporcina alguma modificação nos critérios de parada.

5.9.2 Precondicionamento pela Direita

Como também já foi visto, o precondicionamento do algoritmo do GMRes pela direita é baseado na seguinte equação

$$\mathbf{A}\mathbf{M}^{-1}\mathbf{u} = \mathbf{b}, \text{ com } \mathbf{u} = \mathbf{M}\mathbf{x}.$$

No entanto a variável \mathbf{u} nunca é utilizada explicitamente. E o resíduo sempre pode ser calculado através da utilização do vetor \mathbf{x} em $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x} = \mathbf{b} - \mathbf{A}\mathbf{M}^{-1}\mathbf{u}$. No final o vetor \mathbf{u} que aproxima a solução tem a seguinte forma

$$\mathbf{u}_{(m)} = \mathbf{u}_{(0)} + \sum_{i=1}^m v_i \eta_i,$$

com $\mathbf{u}_{(0)} = \mathbf{M}\mathbf{x}_{(0)}$.

Em termos da variável \mathbf{x} temos

$$\mathbf{x}_{(m)} = \mathbf{x}_{(0)} + \mathbf{M}^{-1} \sum_{i=1}^m v_i \boldsymbol{\eta}_i.$$

Algoritmo do GMRes preconditionado pela direita

1. Escolha $\mathbf{x}_{(0)}$ e calcule $\mathbf{r}_{(0)} = \mathbf{b} - \mathbf{A}\mathbf{x}_{(0)}$ e $\mathbf{v}_{(1)} = \frac{\mathbf{r}_{(0)}}{\|\mathbf{r}_{(0)}\|}$.
 2. *for* $j = 1, \dots, m$
 3. $\hat{\mathbf{v}}_{(j)} = \mathbf{A}\mathbf{M}^{-1}\mathbf{v}_{(j)}$;
 4. $h_{ij} = \langle \mathbf{A}\mathbf{v}_{(j)}, \mathbf{v}_{(i)} \rangle, i = 1, \dots, j$;
 5. $\hat{\mathbf{v}}_{(j+1)} = \mathbf{A}\mathbf{v}_{(j)} - \sum_{i=1}^j h_{i,j} \mathbf{v}_{(i)}$;
 6. $h_{j+1,j} = \|\hat{\mathbf{v}}_{(j+1)}\|$; e
 7. $\mathbf{v}_{(j+1)} = \frac{\hat{\mathbf{v}}_{(j+1)}}{h_{j+1,j}}$;
 8. *end*.
 9. Forme a solução: $\mathbf{x}_{(m)} = \mathbf{x}_{(0)} + \mathbf{M}^{-1}\mathbf{V}_m \mathbf{y}_m$ onde \mathbf{y}_m minimiza $J(\mathbf{y}_{(i)}) = \|\beta \mathbf{e}_1 - \mathbf{H}_m \mathbf{y}_m\|$, onde $\beta = \|\mathbf{r}_{(0)}\|$.
-

Assim, a única modificação é o momento em que o preconditionamento é feito, atentando, é claro, para a possibilidade de um novo critério de parada.

Neste caso, o subespaço de Krylov gerado é

$$\mathcal{K}^m(\mathbf{A}\mathbf{M}^{-1}; \mathbf{r}_{(0)}) = \text{span}\{\mathbf{r}_{(0)}, \mathbf{A}\mathbf{M}^{-1}\mathbf{r}_{(0)}, \dots, (\mathbf{A}\mathbf{M}^{-1})^{m-1}\mathbf{r}_{(0)}\}.$$

5.10 Bi-CGStab Precondicionado

O algoritmo do Bi-CGStab preconditionado para resolver sistemas lineares $\mathbf{A}\mathbf{x} = \mathbf{b}$, com preconditionador \mathbf{M} , pode ser visto a seguir. A matriz \mathbf{M} representa a matriz de preconditionamento e a forma de preconditionamento.

Algoritmo do Bi-CGStab preconditionado

1. Escolha $\mathbf{x}_{(0)}$. Calcule $\mathbf{r}_{(0)} = \mathbf{b} - \mathbf{A}\mathbf{x}_{(0)}$
 2. Escolha $\hat{\mathbf{r}}_{(0)}$ tal que $\langle \mathbf{r}_{(0)}, \hat{\mathbf{r}}_{(0)} \rangle \neq 0$, i.e., $\mathbf{r}_{(0)} = \hat{\mathbf{r}}_{(0)}$
 3. $\rho_{(0)} = \alpha = \omega_{(0)} = 1$
 4. $\mathbf{v}_{(0)} = \mathbf{p}_{(0)} = \mathbf{0}$
 5. Para $i = 1, 2, 3 \dots$ **Do**
 6. $\rho_{(i)} = \langle \mathbf{r}_{(0)}, \hat{\mathbf{r}}_{(0)} \rangle; \beta = \left(\frac{\rho_{(i)}}{\rho_{(i-1)}} \right) \left(\frac{\alpha}{\omega_{(i-1)}} \right)$
 7. $\mathbf{p}_{(i)} = \mathbf{r}_{(i-1)} + \beta(\mathbf{p}_{(i-1)} - \omega_{(i-1)}\mathbf{v}_{(i-1)})$
 8. $\hat{\mathbf{p}}_{(i)} = \mathbf{M}^{-1}\mathbf{p}_{(i)}$
 9. $\mathbf{v}_{(i)} = \mathbf{A}\hat{\mathbf{p}}_{(i)}$
 10. $\alpha = \frac{\rho_{(i)}}{(\hat{\mathbf{r}}_{(0)}, \mathbf{v}_{(i)})}$
 11. $\mathbf{s} = \mathbf{r}_{(i-1)} - \alpha\mathbf{v}_{(i)}$
 12. $\hat{\mathbf{s}} = \mathbf{M}^{-1}\mathbf{s}$
 13. $\mathbf{t} = \mathbf{A}\hat{\mathbf{s}}$
 14. $\omega_{(i)} = \frac{\langle \mathbf{t}, \mathbf{s} \rangle}{\langle \mathbf{t}, \mathbf{t} \rangle}$
 15. $\mathbf{x}_{(i)} = \mathbf{x}_{(i-1)} + \alpha\hat{\mathbf{p}}_{(i)} + \omega_{(i)}\hat{\mathbf{s}}$
 16. se $\mathbf{x}_{(i)}$ "is accurate enough" **EndDo**
 17. $\mathbf{r}_{(i)} = \mathbf{s} - \omega_{(i)}\mathbf{t}$
 18. Fim
-

6 RESULTADOS NUMÉRICOS

Neste capítulo são apresentados os resultados numéricos a partir de um programa gerado para a solucionar os sistemas de equações lineares de ponto de sela não simétricos, tanto em sua forma acoplada quanto segregada. Estes sistemas lineares foram gerados pela simulação de escoamento de fluidos incompressíveis. O programa utilizado para a geração do código foi o Matlab[®] versão 7 (2007 The MathWorks, Inc.) utilizando todos os solucionadores, condicionadores e reordenamentos apresentados. A configuração do computador utilizado é Intel Xeon 1.60GHz, cache 4096 KB, 4 núcleos e 4GB de RAM.

Serão apresentados nas seguintes seções, os resultados comparativos de eficiência para simulações de três dimensões de um canal (Couette), um canal com degrau e um compartimento de reservatório (Caledônia), em que foi utilizada uma descrição semi-Lagrangeana para um fluido monofásico. Para processo de medida do tempo de resolução, optou-se por fazer a medida da resolução de apenas um sistema linear.

Com o intuito de medir o erro dos métodos utilizados, para os tamanhos de malha escolhidos foi possível resolver os sistemas através da utilização de um método direto. No entanto, devido aos erros gerados pelo uso de aritmética finita, mesmo os métodos diretos apresentam erro significativo. Para contornar este fato utilizamos a resolução do método direto de forma iterativa como apresentado no Capítulo 4 pelo algoritmo

Algoritmo do refinamento iterativo

1. Calcular $\mathbf{r}_{(i)} = \mathbf{b} - \mathbf{A}\mathbf{x}_{(i)}$
 2. Obter $\mathbf{z}_{(i)}$ de $\mathbf{A}\mathbf{z}_{(i)} = \mathbf{r}_{(i)}$
 3. Calcular $\mathbf{x}_{(i)} = \mathbf{x}_{(i-1)} + \mathbf{z}_{(i-1)}$
 4. Caso não tenha atingido a precisão necessária, volte ao passo 1.
-

onde a matriz \mathbf{A} que aparece no segundo passo era a matriz \mathbf{M} do algoritmo apresentado no Capítulo 4. Esta forma iterativa é conhecida como *refinamento iterativo* onde foi utilizado como critério de parada o *epsilon de máquina* como tolerância para o resíduo.

Mesmo para os casos utilizados, para não sobrecarregar a memória, foi necessário resolver estes sistemas de forma segregada, utilizando o método da projeção baseada em decomposição *LU* utilizando métodos diretos, como apresentado no Capítulo 3, esta forma de resolução é conhecida como *método de Uzawa*.

De maneira geral, o método da projeção com as aproximações sugeridas forneceram uma solução de forma bem rápida no entanto, com precisão insatisfatória,

como será visto adiante. Este fato mostrou que devido às aproximações escolhidas este método não seria capaz de atingir uma solução de qualidade, o que nos levou a pensar na utilização do refinamento iterativo para garantir qualidade à solução. Esta forma do método da projeção com refinamento iterativo chamamos de ABLUF (Approximate Block LU Factorization). Esta melhoria, de fato rendeu as modificações desejadas na qualidade de solução com uma demora maior na velocidade de solução cujas comparações serão apresentadas nas seções seguintes.

Em todos os testes em que utilizou-se o GMRes, não foi utilizado “restart”(reinício) para que seja possível medir o número exato de iterações para este método.

6.1 Canal 3D

Este exemplo consiste em simular o escoamento de um fluido através de um canal tridimensional. As condições de entrada e saída de fluido são aplicadas nas fronteiras laterais, enquanto que nas outras fronteiras são aplicadas velocidades nulas. Considerou-se para este exemplo, um canal de dimensões:

- $10 \times 5 \times 3$ que gerou uma matriz \mathcal{A} de ordem 1.620, onde 1.5% são elementos não nulos e existem 1.485 componentes de velocidade e 135 de pressão;
- $20 \times 10 \times 3$ que gerou uma matriz \mathcal{A} de ordem 7.790, onde 0.37% são elementos não nulos e existem 7.220 componentes de velocidade e 570 de pressão;
- $50 \times 20 \times 3$ que gerou uma matriz \mathcal{A} de ordem 42.630, onde 0.072% são elementos não nulos e existem 39.690 componentes de velocidade e 2.940 de pressão;

número de Reynolds $Re=10.000$ e CFL 1 e 5.

A geometria, dimensões e condições de entrada e saída podem ser vistos, a partir de um corte transversal, na Figura 4

6.2 Degrau 3D

Este teste consiste na expansão de um fluido sobre um degrau, gerando uma recirculação abaixo da camada limite do escoamento. A geometria, dimensões e condições de entrada e saída podem ser vistos, a partir de um corte transversal, na Figura 5

Neste caso, foram utilizados $L = 1$ e $U = 1$ como valores de referência para comprimentos e velocidade, sendo o tamanho do degrau igual a 1.

A simulação foi feita considerando-se $Re=10.000$, variando CFL, 1 e 5, e malha de dimensões

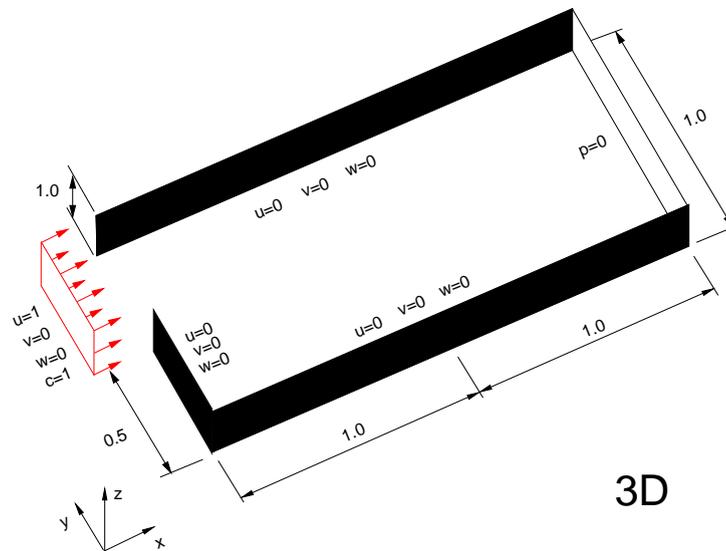


Figura 4: Domínio para o escoamento em um canal.

