

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CURSO DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

**RESOLUÇÃO DE SISTEMAS LINEARES GERADOS NA DISCRETIZAÇÃO DAS
EQUAÇÕES DE NAVIER-STOKES EM MALHAS DE VORONOI**

DISSERTAÇÃO SUBMETIDA À UNIVERSIDADE FEDERAL DE SANTA CATARINA
PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIA DA COMPUTAÇÃO

Viviana Cocco Mariani

Florianópolis, maio de 1997.

**RESOLUÇÃO DE SISTEMAS LINEARES GERADOS NA DISCRETIZAÇÃO DAS
EQUAÇÕES DE NAVIER-STOKES EM MALHAS DE VORONOI**

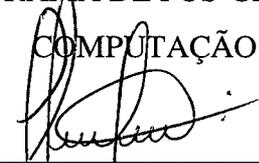
VIVIANA COCCO MARIANI

ESTA DISSERTAÇÃO FOI JULGADA ADEQUADA PARA A OBTENÇÃO DO TÍTULO DE

MESTRE EM CIÊNCIA DA COMPUTAÇÃO

ÁREA DE CONCENTRAÇÃO SISTEMAS DE COMPUTAÇÃO, E APROVADA EM SUA
FORMA FINAL PELO PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA

COMPUTAÇÃO

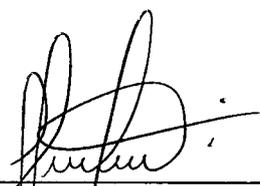


Prof. SÉRGIO PETERS, Dr. Eng. - ORIENTADOR

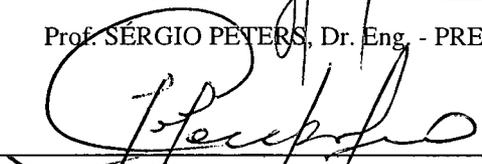


Prof. MURILO SILVA DE CAMARGO, Dr. - COORDENADOR

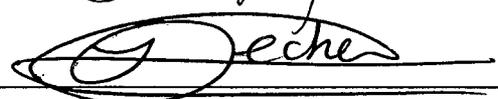
BANCA EXAMINADORA



Prof. SÉRGIO PETERS, Dr. Eng. - PRESIDENTE



Prof. JORGE MÁRIO CAMPAGNOLO, D. Sc.



Prof. ILDEMAR CASSANA DECKER, D. Sc.



Prof. JÚLIO FELIPE SZEREMETA, M. Sc.

Dedico este trabalho aos meus pais Valdemar e Carmelinda, daquela que vos ama muito.

Agradecimentos

Ao meu orientador e amigo Sérgio Peters pela valiosa orientação, incentivo, esclarecimentos e pelas várias contribuições a este trabalho.

Em especial, ao Leandro pelo amor e carinho, além do apoio, dedicação e paciência que tornaram possível esta caminhada.

Ao colega Fabian Corrêa Cardoso, pelo auxílio no desenvolvimento dos programas e pelas discussões constantes.

A todos colegas, professores e funcionários da UFSC, que consolidei laços de amizade e de companheirismo.

Ao professor Murilo, Verinha e Valdete pelo apoio.

Aos membros da banca examinadora pelas contribuições ao aprimoramento da dissertação.

À CAPES pelo apoio financeiro

A Deus, por ter permitido que eu chegasse até aqui.

Resumo

A grande maioria dos problemas em Mecânica dos Fluidos Computacional podem ser modelados por equações diferenciais parciais (EDPs). Utilizando métodos numéricos adequados, estas equações podem ser discretizadas em domínios geométricos de complexidades variadas através de técnicas da Mecânica dos Fluidos Computacional (“Computational Fluid Dynamics”). Discretizações estruturadas apresentam limitações quando aplicadas a problemas com geometrias complexas, neste caso, destacam-se os métodos alternativos com discretização em malhas não-estruturadas, como o método dos volumes finitos baseado na discretização com Diagramas de Voronoi (DV).

As malhas não-estruturadas de Voronoi dão origem a volumes de controle, cujas faces coincidem com o bissetor do segmento de ligação entre dois geradores vizinhos e são ortogonais a este segmento. Características que simplificam as aproximações numéricas existentes, facilitando a realização de balanços de massa, energia, e outros fluxos. Estas malhas são adotadas neste trabalho, para discretização dos domínios geométricos.

O presente trabalho aborda a descrição, análise e implementação de métodos iterativos da família do Gradiente Conjugado (GC), para a solução de sistemas lineares resultantes da discretização das equações de Navier-Stokes e da equação da condução de calor em malhas não-estruturadas de Voronoi, utilizando o método dos volumes finitos. Os métodos iterativos usados na solução do sistema linear são: GC, “Conjugate Gradient Squared”, “BiConjugate Gradient Stabilized” e “Transpose-Free Quasi Minimal Residual”.

No trabalho são descritas e adotadas três tipos de ordenações. Na verificação do desempenho são usados os métodos Gauss-Seidel, “Tridiagonal Matrix Algorithm” e GC. A velocidade de convergência, em especial, dos métodos da família do GC, pode ser melhorada com o uso de um pré-condicionador apropriado. Com este intuito são implementadas três técnicas de pré-condicionamentos: diagonal, SSOR e por fatoração de Cholesky incompleta. Os pré-condicionadores são aplicados no método GC e analisados de acordo com a distribuição dos autovalores.