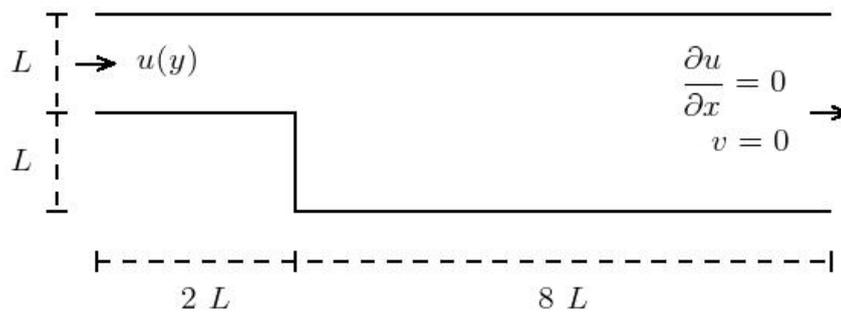


Figura 5: Domínio para o escoamento em um degrau.

- $10 \times 5 \times 3$ que gerou uma matriz \mathcal{A} de ordem 1.620, onde aproximadamente 1.5% são elementos não nulos e existem 1.485 componentes de velocidade e 135 de pressão;
- $20 \times 10 \times 3$ que gerou uma matriz \mathcal{A} de ordem 7.790, onde aproximadamente 0.37% são elementos não nulos e existem 7.220 componentes de velocidade e 570 de pressão;
- $50 \times 20 \times 3$ que gerou uma matriz \mathcal{A} de ordem 42.630, onde aproximadamente 0.072% são elementos não nulos e existem 39.690 componentes de velocidade e 2.940 de pressão.

6.3 Reservatório

Neste teste foi simulado o escoamento em um braço de reservatório.

Esta simulação foi feita considerando-se $Re=10.000$, variando CFL, 1 e 5, e

- Matriz \mathcal{A} de ordem 8.783, onde aproximadamente 0.27% são elementos não nulos e existem 8.061 componentes de velocidade e 722 de pressão;
- Matriz \mathcal{A} de ordem 34.578, onde aproximadamente 0.085% são elementos não nulos e existem 32.303 componentes de velocidade e 2.275 de pressão.

Um exemplo da malha utilizada para simular o enchimento deste reservatório pode ser vista na Figura 6.

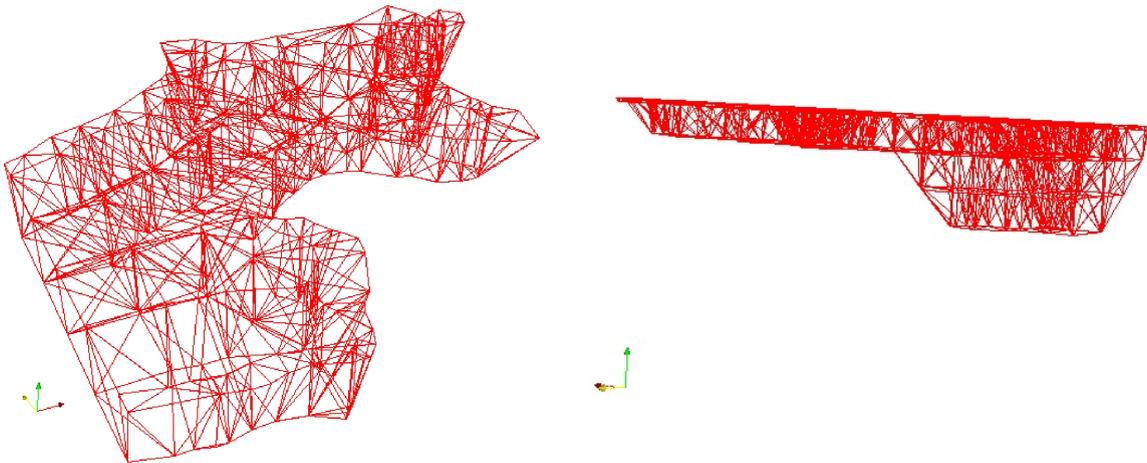


Figura 6: Exemplo de malha utilizada na simulação do compartimento de reservatório.

Todas as tabelas com os resultados obtidos podem ser encontradas nos apêndices. Algumas observações acerca destes testes foram feitas no Capítulo 6.4. Podemos ver nas figuras a seguir o decaimento do resíduo para alguns solucionadores e preconditionadores escolhidos.

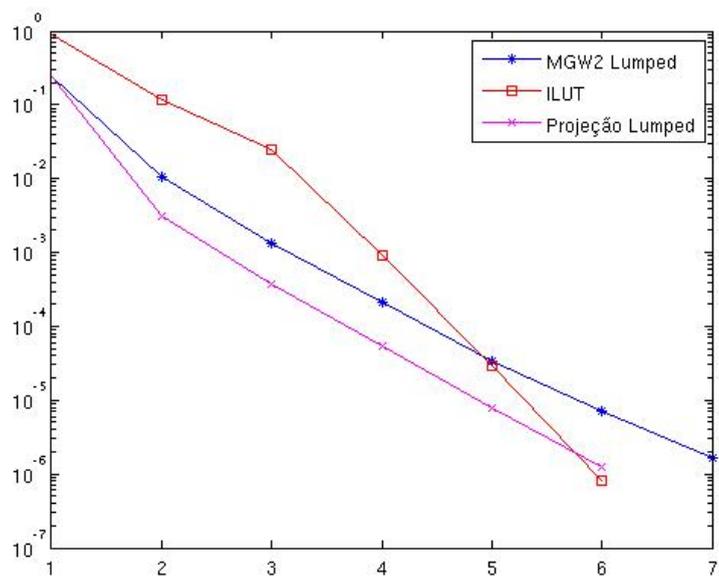


Figura 7: Decaimento do resíduo para alguns preconditionadores utilizando o PE-GMRes.

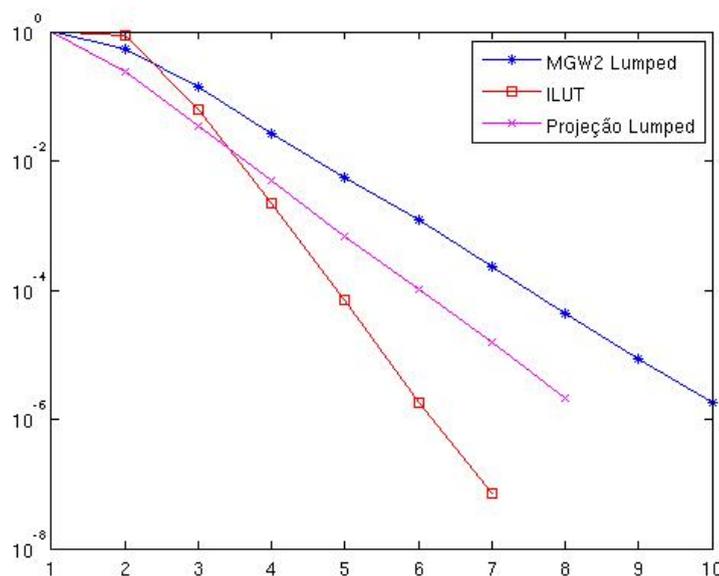


Figura 8: Decaimento do resíduo para alguns preconditionadores utilizando o PD-GMRes.

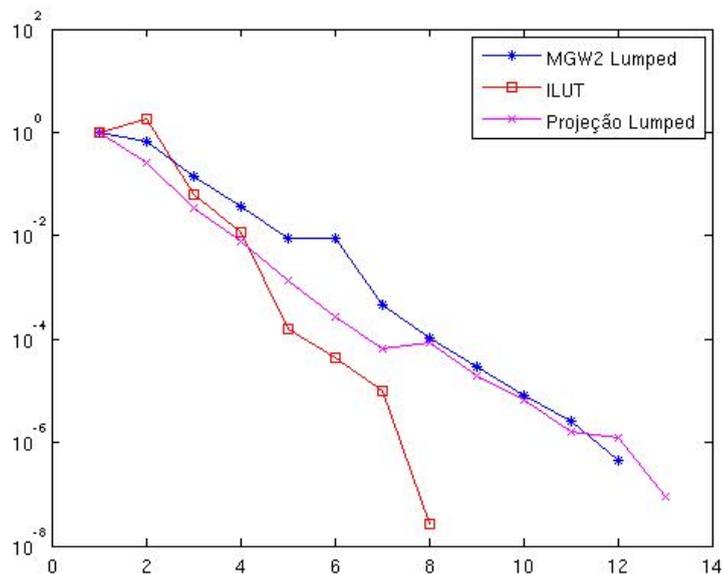


Figura 9: Decaimento do resíduo para alguns preconditionadores utilizando o Bi-CGStab.

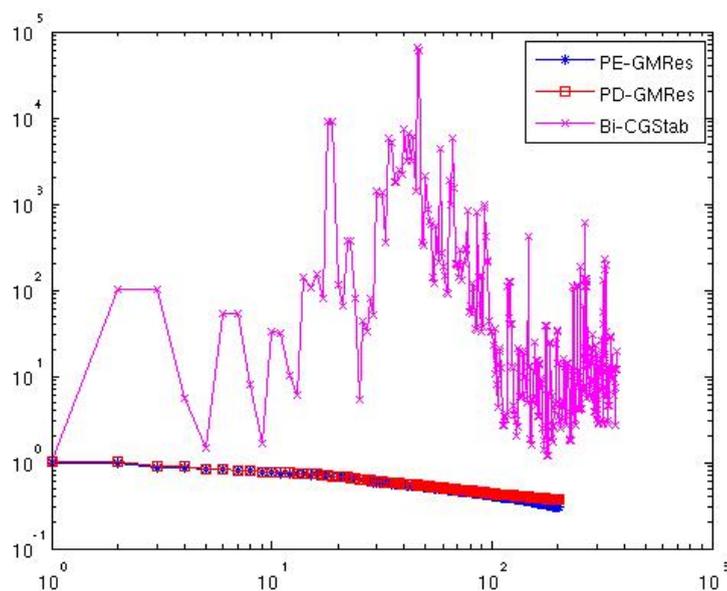


Figura 10: Decaimento do resíduo para o preconditionador diagonal utilizando os três solucionadores.

6.4 Conclusões

Foram utilizados, como testes, todos as combinações possíveis de reordenamentos, métodos, solucionadores, preconditionadores e aproximantes. Foram apresentados

- 2 reordenamentos:
 1. Grau mínimo, e
 2. Cuthill-McKee reverso;
- Para cada um deles utilizamos 2 classes de métodos:
 1. 2 segregados:
 1. Projeção, e
 2. ABLUF;
 2. Cada um deles com 5 aproximantes ou solucionadores:
 1. Diagonal,
 2. Lumped,
 3. Matriz completa aproximada,
 4. CGLS, e
 5. Probing;
 3. 3 acoplados:
 1. GMRes preconditionado pela esquerda,
 2. GMRes preconditionado pela direita,
 3. Bi-CGStab;
 4. Cada um deles com 7 preconditionadores:
 1. Diagonal,
 2. Lumped,
 3. ILU(0),
 4. ILUT(10^{-3}),
 5. Projeção:
 - Com os 5 aproximantes supracitados;
 6. MGW 1:
 - Com os 5 aproximantes supracitados;
 7. MGW 2:

– Com os 5 aproximantes supracitados.

Além destes ainda existem os testes do GMRes e do Bi-CGStab sem preconditionamento. Sendo assim, contabilizados 2.306 testes comparativos, que forneceram informações sobre

1. Tempo de construção dos preconditionadores,
2. Tempo de resolução dos sistemas lineares,
3. Erro relativo,
4. Número de iterações, e
5. Quantidade de memória exigida.

Devido à forma como feita a aproximação para o preconditionador ILU(0) (subseção 5.3.1), este é, para o problema tratado, diversas vezes de posto incompleto e além disso, devido à sua qualidade como aproximador, não foi capaz de promover a convergência em nenhum dos casos tratados. Pro este motivo não foi incluído nas tabelas de resultados.

Os testes realizados com a forma de resolução para os sistemas lineares do complemento de Schur que utiliza o CGLS (subseção 4.5.1) não foi incluído nas tabelas de testes, pois para tempo de construção dos preconditionadores e resolução dos sistemas lineares foi elevado demais e não produziu resultados de qualidade, além disso, em muitos casos não houve convergência.

A técnica de sondagem (seção 5.7) para aproximação do complemento de Schur foi de fato ineficiente. Seu tempo de construção é elevado e não apresentou resultados satisfatórios a ponto de serem incluídos nas tabelas de testes. Esta técnica de aproximação foi utilizada de forma simples e já era esperado o seu comportamento. Esta forma de aproximação fornece resultados bons para matrizes com estrutura regular, que não é o caso.

Quanto a medida de memória requerida, dos formas de resolução que serão apresentadas, há apenas três casos distintos para cada modelo e cada tamanho de malha. Os métodos acoplados com preconditionadores baseados na forma segregada, utilizaram a mesma quantidade de memória que os respectivos métodos segregados. Isto se deve ao fato de não ser necessário montar a matriz acoplada, apenas precisamos descrever a forma como será feita a multiplicação matriz vetor exigida pelo método acoplado, que foi feita por blocos. Tanto as aproximantes na forma diagonal quanto na forma lumped gastaram a mesma quantidade de memória. A resolução pelo

método acoplado sem preconditionador tem gasto de memória que pode ser aproximado ao gasto de memória exigido pelo uso do preconditionador diagonal ou lumped. Como para estes últimos três preconditionadores não houve qualidade na solução encontrada, podem ser descartados, e seu gasto com memória não será levado em consideração.

Assim, podemos destacar os três casos utilizando representantes: utilização da forma segregada com aproximante diagonal, utilização da forma segregada com aproximante da forma matriz aproximada completa e método acoplado com preconditionador ILUT. Sendo que o ILUT para o problema acoplado pode ter mais do que o dobro do uso de memória que o necessário para a forma segregada com aproximante diagonal.

Com relação à utilização dos reordenamentos, a utilização de qualquer um dos apresentados representou um aumento em velocidade, de 80 vezes em relação as resoluções que utilizam fatoração LU sem reordenamentos. Os reordenamentos de grau mínimo e Cuthill-McKee reverso não tiveram diferença significativa entre si. Devido à pequena vantagem obtida pelo Cuthill-McKee reverso optamos por sua utilização em todos os testes apresentados. Esta ligeira vantagem se deve ao fato deste reordenamento atuar de forma global enquanto o de grau mínimo atua de forma local, a Figura 11 mostra a pequena diferença gerada, que pode acarretar um preenchimento maior.

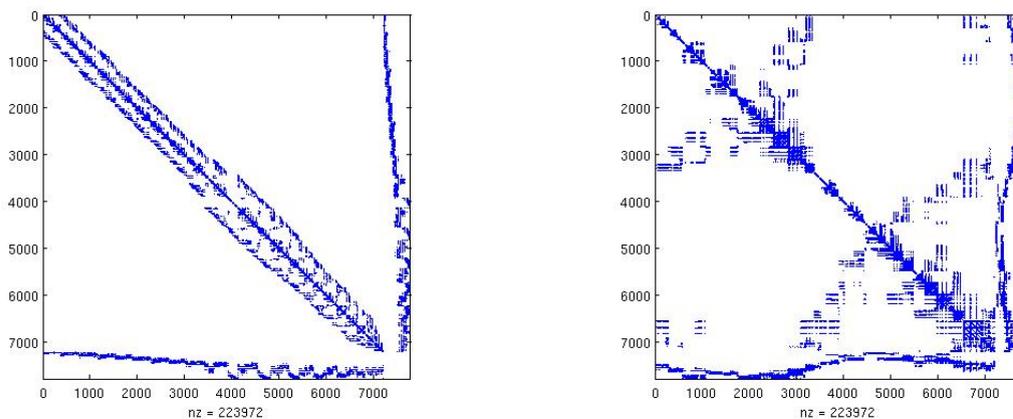


Figura 11: Exemplo: Reordenamentos Cuthill-McKee reverso e grau mínimo.

A forma de reordenamento que estamos utilizando visa beneficiar a decomposição LU do complemento de Schur, isto é, reordenamos a matriz \mathbf{A} e reordenamos a matriz $\mathbf{S} = -\tilde{\mathbf{D}}\mathbf{A}^{-1}\mathbf{G}$. De acordo com o reordenamento feito em \mathbf{S} reordenamos \mathbf{D} e \mathbf{G} para não ter problemas de reordenamento ds variáveis.

Nas tabelas que seguem, a aproximação do complemento de Schur da forma matriz aproximada completa, será chamada para efeito de simplificação de “app”.

Levando em consideração a soma do tempo de construção do preconditionador e de resolução dos sistemas chegou-se aos quadros 1 e 2 que apresenta os dois melhores resultados para cada problema tratado, com CFL=1 e CFL=2, respectivamente.

Tabela 1: Melhores tempos em cada problema tratado, CFL=1

Problema	Solucionador	Precond	Aprox.	T.Prec.	T.Sol.	Erro
Canal	PD-GMRes	Projeção	diagonal	3.30	8.20	3.1e-8
	PD-GMRes	MGW 2	lumped	3.13	9.01	6.3e-8
Degrau	PD-GMRes	Projeção	diagonal	3.30	8.85	9.9e-9
	PD-GMRes	MGW 2	lumped	3.12	9.08	1.2e-7
Caledônia	PE-GMRes	MGW 2	lumped	2.04	9.35	8e-6
	PE-GMRes	Projeção	diagonal	2.19	9.30	4.6e-6

Para a tabela 1 de CFL=1, podemos reparar que há uma predominância do solucionador GMRes preconditionado pela direita no entanto, existe uma variação na combinação solucionador-precondicionador. Além disso, é possível reparar que não há uma diferença significativa de tempo e acurácia da solução entre os dois melhores resultados apresentados para cada problema.

Tabela 2: Melhores tempos em cada problema tratado, CFL=5

Problema	Solucionador	Precond	Aprox.	T.Prec.	T.Sol.	Erro
Canal	Bi-CGStab	projeção	lumped	0.49	1.17	1.4e-7
	PD-GMRes	projeção	diagonal	0.49	1.22	9e-8
Degrau	PD-GMRes	MGW 2	lumped	3.41	12.1	3.3e-8
	Bi-CGStab	MGW 1	lumped	3.42	12.2	2.9e-8
Caledônia	PE-GMRes	MGW 2	lumped	2.10	12.8	3e-5
	PE-GMRes	Projeção	diagonal	2.25	12.4	3.6e-5

Para a tabela 2 de CFL=5, existe uma grande variação solucionadores e preconditionadores. No entanto, parece que tanto para CFL=1 quanto para CFL=2 há uma concordância em relação à simulação do reservatório. Além disso, é possível reparar que não há uma diferença significativa de tempo e acurácia da solução entre os dois melhores resultados apresentados para cada problema.

Apesar disso, é notável que o melhor preconditionador para a resolução de um passo de tempo para um problema de escoamento de fluido, não está determinado. Diversas combinações apresentam bons resultados em termos de velocidade de convergência enquanto que, para caso de necessitar de boa acurácia deve-se ponderar de forma a permitir uma pequena perda em termos de tempo.

REFERÊNCIAS

- ANJOS, G. R. dos. *Solução do Campo Hidrodinâmico em Células Eletroquímicas pelo Método de Elementos Finitos*. Dissertação (Mestrado) — COPPE/UFRJ, Março 2007.
- ARNOLDI, W. The principle of minimized iteration in the solution of the matrix eigenvalue problem. *Quarterly of Applied Mathematics*, v. 9, p. 17–25, 1951.
- BARRETT, R. et al. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. Philadelphia: SIAM press, 1994.
- BENZI, M. Preconditioning techniques for large linear systems: a survey. *Journal of Computational Physics*, v. 182, n. 2, p. 418–477, November 2002.
- BENZI, M.; GOLUB, G. H.; LIESEN, J. Numerical solution of saddle point problems. *Acta Numerica*, v. 14, p. 1–137, May 2005.
- BILLIONNET, A. On interval graphs and matrice profiles. *RAIRO Reck Opér.*, v. 20, p. 245–256, 1986.
- BONDY, J. A.; MURTY, U. S. R. *Graph Theory with Applications*. London: Macmillan, 1976.
- CHANG, W.; GIRALDO, F.; PEROT, B. Analysis of an exact fractional step method. *Journal of Computational Physics*, v. 180, p. 183–199, 2002.
- CHORIN, A. J. Numerical solution of the navier-stokes equations. *Mathmatics of Computation*, v. 22, p. 745–762, 1968.
- CUTHILL, E.; MCKEE, J. Reducing the bandwidth of sparse symmetric matrices. In: *Proc. 24th Nat. Conf. ACM*. [S.l.: s.n.], 1969. p. 157–172.
- DAVIES, A. J. *The finite element method: a first approach*. [S.l.]: Claredon Press - Oxford, 1980.
- DONGARRA, J. J. Performance of various computers using standard linear equation solvers. n. CS - 89 - 85, March 2008.
- FLETCHER, R. Conjugate gradient methods for indefinite systems. In: UNIV. DUNDEE. *Numerical Analysis 1975: Proc. 6th Biennial Dundee Conf*. Dundee: Springer, Berlin, 1976. (Lecture Notes in Mathematics, v. 506), p. 73–89.
- FORTUNA, A. O. *Técnicas Computacionais para Dinâmica dos Fluidos*. [S.l.]: Edusp, 2000.

FREUND, R. W.; GOLUB, G. H.; NACHTIGAL, N. M. Iterative solution of linear systems. *Acta Numerica*, Cambridge University Press, p. 57–100, 1992.

GAMA, R. M. S. da. *Elementos de Cinemática dos Meios Contínuos*. [S.l.]: Universidade do Estado do Rio de Janeiro, 2005.

GEORGE, A. *Computer implementation of the finite element model*. Computer Sci. Dept., Stanford Univ., Stanford, California, 1971.

GEORGE, A.; LIU, J. W. H. An implementation of a pseudoperipheral node finder. *ACM Trans. Math. Softw.*, ACM Press, New York, NY, USA, v. 5, n. 3, p. 284–295, 1979. ISSN 0098-3500. Disponível em: <<http://doi.acm.org/10.1145/355841.355845>>.

GEORGE, A.; LIU, J. W. H. The evolution of the minimum degree ordering algorithm. *SIAM Review*, v. 31, n. 1, p. 1–19, March 1989.

GIBBS, N. E.; JR., W. G. P.; STOCKMEYER, P. K. An algorithm for reducing the bandwidth and profile of a sparse matrix. *SIAM Journal of Numerical Analysis*, v. 13, n. 2, p. 236–250, abr. 1976.

GIRAUD, L. Private communication. February 2008.

GOLUB, G. H.; LOAN, C. F. van. *Matrix Computations*. 3rd. ed. [S.l.]: Johns Hopkins University Press, 1996.

GOLUB, G. H.; O'LEARY, D. P. Some history of the conjugate gradient and lanczos algorithms: 1948–1976. *SIAM Review*, SIAM, v. 31, n. 1, p. 50–102, 1989. Disponível em: <<http://link.aip.org/link/?SIR/31/50/1>>.

GREENBAUM, A. *Iterative methods for solving linear systems*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1997. ISBN 0-89871-396-X.

GRESHO, P. On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via the finite element method that also introduces a nearly consistent mass matrix. part 1: Theory. *International Journal for Numerical Methods in Fluids*, v. 11, p. 587–620, 1990.

GRESHO, P.; CHAN, S. T. On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via the finite element method that also introduces a nearly consistent mass matrix. part 2: Implementation. *International Journal for Numerical Methods in Fluids*, v. 11, p. 621–659, 1990.

HARLOW, F. H.; WELCH, J. E. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *The Physics of Fluids*, v. 8, n. 1, p. 2182–2189, 1965.

HESTENES, M.; STIEFEL, E. Methods of conjugate gradients for solving linear systems. *J. Res. Nat. Bur. Stand*, v. 49, p. 409–436, 1952.

HORN, R.; JOHNSON, C. R. *Matrix Analysis*. [s.n.], 1985. xiii + 561 p. ISBN 0-521-30586-1 (hard), 0-521-38632-2 (soft). Disponível em: <<http://www.loc.gov/catdir/description/cam023/85007736.html>>.

LANCZOS, C. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of Research of the National Bureau of Standards*, v. 45, n. 4, p. 255–282, October 1950.

LANCZOS, C. Solution of systems of linear equations by minimized iterations. *Journal of Research of the National Bureau of Standards*, v. 49, n. 1, p. 33–53, July 1952.