Na discretização das equações de Navier-Stokes são utilizadas três formas distintas de avaliação dos gradientes de pressão: o mínimo resíduo quadrático, a média ponderada entre os gradientes normais projetados e a média ponderada modificada entre os gradientes normais projetados, proposta neste trabalho. Na validação dos gradientes de pressão e dos métodos iterativos, adota-se o cálculo do escoamento laminar incompressível em uma cavidade quadrada de profundidade infinita, sujeita a uma parede superior deslizando.

Abstract

There are various problems in the Computational Fluid Dynamics area that could be modeled by partial differential equations discretized in complex geometrical domains. Structured discretizations exhibit limitations when applied to the problems with these features, in this case, are outstanding the alternative methods with discretization in non-structured meshes, as the finite volume method based in Voronoi Diagrams (VD).

The Voronoi unstructured mesh originates control volumes, whose edge is part of the perpendicular bisector of the two generators, associated with the left and the right Voronoi regions, this edge is a line segment connecting two Voronoi vertexes. These features simplify the numerical existent approximations, easing the balance of mass, energy, etc. These meshes are adopted in this work, to discretization of the geometrical domains. One of the main advantages of the unstructured approach is the adjusting in specific region, to portray adequately the local flow, and also the points can be added and/or retired locally as the geometrical or flow features dictate.

In the present work, the iterative methods of the Conjugate Gradient (CG) family are described, analysed and implemented, for the solution of linear systems, resulting from the discretization of the Navier-Stokes equations and of the equation heat conduction in Voronoi of unstructured mesh, utilizing the finite volume method. The nonstationary iterative methods used in the solution of the linear system are: CG, Conjugate Gradient Squared, BiConjugate Gradient Stabilized and Transpose-Free Quasi Minimal Residual.

In this work are described and adopted three types of ordering, with objective of optimize the connections between nodal points. In the verification of performance is used the Gauss-Seidel, Tridiagonal Matrix Algorithm and CG methods. The convergence rate, in special, of the CG family methods can be improved using an adequate preconditioner, that decrease the condition number of the coefficient matrix. Three types of preconditioning are used, such as: diagonal, SSOR and incomplete Cholesky. The preconditioners are applied in the CG method and analysed through distribution of eigenvalues.

In the numerical discretization of the Navier-Stokes equations are utilized three different evaluations for pressure gradients: the squared residue minimum, the weighted mean between the standard gradients projected, and the modified weighted mean between the standard gradients projected, that is proposed in this work. The iterative methods and pressure gradients are validated, adopting the laminar flow incompressible in a square lid-driven cavity of infinite profundity, with slide upper wal.

Índice

Agradecimentos	iv
Resumo	v
Abstract	vi
Índice	vii
Lista de Figuras	ix
Lista de Tabelas	xiii
Simbologia	xiv
Capítulo 1 - Introdução	01
1.1 Preliminares	01
1.2 Métodos iterativos	04
1.3 Objetivo	05
1.4 Organização do trabalho	06
Capítulo 2 - Diagrama de Voronoi	08
2.1 Introdução	08
2.2 Definição e propriedades elementares	08
2.3 Técnicas de construção	10
2.3.1 Técnica dividir para conquistar - “divide and conquer”	11
2.3.2 Técnica de varredura - “swepline”	12
2.3.3 Técnica incremental	13
2.3.3.1 Gerador do diagrama de Voronoi pela técnica incremental	14
2.4 Estrutura computacional	15
2.5 Conclusão	18
Capítulo 3 - Método dos Volumes Finitos	19
3.1 Introdução	19
3.2 Características	19
3.3 Discretização em malha estruturada e não-estruturada	20
3.4 Disposição das variáveis dependentes na malha computacional	22
3.5 Equações governantes	23
3.5.1 Discretização das equações governantes	24
3.5.1.1 Discretização da equação da condução de calor	25
3.5.1.2 Discretização das equações de Navier-Stokes	26
3.6 Condições de contorno e apresentação das geometrias	36
3.7 Conclusão	37
Capítulo 4 - Métodos da família do Gradiente Conjugado	38
4.1 Introdução	38
4.2 Gradiente Conjugado	38
4.2.1 Convergência do Gradiente Conjugado	42
4.3 CGS	43
4.3.1 Convergência do CGS	44
4.4 Bi-CGSTAB	45
4.4.1 Convergência do Bi-CGSTAB	46
4.5 TFQMR	48

4.6 GMRES	50
4.6.1 Convergência do GMRES	52
4.7 Comparativo entre os métodos não-estacionários	52
4.8 Conclusão	53
Capítulo 5 - Pré-condicionamento	54
5.1 Introdução	54
5.2 Pré-condicionamento explícito e implícito	54
5.3 Pré-condicionamento diagonal	57
5.4 Pré-condicionamento por “Symmetric Successive Overrelaxation”	57
5.5 Pré-condicionamento por fatoração de Cholesky incompleta	60
5.5.1 Pré-condicionamento com “fill-ins”	63
5.6 Pré-condicionamento por série truncada de Maclaurin	64
5.7 Pré-condicionamento por série truncada de Neumann	65
5.8 Pré-condicionamento polinomial	66
5.9 Conclusão	67
Capítulo 6 - Ordenação	68
6.1 Introdução	68
6.2 Definições preliminares	68
6.3 Técnicas de Ordenação	69
6.3.1 Ordenação geométrica	70
6.3.2 Ordenação em linha contínua	71
6.4 TDMA	73
6.5 Gauss-Seidel	75
6.6 Análise da ordenação	76
6.7 Conclusão	84
Capítulo 7 - Resultados: Análise do GC com pré-condicionamento	85
7.1 Introdução	85
7.2 Geometrias	85
7.3 Análise da condição espectral	88
7.4 Efeitos do pré-condicionamento	91
7.5 Análise dos resultados	94
7.6 Conclusão	99
Capítulo 8 - Resultados numéricos: equações de Navier-Stokes	101
8.1 Introdução	101
8.2 Análise dos métodos da família do GC	101
8.3 Análise da correção da pressão	116
8.4 Análise das diferentes formas de avaliação dos gradientes de pressão	122
8.5 Conclusão	123
Capítulo 9 - Conclusões e perspectivas para futuros trabalhos	125
Referências Bibliográficas	130
Apêndice A - Glossário	A.1
Apêndice B - Equações e Propriedades do Gradiente Conjugado	B.1