LEE, M. J.; OH, B. D.; KIM, Y. B. Canonical fractional-step methods and consistent boundary conditions for the incompressible navier-stokes equations. *Journal of Computational Physics*, v. 168, p. 73–100, 2001.

LIU, W.-H.; SHERMAN, A. H. Comparative analysis of the Cuthill-McKee and the reverse Cuthill-McKee ordering algorithms for sparse matrices. *SIAM Journal on Numerical Analysis*, v. 13, n. 2, p. 198–213, abr. 1976.

MEURANT, G. *The Lanczos and Conjugate Gradient Algorithms: From Theory to Finite Precision Computations (Software, Environments, and Tools)*. [S.I.]: SIAM, 2006. ISBN 0898716160.

MEYER, C. D. *Matrix analysis and applied linear algebra*. [S.I.]: SIAM, 2000.

MURPHY, M. F.; GOLUB, G. H.; WATHEN, A. J. A note on preconditioning for indefinite linear systems. *SIAM Journal on Scientific Computing*, v. 21, n. 6, p. 1969–1972, 2000.

NI, M.-J.; KOMORI, S.; MORLEY, N. Projection methods for the calculation of incompressible unsteady flows. *Numerical Heat Transfer*, v. 44, p. 553–551, 2003.

PATANKAR, S. *Numerical heat transfer and fluid flow*. New York: Hemisphere, 1980.

PEROT, J. B. An analysis of the fractional step method. *Journal of Computational Physics*, v. 108, p. 51–58, 1993.

REID, J. K. On the method of conjugate gradients for the solution of large sparse systems of linear equations. In: REID, J. K. (Ed.). *Large Sparse Sets of Linear Equations*. New York: Academic Press, 1971. p. 231–254.

SAAD, Y. *Iterative Methods for Sparse Linear Systems*. 2nd. ed. [S.I.]: SIAM, 2003.

- SAAD, Y.; SCHULTZ, M. H. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, v. 7, n. 3, p. 856–869, 1986.
- SAAD, Y.; VORST, H. A. van der. Iterative solution of linear systems in the 20th century. *Journal of Computational and Applied Mathematics*, v. 123, n. 1-2, p. 1–23, 2000.
- SHEWCHUK, J. R. *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain Edition 1 $\frac{1}{4}$* . [S.l.], August 1994.
- SIEFERT, C.; STURLER, E. de. Probing methods for saddle-point problems. *ETNA*, v. 22, p. 163–183, 2006.
- SIMEONI, F. *Simulação de Escoamentos Multifásicos em Malhas não Estruturadas*. Tese (Doutorado) — Universidade de São Paulo, São Carlos,, Agosto 2005.
- SIMONCINI, V.; SZYLD, D. B. Recent computational developments in krylov subspace methods for linear systems. *Numerical Linear Algebra with Applications*, v. 14, p. 1 – 59, 2007.
- SONNEVELD, P. CGS: a fast lanczos-type solver for nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, v. 10, p. 32–56, 1989.
- TOMÉ, M. F.; MCKEE, S. Gensmac: A computational marker-and-cell method for free surface flows in general domains. *Journal of Computational Physics*, v. 110, n. 1, p. 171–189, 1994.
- TREFETHEN, L. N.; Bau, III, D. (Ed.). *Numerical Linear Algebra*. Philadelphia, PA, USA: SIAM, 1997. xii + 361 p. ISBN 0-89871-361-7.
- VORST, H. A. van der. *Iterative Krylov Methods for Large Linear Systems*. Cambridge: Cambridge University Press, 2003. (Cambridge Monographs on Applied and Computational Mathematics). ISBN 0-521-81828-1.
- VORST, H. Van der. BICGSTAB: a fast and smoothly converging variant of BI-CG for the solution of non-symmetric linear systems. *SIAM Journal on Scien. and Stat. Computing*, v. 13, n. 2, p. 631–644, March 1992.
- YANNAKAKIS, M. Computing the minimum fill-in is np-complete. *SIAM Journal on Algebraic and Discrete Methods*, SIAM, v. 2, n. 1, p. 77–79, 1981. Disponível em: <<http://link.aip.org/link/?SML/2/77/1>>.
- ZIENKIEWICZ, O. C.; TAYLOR, R. L. *The Finite Element Method Volume 1: The Basics*. 5th. ed. [S.l.]: Butterworth-Heinemann, 2000.

ZIENKIEWICZ, O. C.; TAYLOR, R. L. *The Finite Element Method Volume 3: Fluid Mechanics*. 5th. ed. [S.l.]: Butterworth-Heinemann, 2000.

APÊNDICE A – Ranking

Neste Apêndice, se encontram tabelas que apresentam comparações entre as combinações de soappionadores, preconditionadores e aproximadores que, apresentaram os melhores resultados em termos de velocidade de convergência e acurácia do erro relativo, comparando os métodos acoplados e segregados.

Também se encontram tabelas que apresentam os melhores resultados nas três categoria citadas acima para cada soappionador utilizado, que são, GMRes preconditionado pela esquerda, GMRes preconditionado pela direita e Bi-CGStab.

Sempre que necessário, foram feitos comentarios relevantes acerca dos resultados obtidos.

A.1 Canal 3D**Tabela 3:** Ranking segregado/acoplado canal, $20 \times 10 \times 3$ e CFL=1

Método	Solucionador	Precond	Aprox.	T.Sol	Erro
Segregado	Projeção	ICC & ILU	lumped	0.13	4.1e-3
	ABLUF		lumped	0.83	6.6e-5
Acoplado	Bi-CGSTab	ILUT(10^{-3})	***	0.37	1.7e-7
	PE-GMRes	ILUT(10^{-3})	***	0.53	5.4e-11

Tabela 4: Ranking canal acoplado, $20 \times 10 \times 3$ e CFL=1

Solucionador	Precond	Aprox.	T.Sol	Erro
PE-GMRes	ILUT(10^{-3})	***	0.53	5.4e-11
	ILUT(10^{-3})	***	0.53	5.4e-11
PD-GMRES	ILUT(10^{-3})	***	0.42	3.4e-8
	Projeção	lumped	1.32	1e-8
Bi-CGSTab	ILUT(10^{-3})	***	0.37	1.7e-7
	Projeção	lumped	1.51	1.9e-9

Tabela 5: Ranking segregado/acoplado canal, $50 \times 20 \times 3$ e CFL=1

Método	Solucionador	Precond	Aprox.	T.Sol	Erro
Segregado	Projeção	ICC & ILU	lumped	1.02	4.7e-3
	ABLUF		lumped	6.57	2e-6
Acoplado	PD-GMRes	ILUT(10^{-3})	***	2.0	0.21
	PE-GMRes	ILUT(10^{-3})	***	5.86	8.9e-13

Tabela 6: Ranking canal acoplado, $50 \times 20 \times 3$ e CFL=1

Solucionador	Precond	Aprox.	T.Sol	Erro
PE-GMRes	ILUT(10^{-3})	***	5.86	8.9e-13
	ILUT(10^{-3})	***	5.86	8.9e-13
PD-GMRES	ILUT(10^{-3})	***	2.0	0.21
	MGW 1	diag & lump	18.7	5.3e-9
Bi-CGSTab	ILUT(10^{-3})	***	4.31	4.7e-7
	projeção	diagonal	10.0	3.42e-9

Tabela 7: Ranking segregado/acoplado canal, $20 \times 10 \times 3$ e CFL=5

Método	Solucionador	Precond	Aprox.	T.Sol	Erro
Segregado	Projeção	ICC & ILU	lumped	0.13	3.1e-2
	ABLUF		lumped	0.84	1.1e-5
Acoplado	Bi-CGSTab	Projeção	lumped	1.17	1.4e-7
	PE-GMRes	ILUT(10^{-3})	***	2.61	4.5e-11

Tabela 8: Ranking canal acoplado, $20 \times 10 \times 3$ e CFL=5

Solucionador	Precond	Aprox.	T.Sol	Erro
PE-GMRes	Projeção	diagonal	1.82	1.1e-8
	ILUT(10^{-3})	***	2.61	4.5e-11
PD-GMRES	Projeção	diagonal	1.22	9e-8
	Projeção	app	1.57	1.8e-8
Bi-CGSTab	Projeção	lumped	1.17	1.4e-7
	MGW 2	app	1.38	1.3e-9

Tabela 9: Ranking segregado/acoplado canal, $50 \times 20 \times 3$ e CFL=5

Método	Solucionador	Precond	Aprox.	T.Sol	Erro
Segregado	Projeção	ICC & ILU	lumped	1.00	3.3e-2
	ABLUF		diagonal	6.32	6.2e-5
Acoplado	PD-GMRes	Projeção	lumped	12.0	5.8e-8
	PE-GMRes	ILUT(10^{-3})	***	32.3	8.3e-12

Tabela 10: Ranking canal acoplado, $50 \times 20 \times 3$ e CFL=5

Solucionador	Precond	Aprox.	T.Sol	Erro
PE-GMRes	MGW 2	diag & proj	18.4	1.4e-9
	ILUT(10^{-3})	***	32.3	8.3e-12
PD-GMRES	Projeção	lumped	12.0	5.8e-8
	ILUT(10^{-3})	***	23.0	9.6e-9
Bi-CGSTab	Projeção	diagonal	12.2	3.4e-8
	ILUT(10^{-3})	***	22.7	2.4e-7

A.2 Degrau 3D**Tabela 11:** Ranking segregado/acoplado degrau, $10 \times 5 \times 3$ e CFL=1

Método	Solucionador	Precond	Aprox.	T.Sol	Erro
Segregado	Projeção	ICC & ILU	lumped	2e-2	1e-2
	ABLUF		app	0.25	3.4e-5
Acoplado	Bi-CGSTab	ILUT(10^{-3})	***	4.8e-2	3.8e-10
	PE-GMRes	ILUT(10^{-3})	***	0.07	2.2e-11

Tabela 12: Ranking degrau acoplado, $10 \times 5 \times 3$ e CFL=1

Solucionador	Precond	Aprox.	T.Sol	Erro
PE-GMRes	ILUT(10^{-3})	***	0.07	2.2e-11
	ILUT(10^{-3})	***	0.07	2.2e-11
PD-GMRES	ILUT(10^{-3})	***	6.2e-2	8.6e-9
	ILUT(10^{-3})	***	0.10	5.6e-9
Bi-CGSTab	ILUT(10^{-3})	***	4.8e-2	3.8e-10
	ILUT(10^{-3})	***	4.8e-2	3.8e-10

Tabela 13: Ranking segregado/acoplado degrau, $20 \times 10 \times 3$ e CFL=1

Método	Solucionador	Precond	Aprox.	T.Sol	Erro
Segregado	Projeção	ICC & ILU	lumped	0.13	8.2e-3
	ABLUF		lumped	0.83	2.2e-5
Acoplado	Bi-CGSTab	ILUT(10^{-3})	***	0.37	6.5e-8
	PE-GMRes	ILUT(10^{-3})	***	0.53	5.5e-11

Tabela 14: Ranking degrau acoplado, $20 \times 10 \times 3$ e CFL=1

Solucionador	Precond	Aprox.	T.Sol	Erro
PE-GMRes	ILUT(10^{-3})	***	0.53	5.5e-11
	ILUT(10^{-3})	***	0.53	5.5e-11
PD-GMRES	ILUT(10^{-3})	***	0.42	5.1e-8
	Projeção	diagonal	1.37	2.8e-9
Bi-CGSTab	ILUT(10^{-3})	***	0.37	6.5e-8
	Projeção	diagonal	1.52	4.1e-9

Tabela 15: Ranking segregado/acoplado degrau, $50 \times 20 \times 3$ e CFL=1

Método	Solucionador	Precond	Aprox.	T.Sol	Erro
Segregado	Projeção	ICC & ILU	lumped	1.01	7.3e-3
	ABLUF		lumped	6.54	1.3e-6
Acoplado	PD-GMRES	ILUT(10^{-3})	***	1.97	0.25
	PE-GMRes	ILUT(10^{-3})	***	5.81	1.8e-12

Tabela 16: Ranking degrau acoplado, $50 \times 20 \times 3$ e CFL=1

Solucionador	Precond	Aprox.	T.Sol	Erro
PE-GMRes	ILUT(10^{-3})	***	5.81	1.8e-12
	ILUT(10^{-3})	***	5.81	1.8e-12
PD-GMRES	ILUT(10^{-3})	***	1.97	0.25
	Projeção	diagonal	8.85	9.9e-9
Bi-CGSTab	ILUT(10^{-3})	***	4.66	1.1e-6
	Projeção	diagonal	10.0	1.9e-9