Lista de Figuras

Fig. 2.1 - Polígono de Voronoi	09
Fig. 2.2 - Diagrama de Voronoi formado por seis polígonos convexos	10
Fig. 2.3 - União dos subdiagramas $DV(S_1)$ e $DV(S_2)$	12
Fig. 2.4 - Intersecção do diagrama de Voronoi com a varredura em linha horizontal	12
Fig. 2.5 - Adicionando um novo ponto p ao diagrama de Voronoi	13
Fig. 2.6 - Inconsistência topológica na geração de um polígono de Voronoi	14
Fig. 2.7 - Estrutura computacional do diagrama de Voronoi	16
Fig. 2.8 - Diagrama de Voronoi para 9 geradores	17
Fig. 2.9 - Armazenamento da matriz de coeficientes	17
Fig. 3.1 - Volume de controle elementar i,j para malha estruturada	20
Fig. 3.2 - Volume de controle elementar para malha não-estruturada de Voronoi	21
Fig. 3.3 - Volume de controle para u, v, T, P e ρ no arranjo co-localizado de variáveis	23
Fig. 3.4 - Fluxograma para resolução das equações de Navier-Stokes	35
Fig. 3.5.a - Domínio geométrico em formato quadrado e L	36
Fig. 3.5.b - Domínio geométrico em formato triangular	36
Fig. 4.1 - Algoritmo Gradiente Conjugado	40
Fig. 4.2 - Algoritmo CGS	44
Fig. 4.3 - Algoritmo Bi-CGSTAB	45
Fig. 4.4 - Algoritmo Bi-CGSTAB modificado	47
Fig. 4.5 - Algoritmo TFQMR	49
Fig. 4.6 - Algoritmo GMRES recomeçado	51
Fig. 5.1 - Representação do pré-condicionamento à direita e à esquerda	55
Fig. 5.2 - Gradiente Conjugado pré-condicionado implicitamente	56
Fig. 5.3 - Gradiente Conjugado pré-condicionado explicitamente	56
Fig. 5.4 - Algoritmo GCPCI	61
Fig. 6.1 - Grafo representativo da malha com nove pontos nodais.	69
Fig. 6.2 - Malha aleatória de Voronoi e matriz de coeficientes - ordenação vertical	71
Fig. 6.3 - Varredura em linha contínua, para a malha em formato L	72
Fig. 6.4 - Varredura horizontal em malha retangular	73
Fig. 6.5 - Algoritmo TDMA	75
Fig. 6.6 - Algoritmo Gauss-Seidel	76

Fig. 6.7 - Malha hexagonal formada por 2475 volumes	77
Fig. 6.8 - Malha retangular formada por 2401 volumes	77
Fig. 6.9 - Representação matricial (i, j) do <i>sistema a</i> e <i>sistema b</i> , (n = 2475)	78
Fig. 6.10 - Representação matricial (i, j) do <i>sistema c</i> , (n = 2401)	79
Fig. 6.11 - Autovalores para o <i>sistema a</i> e <i>b</i> com a matriz de coeficientes original	79
Fig. 6.12 - Autovalores para o <i>sistema c</i> com a matriz de coeficientes original	80
Fig. 6.13 - Convergência da temperatura no <i>sistema a</i>	81
Fig. 6.14 - Variação do critério de convergência no <i>sistema a</i>	81
Fig. 6.15 - Convergência da temperatura no <i>sistema b</i>	82
Fig. 6.16 - Variação do critério de convergência no <i>sistema b</i>	82
Fig. 6.17 - Convergência da temperatura, até a centésima iteração, no <i>sistema c</i>	83
Fig. 6.18 - Variação do critério de convergência no <i>sistema c</i> .	83
Fig. 7.1 - Malha aleatória com 1003 volumes, geometria em formato L	85
Fig. 7.2 - Malha hexagonal com 976 volumes, geometria triangular	86
Fig. 7.3 - Malha retangular com 1369 volumes, geometria quadrada	86
Fig. 7.4.a - Autovalores do <i>sistema 2</i> sem pré-condicionamento	89
Fig. 7.4.b - Autovalores do <i>sistema 2</i> com PD	89
Fig. 7.5 - Comparativo dos autovalores sem pré-condicionamento e com PD, <i>sistema 2</i>	90
Fig. 7.6.a - Autovalores do <i>sistema 3</i> sem pré-condicionamento	90
Fig. 7.6.b - Autovalores do <i>sistema 3</i> com PD	91
Fig. 7.7 - Comparativo dos autovalores sem pré-condicionamento e com PD, <i>sistema 3</i>	91
Fig. 7.8 - Representação matricial (i, j) do <i>sistema 1</i> , (n = 1003)	92
Fig. 7.9 - Representação matricial (i, j) do <i>sistema 2</i> , (n = 976)	93
Fig. 7.10 - Representação matricial (i, j) do <i>sistema 3</i> , (n = 1369)	93
Fig. 7.11 - Processo iterativo para a temperatura no <i>sistema 1</i>	94
Fig. 7.12 - Comportamento do resíduo pré-condicionado no <i>sistema 1</i>	94
Fig. 7.13 - Comportamento do resíduo pré-condicionado, até a centésima iteração	95
Fig. 7.14 - Temperatura nas primeiras iterações, <i>sistema 2</i>	95
Fig. 7.15 - Convergência da temperatura no <i>sistema 2</i> até a quinquagésima iteração	96
Fig. 7.16 - Comportamento do resíduo pré-condicionado, <i>sistema 2</i> .	96
Fig. 7.17 - Processo iterativo para a temperatura no <i>sistema 3</i>	97
Fig. 7.18 - Temperatura no <i>sistema 3</i> até a trigésima iteração	97
Fig. 7.19 - Comportamento do resíduo pré-condicionado, <i>sistema 3</i>	98

Fig. 7.20 - Resíduo pré-condicionado até a trigésima iteração, <i>sistema 3</i>	98
Fig. 7.21 - Convergência da temperatura utilizando o GCPSSOR	99
Fig. 8.1 - Cavidade quadrada com parede superior deslizante e condições de contorno	101
Fig. 8.2 - Malha aleatória com 530 volumes em geometria quadrada	102
Fig. 8.3 - Velocidade u com MRQ, malha retangular, (1369 pontos nodais)	104
Fig. 8.4 - Velocidade v com MRQ, malha retangular, (1369 pontos nodais)	105
Fig. 8.5 - Velocidade u com MPGNP, malha retangular, (1369 pontos nodais)	105
Fig. 8.6 - Velocidade v com MPGNP, malha retangular, (1369 pontos nodais)	105
Fig. 8.7 - Velocidade u com MPMGNP, malha retangular, (1369 pontos nodais)	106
Fig. 8.8 - Velocidade v com MPMGNP, malha retangular, (1369 pontos nodais)	106
Fig. 8.9 - Velocidade u com MRQ, malha hexagonal, (2475 pontos nodais)	107
Fig. 8.10 - Velocidade v com MRQ, malha hexagonal, (2475 pontos nodais)	107
Fig. 8.11 - Velocidade u com MPGNP, malha hexagonal, (2475 pontos nodais)	108
Fig. 8.12 - Velocidade v com MPGNP, malha hexagonal, (2475 pontos nodais)	108
Fig. 8.13 - Velocidade u com MPMGNP, malha hexagonal, (2475 pontos nodais)	108
Fig. 8.14 - Velocidade v com MPMGNP, malha hexagonal, (2475 pontos nodais)	109
Fig. 8.15 - Velocidade u com MRQ e MPMGNP na malha aleatória, (530 pontos nodais)	109
Fig. 8.16 - Velocidade v com MRQ e MPMGNP na malha aleatória, (530 pontos nodais)	110
Fig. 8.17 - Avaliação dos gradientes de pressão para a velocidade u com Reynolds = 100, malha retangular	110
Fig. 8.18 - Avaliação dos gradientes de pressão para a velocidade v com Reynolds = 100, malha retangular	111
Fig. 8.19 - Avaliação dos gradientes de pressão para a velocidade u com Reynolds = 100, malha hexagonal	111
Fig. 8.20 - Avaliação dos gradientes de pressão para a velocidade v com Reynolds = 100, malha hexagonal	112
Fig. 8.21 - Avaliação dos gradientes de pressão para a velocidade u com Reynolds = 400, malha retangular	113
Fig. 8.22 - Avaliação dos gradientes de pressão para a velocidade v com Reynolds = 400, malha retangular	113
Fig. 8.23 - Avaliação dos gradientes de pressão para a velocidade u com Reynolds = 400, malha hexagonal	114
Fig. 8.24 - Avaliação dos gradientes de pressão para a velocidade v com Reynolds = 400,	114

malha hexagonal

Fig. 8.25 - Avaliação dos gradientes de pressão para a velocidade u e v com Reynolds = 1000	115
Fig. 8.26 - Velocidade u obtida com o GC, utilizando MRQ, MPGNP e MPMGNP	118
Fig. 8.27 - Velocidade u até a iteração 300, com o MRQ, MPGNP e MPMGNP	118
Fig. 8.28 - Avaliação da convergência aplicando GC para o cálculo da velocidade u	119
Fig. 8.29 - Avaliação da convergência aplicando GC na velocidade u até a iteração 300	119