Tabela 17: Ranking segregado/acoplado degrau, $10 \times 5 \times 3$ e CFL=5

Método	Solucionador	Precond	Aprox.	T.Sol	Erro
Segregado	Projeção	ICC & ILU	lumped	1.9e-2	3.3e-2
	ABLUF		lumped	0.13	1.8e-5
Acoplado	Bi-CGSTab	ILUT(10^{-3})	***	0.12	1.1e-6
	PE-GMRes	ILUT(10^{-3})	***	0.26	1.7e-10

Tabela 18: Ranking degrau acoplado, $10 \times 5 \times 3$ e CFL=5

Solucionador	Precond	Aprox.	T.Sol	Erro
PE-GMRes	MGW 1 & 2	app	0.22	1.6e-8
	ILUT(10^{-3})	***	0.26	1.7e-10
PD-GMRES	MGW 2	app	8.7e-2	6.7e-8
	Projeção	app	0.17	9e-9
Bi-CGSTab	MGW 1 & 2	app	8.7e-2	1.8e-8
	MGW 1 & 2	app	8.7e-2	1.8e-8

Tabela 19: Ranking segregado/acoplado degrau, $20 \times 10 \times 3$ e CFL=5

Método	Solucionador	Precond	Aprox.	T.Sol	Erro
Segregado	Projeção	ICC & ILU	lumped	0.12	3.4e-2
	ABLUF		lumped	0.81	1.5e-5
Acoplado	PD-GMRES	ILUT(10^{-3})	***	1.15	5e-7
	PE-GMRes	ILUT(10^{-3})	***	2.6	5.6e-11

Tabela 20: Ranking degrau acoplado, $20 \times 10 \times 3$ e CFL=5

Solucionador	Precond	Aprox.	T.Sol	Erro
PE-GMRes	MGW 2	app	1.44	3.5e-8
	ILUT(10^{-3})	***	2.6	5.6e-11
PD-GMRES	Projeção	lumped	1.15	5e-7
	MGW 2	app	1.15	2.8e-8
Bi-CGSTab	Projeção	lumped	1.15	1.9e-7
	MGW 1 & 2	app	1.38	3.4e-9

Tabela 21: Ranking segregado/acoplado degrau, $50 \times 20 \times 3$ e CFL=5

Método	Solucionador	Precond	Aprox.	T.Sol	Erro
Segregado	Projeção	ICC & ILU	lumped	1.02	4.1e-2
	ABLUF		diag	6.5	7.4e-5
Acoplado	PD-GMRES	MGW 2	diagonal	12.0	3.3e-8
	PE-GMRes	ILUT(10^{-3})	***	32.3	8e-12

Tabela 22: Ranking degrau acoplado, $50 \times 20 \times 3$ e CFL=5

Solucionador	Precond	Aprox.	T.Sol	Erro
PE-GMRes	MGW 2	diag & lump	18.3	2.4e-9
	ILUT(10^{-3})	***	32.3	8e-12
PD-GMRES	MGW 2	diagonal	12.0	3.3e-8
	ILUT(10^{-3})	***	23.0	1e-8
Bi-CGSTab	MGW 1 & 2	diag & lump	12.2	2.9e-8
	Projeção	diagonal	13.3	4.8e-9

A.3 Reservatório**Tabela 23:** Ranking segregado/acoplado Caledônia, sem refinamento e CFL=1

Método	Solucionador	Precond	Aprox.	T.Sol	Erro
Segregado	Projeção	ICC & ILU	lumped	0.24	1.14e-2
	ABLUF		lumped	1.01	2.8e-4
Acoplado	Bi-CGSTab	ILUT(10^{-3})	***	1.14	1e-8
	Bi-CGSTab	Projeção	diagonal	2.41	8.2e-9

Tabela 24: Ranking Caledônia, acoplado sem refinamento e CFL=1

Solucionador	Precond	Aprox.	T.Sol.	Erro
PE-GMRes	ILUT(10^{-3})	***	1.19	2.3e-6
	Projeção	app	3.74	1.5e-6
PD-GMRES	ILUT(10^{-3})	***	1.21	9e-8
	ILUT(10^{-3})	***	1.21	9e-8
Bi-CGSTab	ILUT(10^{-3})	***	1.14	1e-8
	Projeção	diagonal	2.41	8.2e-9

Tabela 25: Ranking segregado/acoplado Caledônia, com refinamento e CFL=1

Método	Solucionador	Precond	Aprox.	T.Sol	Erro
Segregado	Projeção	ICC & ILU	lumped	1.01	1.1e-2
	ABLUF		lumped	5.94	1.7e-4
Acoplado	Bi-CGSTab	ILUT(10^{-3})	***	7.69	1e-7
	Bi-CGSTab	Projeção	diagonal	16.6	1.8e-9

Tabela 26: Ranking Caledônia, acoplado com refinamento e CFL=1

Solucionador	Precond	Aprox.	T.Sol.	Erro
PE-GMRes	ILUT(10^{-3})	***	8.83	6.7e-8
	ILUT(10^{-3})	***	8.83	6.7e-8
PD-GMRES	ILUT(10^{-3})	***	7.95	8.2e-8
	ILUT(10^{-3})	***	7.95	8.2e-8
Bi-CGSTab	ILUT(10^{-3})	***	7.69	1e-7
	Projeção	diagonal	16.6	1.8e-9

Tabela 27: Ranking segregado/acoplado Caledônia, sem refinamento e CFL=5

Método	Solucionador	Precond	Aprox.	T.Sol	Erro
Segregado	Projeção	ICC & ILU	lumped	0.14	4.8e-2
	ABLUF		lumped	0.92	2.2e-3
Acoplado	Bi-CGSTab	ILUT(10^{-3})	***	1.03	3e-6
	Bi-CGSTab	Projeção	diagonal	1.73	3.8e-8

Tabela 28: Ranking Caledônia, acoplado sem refinamento e CFL=5

Solucionador	Precond	Aprox.	T.Sol.	Erro
PE-GMRes	Projeção	diagonal	1.26	7.9e-5
	ILUT(10^{-3})	***	1.37	1.5e-6
PD-GMRES	ILUT(10^{-3})	***	1.24	1.6e-6
	MGW 2	app	2.16	8.6e-7
Bi-CGSTab	ILUT(10^{-3})	***	1.03	3e-6
	Projeção	diagonal	1.73	3.8e-8

Tabela 29: Ranking segregado/acoplado Caledônia, com refinamento e CFL=5

Método	Solucionador	Precond	Aprox.	T.Sol	Erro
Segregado	Projeção	ICC & ILU	lumped	0.94	8.6e-2
	ABLUF		lumped	5.54	2.2e-3
Acoplado	Bi-CGSTab	ILUT(10^{-3})	***	11.5	1.1e-6
	PD-GMRES	ILUT(10^{-3})	***	11.8	6.2e-7

Tabela 30: Ranking Caledônia, acoplado com refinamento e CFL=5

Solucionador	Precond	Aprox.	T.Sol.	Erro
PE-GMRes	Projeção	diagonal	12.4	3.6e-5
	Projeção	diagonal	12.4	3.6e-5
PD-GMRES	ILUT(10^{-3})	***	11.8	6.2e-7
	ILUT(10^{-3})	***	11.8	6.2e-7
Bi-CGSTab	ILUT(10^{-3})	***	11.5	1.1e-6
	ILUT(10^{-3})	***	11.5	1.1e-6

APÊNDICE B – Tabelas de resultados com CFL=1

Neste Apêndice, pode-se encontrar as tabelas com todos os testes com CFL=1. Estas tabelas contêm resultados sobre tempo de construção dos preconditionadores, tempo de resolução dos sistemas lineares, erro relativo, e número de iterações.

Sempre que necessário, foram feitos comentários relevantes acerca dos resultados obtidos.

Tabela 31: Canal, segregado com malha $20 \times 10 \times 3$ e CFL=1

Solucionador	Aprox.	Precond	T.Prec	T.Sol	Erro	Iter
Projeção	diag	ICC & ILU	0.48	0.24	0.48	***
	lumped		0.48	0.13	4.1e-3	***
	app		3.63	0.24	0.38	***
ABLUF	diag		0.48	0.91	2.6e-2	***
	lumped		0.48	0.83	6.6e-5	***
	app		3.60	1.03	1.5e-2	***

Tabela 32: Canal, segregado com malha $50 \times 20 \times 3$ e CFL=1

Solucionador	Aprox.	Precond	T.Prec	T.Sol	Erro	Iter
Projeção	diag	ICC & ILU	3.30	1.16	0.44	***
	lumped		3.31	1.02	4.7e-3	***
	app		110	2.30	0.94	***
ABLUF	diag		3.30	6.17	1.6e-2	***
	lumped		3.31	6.57	2e-6	***
	app		110	11.3	0.77	***

Tabela 33: Degrau, segregado com malha $10 \times 5 \times 3$ e CFL=1

Solucionador	Aprox.	Precond	T.Prec	T.Sol	Erro	Iter
Projeção	diag	ICC & ILU	6.6e-2	2.5e-2	0.47	***
	lumped		6.6e-2	2e-2	1e-2	***
	app		0.19	0.10	0.13	***
ABLUF	diag		6.6e-2	0.13	2.6e-2	***
	lumped		6.7e-2	0.12	1.6e-4	***
	app		0.17	0.25	3.4e-5	***

Tabela 34: Degrau, segregado com malha 20x10x3 e CFL=1

Solucionador	Aprox.	Precond	T.Prec	T.Sol	Erro	Iter
Projeção	diag	ICC & ILU	0.48	0.28	0.49	***
	lumped		0.48	0.13	8.2e-3	***
	app		3.62	0.24	0.37	***
ABLUF	diag		0.47	1.00	2.8e-2	***
	lumped		0.48	0.83	2.2e-5	***
	app		3.60	1.03	1.5e-2	***

Tabela 35: Degrau, segregado com malha 50x20x3 e CFL=1

Solucionador	Aprox.	Precond	T.Prec	T.Sol	Erro	Iter
Projeção	diag	ICC & ILU	5.56	2.39	0.43	***
	lumped		3.31	1.01	7.3e-3	***
	app		110	2.21	0.93	***
ABLUF	diag		3.3	6.17	1.6e-2	***
	lumped		3.3	6.54	1.3e-6	***
	app		110	11.5	0.77	***

Tabela 36: Caledônia, segregado sem refinamento e CFL=1

Solucionador	Aprox.	Precond	T.Prec	T.Sol	Erro	Iter
Projeção	diag	ICC & ILU	0.36	0.24	0.42	***
	lumped		0.36	0.24	1.14e-2	***
	app		3.45	0.25	0.97	***
ABLUF	diag		0.36	0.99	1.5e-2	***
	lumped		0.36	1.01	2.8e-4	***
	app		3.43	1.00	0.78	***

Tabela 37: Caledônia, segregado com refinamento e CFL=1

Solucionador	Aprox.	Precond	T.Prec	T.Sol	Erro	Iter
Projeção	diag	ICC & ILU	2.20	1.12	0.47	***
	lumped		2.20	1.01	1.1e-2	***
	app		50	1.15	0.97	***
ABLUF	diag		2.19	5.77	2.6e-2	***
	lumped		2.20	5.94	1.7e-4	***
	app		50	5.97	0.87	***

Tabela 38: Canal, acoplado com malha $20 \times 10 \times 3$ e CFL=1

Solucionador	Precond	Aprox.	T.Prec	T.Sol	Erro	Iter	
PE-GMRes	Nenhum	***	0.23	8.13	2.7e-4	200	
	Diagonal	***	0.23	8.23	2.2e-3	200	
	Lumped	***	0.23	8.21	0.97	200	
	ILUT(10^{-3})	***	1.39	0.53	5.4e-11	6	
	Projeção	diagonal		0.48	1.84	7.8e-10	10
		lumped		0.48	1.97	1.4e-9	10
		app		3.60	2.14	9.2e-10	8
	MGW 1	diagonal		0.48	3.65	5.1e-9	22
		lumped		0.45	3.64	5.1e-9	22
		app		3.60	3.05	6e-9	18
	MGW 2	diagonal		0.48	1.95	1.1e-8	11
		lumped		0.45	1.95	1.1e-8	11
		app		3.62	1.58	1.4e-8	8
	PD-GMRes	Diagonal	***	0.23	8.23	7.9e-4	200
		Lumped	***	0.23	8.19	0.99	200
ILUT(10^{-3})		***	1.39	0.42	3.4e-8	5	
Projeção		diagonal		0.48	1.25	4.6e-8	7
		lumped		0.48	1.32	1.e-8	7
		app		3.60	1.75	3e-9	7
MGW 1		diagonal		0.48	2.38	4.9e-8	15
		lumped		0.45	2.38	4.9e-8	15
		app		3.59	2.47	1.5e-8	15
MGW 2		diagonal		0.48	1.37	1.9e-8	8
		lumped		0.45	1.37	19e-8	8
		app		3.59	1.27	6.4e-9	7
Bi-CGStab		Nenhum	***	0.23	2.73	2.2e-4	200
		Diagonal	***	0.23	2.92	0.39	200
		Lumped	***	nc	nc	nc	nc
	ILUT(10^{-3})	***	1.39	0.37	1.7e-7	3	
	Projeção	diagonal		0.48	1.56	3.19e-9	5
		lumped		0.48	1.51	1.9e-9	5
		app		3.59	1.81	3.8e-8	4
	MGW 1	diagonal		0.48	1.23	4e-8	4
		lumped		0.45	1.23	4e-8	4
		app		3.61	1.36	1.1e-7	4
	MGW 2	diagonal		0.48	1.23	4e-8	4
		lumped		0.45	1.23	4e-8	4
		app		3.62	1.35	1.1e-7	4