Lista de Tabelas

Tab. 1.1 - Características dos métodos diretos e iterativos	03
Tab. 2.1 - Disposição dos vizinhos no diagrama de Voronoi para 9 geradores	17
Tab. 3.1 - Representação de ϕ , S^ϕ e Γ^ϕ .	24
Tab. 3.2 - Condições de contorno.	37
Tab. 4.1 - Operações aproximadas dos métodos iterativos para a iteração k	52
Tab. 6.1 - Características dos sistemas resultantes nas ordenações	78
Tab. 6.2 - Resultados obtidos com o GS, TDMA e GC utilizando as ordenações	80
Tab. 7.1 - Descrição dos sistemas resultantes da discretização da equação do calor	87
Tab. 7.2 - Resultados do GC pré-condicionado	87
Tab. 7.3 - Autovalores das matrizes pré-condicionadas	92
Tab. 8.1 - Processo iterativo nas equações de Navier-Stokes com Reynolds = 100	103
Tab. 8.2 - Processo iterativo nas equações de Navier-Stokes com Reynolds = 400	112
Tab. 8.3 - Processo iterativo nas equações de Navier-Stokes com Reynolds = 1000	11??
Tab. 8.4 - Performance (iterações) do GC, CGS, Bi-CGSTAB e TFQMR com o MRQ, malha retangular	116
Tab. 8.5 - Performance (iterações) do GC, CGS, Bi-CGSTAB e TFQMR com o MPGNP e MPMGNP, malha retangular	117
Tab. 8.6 - Performance (iterações) do GC, CGS, Bi-CGSTAB e TFQMR com o MRQ, malha hexagonal	120
Tab. 8.7 - Performance (iterações) do GC, CGS, Bi-CGSTAB e TFQMR com o MPGNP, malha hexagonal	120
Tab. 8.8 - Performance (iterações) do GC, CGS, Bi-CGSTAB e TFQMR com o MPMGNP, malha hexagonal	121
Tab. 8.9 - Performance (iterações) do GC, CGS, Bi-CGSTAB e TFQMR com o MRQ, malha aleatória	121
Tab. 8.10 - Performance (iterações) do GC, CGS, Bi-CGSTAB e TFQMR com o MPMGNP, malha aleatória	122
Tab. 8.11 - Avaliação dos gradientes de pressão na malha retangular (1369 pontos nodais)	122
Tab. 8.12 - Avaliação dos gradientes de pressão na malha hexagonal (2475 pontos nodais)	123
Tab. 8.13 - Avaliação dos gradientes de pressão na malha aleatória (530 pontos nodais)	123

Simbologia

Símbolo	Significado
$(,)$	- produto escalar
$\ \ _{\infty}$	- norma infinita de um vetor ou matriz
$\ \ _2$	- norma segunda de um vetor ou matriz
A	- matriz de coeficientes
A^{-1}	- matriz inversa
A_{ij}	- coeficientes dos vizinhos do ponto nodal central p_i
AP_i	- coeficiente do ponto nodal central i
b	- vetor de termos independentes
$B(,)$	- bissetor
B_L	- matriz estritamente triangular inferior formada pelos elementos abaixo da $\text{diag}(A)$
B_U	- matriz estritamente triangular superior formada pelos elementos acima da $\text{diag}(A)$
C	- matriz de coeficientes do sistema transformado pelo pré-condicionamento
c_p	- calor específico
D	- matriz diagonal
$d(,)$	- distância Euclidiana
d_n	- superfície de integração normal
$d\mathcal{V}$	- derivada do volume de controle
e	- vetor diferença entre a solução esperada e a aproximada
E	- matriz erro
\in	- indica pertence a
e	- ponto este
E	- matriz da decomposição da matriz A no pré-condicionamento por Neumann
G	- matriz da decomposição da matriz A no pré-condicionamento por Neumann
G_k	- matriz iterativa que depende da matriz A
I	- matriz identidade
\vec{p}	- vetor base na direção y
J	- fluxo convectivo e difusivo de ϕ
l_{ij}	- elementos da matriz triangular inferior
L	- matriz triangular inferior (<i>lower</i>)
\underline{L}	- matriz triangular inferior aproximada
$L_{i,j}$	- distância entre o ponto nodal central do volume de controle em relação a cada ponto vizinho
L^T	- transposta da matriz triangular inferior
n	- ponto norte
N	- matriz auxiliar do método SSOR
N_k	- matriz do método iterativo que assume valores diferenciados em cada método
p	- pressão
p^k	- direção vetorial a ser seguida na iteração k (métodos da família do GC)
p_i	- ponto nodal central do Diagrama de Voronoi
P	- matriz de pré-condicionamento
P_D	- Pré-condicionador à direita
P_E	- Pré-condicionador à esquerda
q'''	- geração de calor
\Re	- conjunto dos números reais

\mathcal{R}^2	- plano Euclideano
$R(p_i)$	- região do Diagrama de Voronoi
r^k	- vetor resíduo (métodos da família do GC)
\tilde{r}_i	- vetor resíduo auxiliar
R	- matriz resíduo no cálculo do pré-condicionamento por fatoração de Cholesky
s	- ponto onde o bissetor entre p e p_i intercepta uma face do volume de controle de p_i
S	- conjunto finito de pontos no plano
s	- ponto sul
$S_{i,j}$	- distância entre dois vértices consecutivos no volume de controle
t	- ponto onde o bissetor entre p e p_i intercepta uma face do volume de controle de p_i
T	- temperatura
T_i	- temperatura do ponto nodal central
T_j	- temperatura de cada vizinho do ponto nodal central
u	- vetor independente do sistema pré-condicionado
U	- matriz triangular superior (<i>uper</i>)
\mathbf{U}	- matriz triangular superior aproximada
u	- componente da velocidade na direção x
u_i	- vetor direção
v	- vetor solução do sistema pré-condicionado
v_p	- componente da velocidade na direção y
\mathbf{V}	- vetor velocidade
V_i	- volume de controle do ponto nodal i
w	- ponto oeste
x	- vetor solução
x	- coordenada cartesiana
y	- coordenada cartesiana
z^k	- vetor

Gregos

Símbolo	Significado
α	- escalar (métodos GC, GCPDE, GCPDI)
β	- escalar (métodos GC, GCPDE, GCPDI)
∇	- operador vetorial <i>nabla</i>
ρ	- massa específica do fluido
ϕ	- variável genérica
μ	- viscosidade dinâmica do fluido
Γ^ϕ	- coeficiente de difusão de ϕ
ΔS	- área do volume de controle
$\Delta \mathcal{V}$	- volume do volume de controle
Δt	- variação no tempo
Δx	- variação da posição na direção x
ψ	- parâmetro de tolerância para o pré-condicionamento LU ou LL^T
ξ	- critério de parada
S^ϕ	- termo fonte
Δy	- variação da posição na direção y
ω	- coeficiente de relaxação

∇	- gradiente
φ	- funcional do Gradiente Conjugado
δ^k	- escalares do método GC que faz uso de três termos de recorrência
∂^k	- escalares do método GC que faz uso de três termos de recorrência
∂	- indica derivada parcial
Δ	- variação incremental de uma grandeza
Σ	- indica o somatório de elementos
Π	- indica o produtório de elementos

Sub e Superscritos

Símbolo	Significado
0	- índice que representa a iteração anterior
1	- índice que representa a iteração atual
i	- índice que indica o ponto nodal central
ij	- conexão entre os volumes i e j
j	- índice que indica os pontos nodais vizinhos
k	- índice que representa a iteração anterior
k+1	- índice que representa a iteração atual
NV	- número de vizinhos
T	- indica a transposta de uma matriz ou vetor

Abreviaturas

Símbolo	Significado
Bi-CG	- “BiConjugate Gradient”
Bi-CGSTAB	- “BiConjugate Gradient Stabilized”
GC	- Gradiente Conjugado
GCPDE	- Gradiente Conjugado com pré-condicionamento diagonal explícito
GCPDI	- Gradiente Conjugado com pré-condicionamento diagonal implícito
CGS	- “Conjugate Gradient Squared”
CPU	- “Central Processing Unit”
DV	- Diagrama de Voronoi
EDPs	- Equações Diferenciais Parciais
GMRES	- “Generalized Minimal Residual”
GS	- Gauss-Seidel
IC(0)	- fatoração de Cholesky incompleta
GCIC(0)	- GC com fatoração de Cholesky incompleta com 0 diagonais extras
MDF	- Método das Diferenças Finitas
MEF	- Método dos Elementos Finitos
MVF	- Método dos Volumes Finitos
MPGNP	- Média Ponderada entre os Gradientes Normais Projetados
MPMGNP	- Média Ponderada Modificada entre os Gradientes Normais Projetados
MRQ	- Mínimo Resíduo Quadrático
SSOR	- “Symmetric Successive Overrelaxation”
TDMA	- “Tridiagonal Matrix Algorithm”
TFQMR	- “Transpose-Free Quasi Minimal Residual”

Capítulo 1 - Introdução

1.1 Preliminares

A solução numérica de problemas em Mecânica dos Fluidos e Transferência de Calor, denominada como “Computational Fluid Dynamics” (CFD) corresponde a uma etapa importante no projeto da geometria de escoamentos. A simulação numérica de problemas complexos de engenharia, física e matemática é uma realidade, devido ao desenvolvimento de computadores de alta velocidade e com grande capacidade de armazenamento. Por esta razão analistas numéricos dedicam-se ao desenvolvimento de algoritmos eficientes, que solucionem diversos problemas de engenharia.

Os métodos numéricos para a solução de problemas envolvendo escoamento de fluidos em geometrias regulares estão bem desenvolvidos atualmente, porém na maioria das aplicações em engenharia as geometrias encontradas são complexas (Ronzani e Nieckele, 1995). Estimulados pelo desenvolvimento de vários métodos de geração de malhas não-estruturadas, que se ajustam aos mais variados formatos de contornos, novos algoritmos passaram a ser desenvolvidos usando um sistema de coordenadas localmente ortogonais.

Tais algoritmos devem permitir o uso de malhas adaptáveis a geometria do problema a ser solucionado. Neste contexto destacam-se as malhas não-estruturadas, como é o caso das malhas não-estruturadas geradas pelos Diagramas de Voronoi (DV), que são independentes de qualquer coordenada global (Taniguchi *et al.*, 1991). O DV é uma das principais construções da Geometria Computacional (Preparata and Shamos, 1985). Aplicações dos diagramas são encontradas em diversas áreas conforme relatado em Aurenhammer (1991):

- i) otimização de construções geométricas;
- ii) reconhecimento de padrões;
- iii) meteorologia;
- iv) crescimento de cristais;
- v) robótica.

Em especial, os DV são aplicados na área da CFD (Weatherill, 1992; Marcondes, 1996; Cardoso, 1997) sendo um campo de pesquisa emergente e atual. Detalhes sobre os DV serão apresentados no capítulo 2.

Neste trabalho propõe-se a solução numérica de um problema de escoamento de fluidos modelado em coordenadas cartesianas. A técnica de volumes finitos é usada para discretização

do domínio, com uma configuração de malha co-localizada para todas as variáveis. O processo de discretização das equações diferenciais pelo método dos volumes finitos (MVF), consiste em dividir o domínio físico em um número de volumes de controle não sobrepostos (Patankar, 1980), onde promovem-se balanços locais das grandezas físicas envolvidas. O MVF assim como outros métodos numéricos visa:

- (i) solução de escoamento sobre as geometrias arbitrárias;
- (ii) conservação local das propriedades do fluido;
- (iii) economia de tempo computacional.

A solução numérica dos problemas de escoamentos citados pode ser obtida pela discretização numérica das equações diferenciais parciais (EDPs), que compõem o modelo matemático do problema físico. Nesta discretização as equações diferenciais são integradas sobre cada volume de controle discreto, são avaliados os fluxos convectivos e difusivos através das faces dos volumes utilizando-se as variáveis nodais armazenadas no centro de cada volume de controle. Neste processo, a equação discretizada contém os valores das variáveis dependentes em um conjunto finito de pontos. Essa discretização da variável dependente é o que torna possível representar as equações diferenciais por equações algébricas discretas (Masliska, 1995; Tanyi and Thatcher, 1996).