Tabela 39: Canal, acoplado com malha $50 \times 20 \times 3$ e CFL=1

Solucionador	Precond	Aprox.	T.Prec	T.Sol	Erro	Iter	
PE-GMRes	Nenhum	***	1.40	0.95	1.01	9	
	Diagonal	***	1.42	0.98	0.99	9	
	Lumped	***	1.43	0.95	0.99	9	
	ILUT(10^{-3})	***	35.5	5.86	8.9e-13	12	
	Projeção	diagonal		3.30	13.2	8.1e-9	10
		lumped		3.30	15.4	1.3e-9	10
		app		110	46.5	9e-9	15
	MGW 1	diagonal		3.29	28.9	1.7e-8	22
		lumped		3.12	28.9	1.7e-8	22
		app		110	72.3	3.2e-8	30
	MGW 2	diagonal		3.29	15.4	1.3e-	11
		lumped		3.12	15.4	1.3e-9	11
		app		110	41.7	9.7e-8	15
	PD-GMRes	Diagonal	***	1.42	0.99	0.99	9
		Lumped	***	1.43	0.95	0.97	9
ILUT(10^{-3})		***	35.5	2.0	0.21	3	
Projeção		diagonal		3.30	8.2	3.1e-8	7
		lumped		3.30	10.3	1e-8	7
		app		110	32.6	7.2e-8	11
MGW 1		diagonal		3.29	18.7	5.3e-9	15
		lumped		3.12	18.7	5.3e-	15
		app		110	55.6	5.7e-8	23
MGW 2		diagonal		3.29	9.01	6.3e-8	7
		lumped		3.13	9.01	6.3e-8	7
		app		110	29.7	1.6e-7	11
Bi-CGStab		Nenhum	***	1.40	18.5	0.68	198
		Diagonal	***	1.42	19.5	0.93	200
		Lumped	***	nc	nc	nc	nc
	ILUT(10^{-3})	***	35.5	4.31	4.7e-7	6	
	Projeção	diagonal		3.30	10.0	3.42e-9	5
		lumped		3.31	9.56	1.1e-8	4
		app		110	44.4	1.6e-7	8
	MGW 1	diagonal		3.29	9.20	4.14e-8	4
		lumped		3.12	9.19	4.14e-8	4
		app		110	40.0	2.1e-7	8
	MGW 2	diagonal		3.29	9.20	4.14e-8	4
		lumped		3.12	9.20	4.1e-8	4
		app		110	40.3	2.1e-7	8

Tabela 40: Degrau, acoplado com malha 10x5x3 e CFL=1

Solucionador	Precond	Aprox.	T.Prec	T.Sol	Erro	Iter	
PE-GMRes	Nenhum	***	3.8e-2	1.71	2.3e-7	183	
	Diagonal	***	3.9e-2	1.26	7.1e-8	157	
	Lumped	***	3.9e-2	2.09	0.84	200	
	ILUT(10^{-3})	***	0.11	0.07	2.2e-11	5	
	Projeção	diagonal	6.6e-2	0.27	1.6e-8	10	
		lumped	6.6e-2	0.26	6.5e-9	9	
		app	0.17	0.24	3.9e-9	5	
	MGW 1	diagonal	6.6e-2	0.49	2.2e-8	20	
		lumped	6.2e-2	0.49	2.2e-8	20	
		app	0.17	0.23	1.8e-8	12	
	MGW 2	diagonal	6.6e-2	0.27	7.9e-9	10	
		lumped	6.2e-2	0.27	7.9e-9	10	
		app	0.17	0.13	4e-9	6	
	PD-GMRes	Diagonal	***	3.9e-2	1.15	1.3e-7	150
		Lumped	***	3.9e-2	2.1	0.95	200
ILUT(10^{-3})		***	0.11	6.2e-2	8.6e-9	4	
Projeção		diagonal	6.6e-2	0.21	4.4e-8	8	
		lumped	6.7e-2	0.19	3.5e-8	7	
		app	0.17	0.17	6.1e-8	4	
MGW 1		diagonal	6.6e-2	0.36	1e-7	15	
		lumped	6.2e-2	0.36	1e-7	15	
		app	0.17	0.16	6.2e-8	9	
MGW 2		diagonal	6.6e-2	0.21	4e-8	8	
		lumped	6.2e-2	0.21	4e-8	8	
		app	0.17	0.10	5.6e-9	5	
Bi-CGStab		Nenhum	***	3.8e-2	0.4	5.6e-7	197
		Diagonal	***	3.9e-2	0.37	1.6e-7	165
		Lumped	***	nc	nc	nc	nc
	ILUT(10^{-3})	***	0.11	4.8e-2	3.8e-10	3	
	Projeção	diagonal	6.6e-2	0.24	1.1e-8	6	
		lumped	6.6e-2	0.2	3.3e-8	5	
		app	0.17	0.17	1e-8	3	
	MGW 1	diagonal	6.6e-2	0.19	4.2e-8	5	
		lumped	6.2e-2	0.19	4.2e-8	5	
		app	0.17	0.08	1e-8	3	
	MGW 2	diagonal	6.6e-2	0.19	4.2e-8	5	
		lumped	6.2e-2	0.19	4.2e-8	5	
		app	0.17	0.08	4.8e-8	3	

Tabela 41: Degrau, acoplado com malha 20x10x3 e CFL=1

Solucionador	Precond	Aprox.	T.Prec	T.Sol	Erro	Iter	
PE-GMRes	Nenhum	***	0.23	8.1	5e-4	200	
	Diagonal	***	0.24	8.2	7.2e-2	200	
	Lumped	***	0.23	8.18	0.97	200	
	ILUT(10^{-3})	***	1.39	0.53	5.5e-11	6	
	Projeção	diagonal	0.47	1.81	3.4e-9	10	
		lumped	0.47	1.78	4.8e-9	9	
		app	3.61	2.19	1.5e-9	8	
	MGW 1	diagonal	0.47	3.45	5.2e-9	21	
		lumped	0.44	3.43	5.2e-9	21	
		app	3.62	3.01	8.1e-9	18	
	MGW 2	diagonal	0.47	1.79	8.7e-9	10	
		lumped	0.44	1.78	8.7e-9	10	
		app	3.61	1.55	5e-9	8	
	PD-GMRes	Diagonal	***	0.23	8.2	2.9e-2	200
		Lumped	***	0.23	8.2	0.99	200
ILUT(10^{-3})		***	1.39	0.42	5.1e-8	5	
Projeção		diagonal	0.47	1.37	2.8e-9	8	
		lumped	0.47	1.32	5.6e-9	7	
		app	3.61	1.52	1.6e-7	6	
MGW 1		diagonal	0.47	2.34	9.8e-9	15	
		lumped	0.44	2.35	9.8e-9	15	
		app	3.58	2.21	4.2e-8	14	
MGW 2		diagonal	0.47	1.21	1.5e-7	7	
		lumped	0.44	1.21	1.5e-7	7	
		app	3.58	1.05	9.2e-8	6	
Bi-CGStab		Nenhum	***	0.23	2.69	7.1e-3	199
		Diagonal	***	0.23	2.89	0.48	199
		Lumped	***	nc	nc	nc	nc
	ILUT(10^{-3})	***	1.39	0.37	6.5e-8	3	
	Projeção	diagonal	0.47	1.52	4.1e-9	5	
		lumped	0.47	1.19	1.4e-8	4	
		app	3.58	1.53	1e-8	4	
	MGW 1	diagonal	0.47	1.2	2.2e-7	4	
		lumped	0.44	1.2	2.2e-7	4	
		app	3.6	1.09	4.9e-7	2	
	MGW 2	diagonal	0.47	1.2	2.2e-7	4	
		lumped	0.44	1.2	2.2e-7	4	
		app	3.6	1.08	4.9e-7	4	

Tabela 42: Degrau, acoplado com malha 50x20x3 e CFL=1

Solucionador	Precond	Aprox.	T.Prec	T.Sol	Erro	Iter	
PE-GMRes	Nenhum	***	1.4	0.96	1.01	9	
	Diagonal	***	1.44	0.97	0.99	9	
	Lumped	***	1.49	0.93	0.99	9	
	ILUT(10^{-3})	***	35.5	5.81	1.8e-12	12	
	Projeção	diagonal		3.3	13.4	1.27e-8	10
		lumped		3.31	15.64	5.2e-9	10
		app		110	44.9	1.2e-8	14
	MGW 1	diagonal		3.3	28.6	1.8e-8	22
		lumped		3.13	28.6	1.8e-8	22
		app		110	78.4	4.2e-8	30
	MGW 2	diagonal		3.3	15.3	4.1e-9	11
		lumped		3.13	15.3	4.1e-9	11
		app		110	41.7	4e-8	15
	PD-GMRes	Diagonal	***	1.43	0.99	0.99	9
		Lumped	***	1.43	0.93	1.12	9
ILUT(10^{-3})		***	35.6	1.97	0.25	3	
Projeção		diagonal		3.3	8.85	9.9e-9	7
		lumped		3.31	10.4	1.6e-8	7
		app		110	33.7	1.6e-7	11
MGW 1		diagonal		3.29	17.6	6.3e-8	14
		lumped		3.13	17.6	6.3e-8	14
		app		110	58.7	1.3e-7	
MGW 2		diagonal		3.29	9.07	1.2e-7	7
		lumped		3.12	9.08	1.2e-7	7
		app		110	29.8	2.6e-7	11
Bi-CGStab	Nenhum	***	1.4	18.5	0.76	186	
	Diagonal	***	1.42	19.6	0.94	193	
	Lumped	***	nc	nc	nc	nc	
	ILUT(10^{-3})	***	35.6	4.66	1.1e-6	6	
	Projeção	diagonal		3.3	10.0	1.9e-9	5
		lumped		3.31	9.24	5.9e-9	4
		app		110	46.4	3.2e-7	8
	MGW 1	diagonal		3.29	10.3	5.6e-9	5
		lumped		3.12	10.3	5.6e-9	5
		app		110	41.6	3.2e-7	8
	MGW 2	diagonal		3.3	10.3	5.6e-9	5
		lumped		3.13	10.4	5.6e-9	5
app			110	41.8	3.2e-7	8	

Tabela 43: Caledônia, acoplado sem refinamento e CFL=1

Solucionador	Precond	Aprox.	T.Prec	T.Sol	Erro	Iter	
PE-GMRes	Nenhum	***	0.23	8.95	0.96	200	
	Diagonal	***	0.24	9.05	0.88	200	
	Lumped	***	0.24	9.04	0.99	200	
	ILUT(10^{-3})	***	3.62	1.19	2.3e-6	5	
	Projeção	diagonal		0.36	1.47	2.1e-5	6
		lumped		0.36	1.62	2.7e-5	6
		app		3.44	3.74	1.5e-6	15
	MGW 1	diagonal		0.36	2.74	2.2e-5	13
		lumped		0.33	2.73	2.2e-5	13
		app		3.43	5.02	8.5e-6	29
	MGW 2	diagonal		0.36	1.66	1e-5	7
		lumped		0.33	1.65	1e-5	7
		app		3.43	2.89	3.7e-6	15
	PD-GMRes	Diagonal	***	0.23	9.07	0.90	200
		Lumped	***	0.24	9.05	0.99	200
ILUT(10^{-3})		***	3.60	1.21	9e-8	6	
Projeção		diagonal		0.36	1.95	4.6e-6	10
		lumped		0.36	1.83	8.4e-6	8
		app		3.43	4.28	2.3e-6	17
MGW 1		diagonal		0.36	3.89	8e-6	20
		lumped		0.33	3.88	8e-6	20
		app		3.42	6.26	3.4e-6	35
MGW 2		diagonal		0.36	2.03	7.3e-6	10
		lumped		0.33	2.03	7.3e-6	10
		app		3.42	3.26	2.5e-6	18
Bi-CGStab		Nenhum	***	0.23	2.60	0.95	171
		Diagonal	***	nc	nc	nc	nc
		Lumped	***	nc	nc	nc	nc
	ILUT(10^{-3})	***	3.62	1.14	9.9e-9	4	
	Projeção	diagonal		0.36	2.41	8.2e-9	7
		lumped		0.35	2.01	8.8e-8	5
		app		3.42	5.33	1.3e-7	12
	MGW 1	diagonal		0.35	1.82	6.5e-8	5
		lumped		0.32	1.81	6.5e-8	5
		app		3.42	3.81	8e-8	12
	MGW 2	diagonal		0.35	1.82	6.5e-8	5
		lumped		0.32	1.81	6.5e-8	5
		app		3.42	3.85	8e-8	12

Tabela 44: Caledônia, acoplado com refinamento e CFL=1

Solucionador	Precond	Aprox.	T.Prec	T.Sol	Erro	Iter	
PE-GMRes	Nenhum	***	1.19	57.5	0.98	200	
	Diagonal	***	1.21	58.1	0.95	200	
	Lumped	***	1.21	58.1	0.99	200	
	ILUT(10^{-3})	***	64.5	8.83	6.7e-8	7	
	Projeção	diagonal		2.19	9.30	4.6e-6	8
		lumped		2.19	9.45	7.1e-6	7
		app		49.9	34.2	1.6e-6	22
	MGW 1	diagonal		2.17	16.2	6.6e-6	15
		lumped		2.04	16.2	6.6e-6	15
		app		49.9	53.9	1.2e-6	44
	MGW 2	diagonal		2.17	9.36	8e-6	8
		lumped		2.04	9.35	8e-6	8
		app		49.9	27.8	2.3e-6	22
	PD-GMRes	Diagonal	***	1.20	57.8	0.96	200
		Lumped	***	1.21	57.8	0.99	200
ILUT(10^{-3})		***	64.3	7.95	8.2e-8	7	
Projeção		diagonal		2.19	12.6	8.2e-7	12
		lumped		2.19	10.4	3.0e-6	9
		app		49.8	38.7	3.8e-7	26
MGW 1		diagonal		2.17	22.8	1.8e-6	22
		lumped		2.04	22.8	1.8e-6	22
		app		49.8	61.6	2.3e-7	51
MGW 2		diagonal		2.17	11.9	2.3e-6	11
		lumped		2.04	11.9	2.3e-6	11
		app		49.8	31.6	4.6e-7	26
Bi-CGStab	Nenhum	***	1.19	14.3	0.98	196	
	Diagonal	***	1.21	15.4	0.95	143	
	Lumped	***	nc	nc	nc	nc	
	ILUT(10^{-3})	***	64.4	7.69	1.0e-7	4	
	Projeção	diagonal		2.19	16.6	1.8e-9	9
		lumped		2.19	14.4	2.3e-8	7
		app		49.8	54.6	3.7e-7	19
	MGW 1	diagonal		2.17	13.5	5.0e-9	7
		lumped		2.04	13.5	5.0e-9	7
		app		49.8	43.6	2.9e-7	19
	MGW 2	diagonal		2.17	13.5	5.0e-9	7
		lumped		2.04	13.5	5.0e-9	7
app			49.8	43.6	2.9e-7	19	

APÊNDICE C – Tabelas de resultados com CFL=5

Neste Apêndice, pode-se encontrar as tabelas com todos os testes com CFL=5. Estas tabelas contêm resultados sobre tempo de construção dos preconditionadores, tempo de resolução dos sistemas lineares, erro relativo, e número de iterações.

Sempre que necessário, foram feitos comentários relevantes acerca dos resultados obtidos.

Tabela 45: Canal, segregado com malha $20 \times 10 \times 3$ e CFL=5

Solucionador	Aprox.	Precond	T.Prec	T.Sol	Erro	Iter
Projeção	diag	ICC & ILU	0.49	0.14	0.32	***
	lumped		0.49	0.13	3.1e-2	***
	app		3.74	0.24	0.5	***
ABLUF	diag		0.49	0.82	3.1e-3	***
	lumped		0.49	0.84	1.1e-5	***
	app		3.73	1.03	4.8e-2	***

Tabela 46: Canal, segregado com malha $50 \times 20 \times 3$ e CFL=5

Solucionador	Aprox.	Precond	T.Prec	T.Sol	Erro	Iter
Projeção	diag	ICC & ILU	3.59	1.08	0.14	***
	lumped		3.60	1.00	3.3e-2	***
	app		121	2.11	0.92	***
ABLUF	diag		3.59	6.32	6.2e-5	***
	lumped		3.60	6.03	1e-4	***
	app		121	11.0	0.71	***

Tabela 47: Degrau, segregado com malha $10 \times 5 \times 3$ e CFL=5

Solucionador	Aprox.	Precond	T.Prec	T.Sol	Erro	Iter
Projeção	diag	ICC & ILU	6.9e-2	2.3e-2	0.34	***
	lumped		7e-2	1.9e-2	3.3e-2	***
	app		0.19	0.10	3.6e-2	***
ABLUF	diag		7e-2	0.14	9.4e-3	***
	lumped		7e-2	0.13	1.8e-5	***
	app		0.18	0.21	6.7e-8	***

Tabela 48: Degrau, segregado com malha 20x10x3 e CFL=5

Solucionador	Aprox.	Precond	T.Prec	T.Sol	Erro	Iter
Projeção	diag	ICC & ILU	0.49	0.14	0.33	***
	lumped		0.49	0.12	3.4e-2	***
	app		3.74	0.24	0.49	***
ABLUF	diag		0.49	0.81	3.4e-3	***
	lumped		0.49	0.81	1.5e-5	***
	app		3.73	1.05	4.7e-2	***

Tabela 49: Degrau, segregado com malha 50x20x3 e CFL=5

Solucionador	Aprox.	Precond	T.Prec	T.Sol	Erro	Iter
Projeção	diag	ICC & ILU	6.22	1.4	0.13	***
	lumped		3.62	1.02	4.1e-2	***
	app		121	2.06	0.92	***
ABLUF	diag		3.59	6.5	7.4e-5	***
	lumped		3.6	6.06	9.9e-5	***
	app		121	11.0	0.71	***

Tabela 50: Caledônia, segregado sem refinamento e CFL=5

Solucionador	Aprox.	Precond	T.Prec	T.Sol	Erro	Iter
Projeção	diag	ICC & ILU	0.36	0.14	0.35	***
	lumped		0.36	0.14	4.8e-2	***
	app		3.40	0.25	1.11	***
ABLUF	diag		0.36	0.84	1.1e-2	***
	lumped		0.36	0.92	2.2e-3	***
	app		6.38	1.03	0.58	***

Tabela 51: Caledônia, segregado com refinamento e CFL=5

Solucionador	Aprox.	Precond	T.Prec	T.Sol	Erro	Iter
Projeção	diag	ICC & ILU	2.25	0.96	0.44	***
	lumped		2.26	0.94	8.6e-2	***
	app		51.1	1.00	0.93	***
ABLUF	diag		2.25	5.45	2.0e-2	***
	lumped		2.26	5.54	2.2e-3	***
	app		51.2	5.82	0.78	***

Tabela 52: Canal, acoplado com malha $20 \times 10 \times 3$ e CFL=5

Solucionador	Precond	Aprox.	T.Prec	T.Sol	Erro	Iter	
PE-GMRes	Nenhum	***	0.23	8.14	1.2e-4	200	
	Diagonal	***	0.23	8.20	2.2e-5	200	
	Lumped	***	0.23	8.20	0.91	200	
	ILUT(10^{-3})	***	7.66	2.61	4.5e-11	7	
	Projeção	diagonal	0.49	1.82	1.1e-8	10	
		lumped	0.49	1.96	7.5e-9	10	
		app	3.71	2.00	9.6e-9	7	
	MGW 1	diagonal	0.49	3.65	2e-8	22	
		lumped	0.46	3.65	2e-8	22	
		app	3.72	2.81	3e-8	16	
	MGW 2	diagonal	0.49	1.97	2.8e-8	11	
		lumped	0.46	1.96	2.8e-8	11	
		app	3.72	1.58	3.7e-9	8	
	PD-GMRes	Diagonal	***	0.23	8.19	1.3e-5	200
		Lumped	***	0.23	8.23	0.97	200
ILUT(10^{-3})		***	7.64	1.79	8.9e-8	5	
Projeção		diagonal	0.49	1.22	9e-8	7	
		lumped	0.49	1.32	6.5e-8	7	
		app	3.73	1.57	1.8e-8	6	
MGW 1		diagonal	0.49	2.39	6.7e-8	15	
		lumped	0.46	2.39	6.7e-8	15	
		app	3.73	2.02	1.5e-7	12	
MGW 2		diagonal	0.49	1.37	3.8e-8	8	
		lumped	0.46	1.37	3.8e-8	8	
		app	3.73	1.13	3.1e-8	6	
Bi-CGStab		Nenhum	***	0.23	2.69	3.8e-4	200
		Diagonal	***	0.23	2.90	2.7e-3	200
		Lumped	***	nc	nc	nc	nc
	ILUT(10^{-3})	***	7.63	1.46	1.7e-7	3	
	Projeção	diagonal	0.49	1.23	8.9e-8	4	
		lumped	0.49	1.17	1.4e-7	4	
		app	3.73	1.65	1.7e-8	4	
	MGW 1	diagonal	0.49	1.38	1.1e-8	5	
		lumped	0.46	1.38	1.1e-8	5	
		app	3.73	1.65	8.3e-9	4	
	MGW 2	diagonal	0.49	1.38	1.1e-8	5	
		lumped	0.46	1.38	1.1e-8	5	
		app	3.74	1.38	1.3e-9	4	

Tabela 53: Canal, acoplado com malha $50 \times 20 \times 3$ e CFL=5

Solucionador	Precond	Aprox.	T.Prec	T.Sol	Erro	Iter	
PE-GMRes	Nenhum	***	1.40	0.92	1.06	9	
	Diagonal	***	1.42	0.95	0.99	9	
	Lumped	***	1.43	0.95	1.19	9	
	ILUT(10^{-3})	***	242	32.3	8.3e-12	8	
	Projeção	diagonal		3.59	18.4	6e-10	15
		lumped		3.60	20.2	8e-10	15
		app		121	42.0	4.5e-9	13
	MGW 1	diagonal		3.58	35.0	5e-10	30
		lumped		3.41	35.0	5e-10	30
		app		121	70.3	1e-8	27
	MGW 2	diagonal		3.58	18.4	1.4e-9	15
		lumped		3.41	18.4	1.4e-9	15
		app		121	36.5	8.5e-9	13
	PD-GMRes	Diagonal	***	1.42	0.94	1.53	9
		Lumped	***	1.43	0.95	1.09	9
ILUT(10^{-3})		***	242	23.0	9.6e-9	6	
Projeção		diagonal		3.59	12.1	3e-8	10
		lumped		3.59	12.0	5.8e-8	9
		app		121	31.3	3.7e-8	10
MGW 1		diagonal		3.58	22.9	4e-8	20
		lumped		3.41	22.9	4e-8	20
		app		121	53.8	3.5e-8	21
MGW 2		diagonal		3.58	12.1	3.3e-8	10
		lumped		3.41	12.1	3.3e-8	10
		app		121	27.3	7.9e-8	10
Bi-CGStab	Nenhum	***	1.40	18.4	4.1e-2	196	
	Diagonal	***	1.42	19.5	0.88	200	
	Lumped	***	nc	nc	nc	nc	
	ILUT(10^{-3})	***	242	22.7	2.7e-9	4	
	Projeção	diagonal		3.59	12.2	3.4e-8	6
		lumped		3.60	13.4	1.14e-7	6
		app		121	41.4	7.7e-8	7
	MGW 1	diagonal		3.58	13.2	6.1e-9	6
		lumped		3.41	13.2	6.1e-9	6
		app		121	37.0	3.8e-7	7
	MGW 2	diagonal		3.58	13.2	6.1e-9	6
		lumped		3.41	13.2	6.1e-9	6
app			121	37.0	3.8e-7	7	