Quando as EDPs são não-lineares, após a discretização, gera-se um sistema de equações algébricas também não-lineares, que são resolvidas a partir da linearização. A resolução dos sistemas de equações algébricas lineares é a etapa que, geralmente, envolve um grande custo em processamento computacional. Os sistemas lineares, provenientes de malhas não-estruturadas, são geralmente esparsos e sem lei de formação, exigindo algoritmos de resolução robustos, no sentido de prever as diversas conectividades possíveis. Por outro lado, malhas não-estruturadas conseguem capturar com facilidade geometrias complexas (Marcondes *et al.*, 1995).

Geralmente o sistema de equações algébricas lineares pode ser expresso como,

$$Ax = b \quad (1.1)$$

onde a matriz A de ordem $n \times n$, gerada da discretização em um volume de controle é positiva definida e simétrica para a equação da condução de calor e é positiva definida e assimétrica para as equações de Navier-Stokes, b é o vetor dos termos independentes e x é o vetor de incógnitas, sendo A diagonalmente dominante em ambas equações.

O sistema resultante após a discretização numérica da equação da condução de calor, usada no presente trabalho, é linear, contudo na discretização das equações de Navier-Stokes o sistema resultante é não-linear, cujas soluções são obtidas por linearização iterativa. Desta forma, a solução do sistema linear representa uma etapa intermediária do processo de linearização, cujo custo computacional para encontrá-la não deve ser muito alto (Papadrakakis, 1993).

Os métodos usados para resolver os sistemas de equações lineares podem ser diretos e iterativos. Nos métodos diretos a matriz de coeficientes é normalmente fatorada, e os fatores são usados para resolver o sistema de equações. Vastamente usada é a decomposição $A=LDL^T$ da matriz simétrica de coeficientes e, a decomposição $A=LU$ quando a matriz é assimétrica. A maior razão para utilização destes métodos é o fato de que a solução, para um dado problema, é obtida em um número finito de passos, que podem ser definidos a priori (Angeleri *et al.*, 1989).

Uma desvantagem associada ao uso dos métodos diretos, para matrizes esparsas, é que os fatores L e U apresentam um número maior de elementos não nulos que a matriz original de coeficientes, devido ao processo de enchimento-“fill-ins”¹ durante o processo de fatoração (Brussino and Sonnad, 1989). Logo, a armazenagem requerida pelos fatores pode ser maior, exigindo tempo de processamento adicional. Algumas características dos métodos diretos e iterativos são descritas na tabela 1.1.

Tabela 1.1 - Características dos métodos diretos e iterativos.

Características	métodos diretos	métodos iterativos
operações	sequência recursiva	sequência iterativa
matriz de coeficientes	é modificada gerando “fill-ins” e erros de arredondamento	é preservada
número de operações	finito	dependem de um contador ou critério de parada
aplicado em sistemas	de médio porte	de grandes dimensões e esparsos
melhorar a performance	através de ordenações nos elementos da matriz	com coeficiente de relaxação apropriado, pré-condicionamentos e ordenações das variáveis

Em função das características apresentadas na tabela 1.1, tem crescido o interesse pelos métodos iterativos, devido ao desenvolvimento crescente de técnicas eficientes de aceleração,

¹ “Fill-ins” ocorrem quando elementos da matriz original de coeficientes, inicialmente nulos, passam a ser não-nulos após a fatoração, conforme Brussino e Sonnad (1989).

conhecidas como *pré-condicionadores*, que podem melhorar a velocidade de convergência encontrando a solução almejada em menor número de iterações e tempo de processamento (Angeleri *et al.*, 1989), e também devido a utilização de computadores paralelo, onde os softwares, empregando métodos iterativos, escritos para os computadores paralelos permitem aumentar a quantidade de processos realizados em um determinado período de tempo através da divisão de uma tarefa de computação entre os diversos processadores.

1.2 Métodos iterativos

Nos métodos iterativos² típicos, inicia-se o processo com um valor inicial para o vetor de incógnitas, que é refinado em estágios sucessivos até que satisfaça as equações originais do sistema, de acordo com um nível de exatidão especificado (Barrett *et al.*, 1994). O esforço computacional exigido para que cada equação seja resolvida, depende da velocidade de convergência e da precisão adotada.

Os métodos iterativos tornam-se inadequados em aplicações práticas se a velocidade de convergência for lenta, levando a um tempo de processamento excessivo. Contudo, uma convergência lenta, frequentemente, pode ser melhorada pela técnica conhecida como *pré-condicionamento*, onde o sistema original de equações é modificado pela pré-multiplicação por uma matriz, que é escolhida de tal forma que o sistema equivalente obtenha uma matriz de coeficientes próxima da identidade. O desenvolvimento de eficientes pré-condicionadores tem sido um dos atrativos para o interesse crescente nos métodos iterativos (Axelsson, 1985; Angeleri *et al.*, 1989; Campos, 1995; Jones and Plassmann, 1995; Saint-Georges *et al.*, 1996). No capítulo 5 são descritos alguns pré-condicionadores encontrados na literatura, e no capítulo 7 são apresentados os resultados obtidos com a aplicação dos pré-condicionadores na solução da equação da condução de calor