Tabela 54: Degrau, acoplado com malha 10x5x3 e CFL=5

Solucionador	Precond	Aprox.	T.Prec	T.Sol	Erro	Iter	
PE-GMRes	Nenhum	***	3.8e-2	1.9	2.3e-6	192	
	Diagonal	***	3.9e-2	0.81	7.5e-8	12.6	
	Lumped	***	3.9e-2	2.09	0.74	200	
	ILUT(10^{-3})	***	0.41	0.26	1.7e-10	6	
	Projeção	diagonal	6.9e-2	0.23	1.1e-8	8	
		lumped	7e-2	0.24	2.1e-8	8	
		app	0.18	0.25	9.1e-9	5	
	MGW 1	diagonal	6.9e-2	0.44	2.5e-8	17	
		lumped	6.5e-2	0.44	2.5e-8	17	
		app	0.18	0.22	1.6e-8	11	
	MGW 2	diagonal	7e-2	0.25	1.7e-8	9	
		lumped	6.5e-2	0.25	1.7e-8	9	
		app	0.18	0.22	1.6e-8	5	
	PD-GMRes	Diagonal	***	3.9e-2	0.67	6.2e-7	114
		Lumped	***	3.9e-2	2.1	0.9	200
ILUT(10^{-3})		***	0.41	0.17	7.1e-7	4	
Projeção		diagonal	6.9e-2	0.16	1.9e-7	6	
		lumped	7e-2	0.17	1.3e-7	6	
		app	0.18	0.17	9e-9	4	
MGW 1		diagonal	6.9e-2	0.32	1.3e-7	13	
		lumped	6.5e-2	0.32	1.3e-7	13	
		app	0.17	0.15	1.7e-7	8	
MGW 2		diagonal	6.9e-2	0.19	4.3e-8	7	
		lumped	6.5e-2	0.19	4.3e-8	7	
		app	0.18	0.08	6.7e-8	4	
Bi-CGStab	Nenhum	***	3.8e-2	0.4	1.3e-5	199	
	Diagonal	***	3.9e-2	0.26	5.3e-7	119	
	Lumped	***	nc	nc	nc	nc	
	ILUT(10^{-3})	***	0.41	0.12	1.1e-6	2	
	Projeção	diagonal	7e-2	0.17	2.3e-8	4	
		lumped	7e-2	0.13	5.4e-7	3	
		app	0.17	0.15	2.4e-8	2	
	MGW 1	diagonal	6.9e-2	0.15	1e-7	4	
		lumped	6.5e-2	0.15	1e-7	4	
		app	0.18	0.08	1.8e-8	3	
	MGW 2	diagonal	6.9e-2	0.15	1e-7	4	
		lumped	6.5e-2	0.15	1e-7	4	
app		0.18	0.08	1.8e-8	3		

Tabela 55: Degrau, acoplado com malha 20x10x3 e CFL=5

Solucionador	Precond	Aprox.	T.Prec	T.Sol	Erro	Iter	
PE-GMRes	Nenhum	***	0.22	8.07	1.1e-4	200	
	Diagonal	***	0.23	8.16	3.7e-5	200	
	Lumped	***	0.23	8.19	0.9	200	
	ILUT(10^{-3})	***	7.64	2.6	5.6e-11	7	
	Projeção	diagonal	0.49	1.77	4.8e-9	10	
		lumped	0.49	1.93	5.8e-9	10	
		app	3.73	1.99	1.2e-8	7	
	MGW 1	diagonal	0.49	3.58	1.1e-8	22	
		lumped	0.46	3.59	1.1e-8	22	
		app	3.72	2.86	3.4e-8	16	
	MGW 2	diagonal	0.49	1.95	1.3e-8	11	
		lumped	0.46	1.95	1.3e-8	11	
		app	3.72	1.44	3.5e-8	7	
	PD-GMRes	Diagonal	***	0.23	8.16	2.7e-5	200
		Lumped	***	0.23	8.18	0.97	200
ILUT(10^{-3})		***	7.64	1.79	1e-7	5	
Projeção		diagonal	0.49	1.2	7.7e-8	7	
		lumped	0.49	1.15	5e-7	6	
		app	3.72	1.59	3e-8	6	
MGW 1		diagonal	0.49	2.35	4e-8	15	
		lumped	0.46	2.35	4e-8	15	
		app	3.72	2.07	1.2e-7	12	
MGW 2		diagonal	0.49	1.36	7.1e-8	8	
		lumped	0.46	1.36	7.1e-8	8	
		app	3.73	1.15	2.8e-8	6	
Bi-CGStab		Nenhum	***	0.23	2.63	4.2e-5	199
		Diagonal	***	0.23	2.87	1e-2	192
		Lumped	***	nc	nc	nc	nc
	ILUT(10^{-3})	***	7.65	1.46	1.7e-7	3	
	Projeção	diagonal	0.49	1.21	2.6e-8	4	
		lumped	0.49	1.15	1.9e-7	4	
		app	3.72	1.66	1.7e-8	4	
	MGW 1	diagonal	0.49	1.35	8.8e-9	5	
		lumped	0.46	1.35	8.8e-9	5	
		app	3.73	1.41	3.4e-9	4	
	MGW 2	diagonal	0.49	1.35	8.8e-9	5	
		lumped	0.46	1.35	8.8e-9	5	
		app	3.73	1.38	3.4e-9	4	

Tabela 56: Degrau, acoplado com malha 50x20x3 e CFL=5

Solucionador	Precond	Aprox.	T.Prec	T.Sol	Erro	Iter	
PE-GMRes	Nenhum	***	1.4	0.92	1.05	9	
	Diagonal	***	1.42	0.98	0.99	9	
	Lumped	***	1.49	0.94	1.25	9	
	ILUT(10^{-3})	***	243	32.3	8e-12	8	
	Projeção	diagonal		3.59	18.5	1.2e-9	15
		lumped		3.59	20.1	8.1e-10	15
		app		121	44.4	4.8e-9	13
	MGW 1	diagonal		3.58	34.9	2.22e-9	30
		lumped		3.42	34.9	2.2e9	30
		app		121	75.3	1.6e-8	27
	MGW 2	diagonal		3.58	18.3	2.4e-9	15
		lumped		3.41	18.38	2.4e-9	15
		app		121	38.9	1.8e-8	13
	PD-GMRes	Diagonal	***	1.42	1.06	7.18	9
		Lumped	***	1.43	0.93	1.07	9
ILUT(10^{-3})		***	243	23	1e-8	6	
Projeção		diagonal		3.59	12.1	4e-8	10
		lumped		3.59	12.1	7.2e-8	9
		app		121	33.1	7.2e-8	10
MGW 1		diagonal		3.58	22.8	2.9e-8	20
		lumped		3.41	22.9	2.9e-8	20
		app		121	57.6	7.5e-8	21
MGW 2		diagonal		3.58	12	3.3e-8	10
		lumped		3.41	12.1	3.3e-8	10
		app		121	29.1	1.5e-7	10
Bi-CGStab	Nenhum	***	1.4	18.5	0.22	186	
	Diagonal	***	1.42	19.5	0.89	194	
	Lumped	***	nc	nc	nc	nc	
	ILUT(10^{-3})	***	243	19.6	1.2e-7	3	
	Projeção	diagonal		3.59	13.3	4.8e-9	6
		lumped		3.6	13.4	6.6e-8	6
		app		121	44.4	1.2e-6	7
	MGW 1	diagonal		3.59	12.2	2.9e-8	6
		lumped		3.42	12.2	2.9e-8	6
		app		121	39.7	5e-7	
	MGW 2	diagonal		3.58	12.2	2.9e-8	6
		lumped		3.42	12.2	2.9e-8	6
app			121	39.7	5e-7	7	

Tabela 57: Caledônia, acoplado sem refinamento e CFL=5

Solucionador	Precond	Aprox.	T.Prec	T.Sol	Erro	Iter	
PE-GMRes	Nenhum	***	0.23	8.92	0.87	200	
	Diagonal	***	0.23	9.02	0.88	200	
	Lumped	***	0.23	9.07	1.01	200	
	ILUT(10^{-3})	***	2.96	1.37	1.5e-6	7	
	Projeção	diagonal		0.36	1.26	7.9e-5	6
		lumped		0.36	1.44	7.9e-5	6
		app		3.38	2.74	2.3e-6	11
	MGW 1	diagonal		0.36	2.49	2.3e-4	13
		lumped		0.33	2.50	2.3e-4	13
		app		3.35	3.07	3.9e-5	21
	MGW 2	diagonal		0.36	1.53	1.6e-4	7
		lumped		0.33	1.53	1.6e-4	7
		app		3.35	1.95	3.1e-6	11
	PD-GMRes	Diagonal	***	0.23	9.01	0.87	200
		Lumped	***	0.23	9.05	1.00	200
ILUT(10^{-3})		***	2.96	1.24	1.6e-6	7	
Projeção		diagonal		0.36	1.80	2e-5	10
		lumped		0.36	1.48	7.8e-5	7
		app		3.35	2.90	7e-7	13
MGW 1		diagonal		0.36	3.43	3.2e-5	19
		lumped		0.33	3.43	3.2e-5	19
		app		3.35	4.17	9.1e-7	26
MGW 2		diagonal		0.36	1.77	4.3e-5	10
		lumped		0.33	1.77	4.3e-5	10
		app		3.35	2.16	8.5e-7	13
Bi-CGStab		Nenhum	***	0.23	2.57	0.89	164
		Diagonal	***	0.23	2.81	0.86	185
		Lumped	***	nc	nc	nc	nc
	ILUT(10^{-3})	***	2.96	1.03	3e-6		
	Projeção	diagonal		0.36	1.73	3.8e-8	6
		lumped		0.36	1.46	9.7e-7	4
		app		3.35	3.18	5.4e-8	9
	MGW 1	diagonal		0.36	2.11	1.3e-7	7
		lumped		0.33	2.12	1.3e-7	7
		app		3.35	2.21	1.4e-7	8
	MGW 2	diagonal		0.36	2.12	1.3e-7	7
		lumped		0.33	2.12	1.3e-7	7
		app		3.35	2.20	1.4e-7	8

Tabela 58: Caledônia, acoplado com refinamento e CFL=5

Solucionador	Precond	Aprox.	T.Prec	T.Sol	Erro	Iter	
PE-GMRes	Nenhum	***	1.19	57.9	0.94	200	
	Diagonal	***	1.21	58.1	0.92	200	
	Lumped	***	1.21	58.3	0.99	200	
	ILUT(10^{-3})	***	76.6	12.9	6.2e-7	9	
	Projeção	diagonal		2.25	12.4	3.6e-5	13
		lumped		2.26	13.5	9.2e-5	12
		app		51.2	28.3	7.2e-6	17
	MGW 1	diagonal		2.23	26.9	1.1e-4	27
		lumped		2.1	26.9	1.1e-4	27
		app		51.2	46.2	6.7e-6	35
	MGW 2	diagonal		2.24	12.7	3e-5	13
		lumped		2.10	12.8	3e-5	13
		app		51.2	23.6	9.9e-6	17
	PD-GMRes	Diagonal	***	1.21	58.0	0.92	
		Lumped	***	1.21	57.9	0.99	200
ILUT(10^{-3})		***	76.6	11.8	6.2e-7	9	
Projeção		diagonal		2.25	19.6	1.9e-5	20
		lumped		2.26	16.6	8.4e-5	16
		app		51.2	31.6	4.2e-7	20
MGW 1		diagonal		2.24	39.0	5.4e-5	38
		lumped		2.10	39.0	5.4e-5	38
		app		51.1	51.9	2.1e-6	40
MGW 2		diagonal		2.24	18.5	6.5e-5	19
		lumped		2.10	18.5	6.5e-5	19
		app		51.2	26.4	2.3e-6	20
Bi-CGStab		Nenhum	***	1.19	14.3	0.92	190
		Diagonal	***	1.21	15.4	0.97	134
		Lumped	***	nc	nc	nc	nc
	ILUT(10^{-3})	***	76.6	11.5	1.1e-6	5	
	Projeção	diagonal		2.25	269	4.6e-5	19
		lumped		2.26	315	1e-3	14
		app		51.2	43.5	5.8e-8	14
	MGW 1	diagonal		2.24	314	7e-6	80
		lumped		2.10	314	7e-6	80
		app		51.2	36.2	2e-7	14
	MGW 2	diagonal		2.24	314	7e-6	80
		lumped		2.10	314	7e-6	80
		app		51.2	36.2	2e-7	14

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)