Um dos métodos iterativos, mais populares, para resolver a equação (1.1) é o método denominado gradiente conjugado (GC). O GC foi introduzido nos anos 50 por Hestenes e Stiefel (Golub and Van Loan, 1996; Golub and O'Leary, 1989). O método GC resolve sistemas lineares, onde a matriz A é positiva definida e simétrica, contudo não tem garantia de convergência para

² Os métodos iterativos podem ser classificados em estacionários e não-estacionários. Neste trabalho são utilizados os métodos não-estacionários para resolução dos sistemas lineares.

matrizes assimétricas e não positiva definida (Campos., 1995). Este método possui uma forma elegante de tratar a esparsidade da matriz A , uma vez que necessita somente da avaliação do produto vetor-matriz através do processo iterativo. Com o uso dos pré-condicionadores o GC tornou-se um método competitivo e atrativo para resolver sistemas lineares que surgem da discretização em elementos finitos e volumes finitos (Coutinho, 1991; Saint-Georges *et al.*, 1996)

Na resolução das matrizes assimétricas, que surgem da discretização das equações de Navier-Stokes, usando as variáveis primitivas velocidade e pressão, exploram-se os métodos: Gradiente Conjugado, “Conjugate Gradients Squared” (CGS) (Sonneveld, 1989), “BiConjugate Gradient Stabilized” (Bi-CGSTAB) (Van der Vorst, 1992), e “Transpose-Free Quasi-Minimal Residual” (TFQMR). Recentes estudos indicam que o CGS é tão competitivo quanto o GMRES, quanto a velocidade de convergência, economia na memória de armazenamento de variáveis e tempo de processamento, porém em muitas situações o CGS tem convergência irregular, podendo encontrar uma solução inconsistente .

O Bi-CGSTAB pode apresentar convergência irregular e até mesmo mais lenta que o CGS para problemas complexos. O TFQMR é definido pela minimização parcial da norma do resíduo, o que pode remediar o comportamento de convergência irregular encontrado com o CGS (Lin *et al.*, 1995). Cabe ressaltar que estes métodos são descritos no capítulo 4 do presente trabalho.

Presentemente, algumas ordenações das variáveis na malha computacional são exploradas, utilizando as características intrínsecas da malha geométrica. Os métodos GC, “Tridiagonal Matrix Algorithm” (TDMA) e Gauss-Seidel (GS) são aplicados na solução do sistema linear previamente ordenado. Tais métodos são frequentemente empregados na área de Mecânica dos Fluidos Computacional (Tanyi and Thatcher, 1996; Cardoso, 1997).

1.3 Objetivo

O objetivo desta dissertação é reunir e analisar alguns dos métodos iterativos, considerados representativos do universo utilizado hoje, para a solução dos sistemas de equações lineares algébricas. Os sistemas lineares apresentados, são resultantes da discretização das equações de Navier-Stokes e da equação da condução de calor, empregando malhas não-estruturadas geradas por DV (Taniguchi, *et al.*, 1991; Marcondes, 1996).

Os métodos usados para a solução do sistema de equações são: GC, CGS, Bi-CGSTAB, TFQMR, GS e TDMA. Os pré-condicionadores, que têm como principal função acelerar a velocidade de convergência dos métodos iterativos, são encontrados em grande número na literatura, contudo neste trabalho serão aplicados apenas alguns: Diagonal, SSOR e fatoração de Cholesky incompleta.

Não pretende-se com isto cobrir todas as alternativas de métodos iterativos, ordenações e pré-condicionadores encontrados na literatura, nem mesmo dar uma comparação que permita a escolha do melhor método entre eles, mas o que objetiva-se é expor algumas alternativas de tal forma que o leitor possa escolher o caminho que pretende seguir, a fim de solucionar os seus problemas de interesse.

1.4 Organização do trabalho

No presente capítulo, procurou-se mostrar a relevância deste trabalho e os principais objetivos almejados. O capítulo 2 apresenta as definições básicas, as propriedades fundamentais, as técnicas de construção do gerador do DV e a estrutura computacional do gerador do DV, que é utilizada para armazenar os elementos da matriz de coeficientes.

No capítulo 3, face ao problema físico a ser resolvido, propõe-se o uso do MVF para a discretização numérica das equações de Navier-Stokes e da condução de calor transformando-as em sistema de equações algébricas lineares, através de um processo de integração aproximada.

Existem dois tipos de discretização do domínio físico: por malha estruturada e não-estruturada. Neste trabalho é empregado o segundo tipo, cujas vantagens e desvantagens são apresentadas. Adota-se o armazenamento das variáveis envolvidas, no centro do volume de controle a ser discretizado e são descritas as condições de contorno das geometrias utilizadas.

No capítulo 4 são considerados os métodos iterativos não-estacionários utilizados no trabalho, são apresentados os algoritmos e uma breve descrição do comportamento de convergência. Diversas técnicas de pré-condicionamentos são apresentadas no capítulo 5, e três delas são consideradas para a análise da performance dos métodos.

No capítulo 6 são apresentadas duas ordenações para as variáveis da malha computacional, aplicando-se os métodos GS, TDMA e GC nas equações ordenadas. A análise dos resultados obtidos na solução da equação da condução de calor, aplicando os diversos pré-condicionadores com o método GC é apresentada no capítulo 7.

O capítulo 8 reúne os diversos resultados encontrados na solução das equações de Navier-Stokes com os métodos não-estacionários para sistemas assimétricos, utilizando número de Reynolds variando de 100 a 1000, e três formas diferentes de avaliação dos gradientes de pressão. As conclusões e perspectivas para trabalhos futuros são apresentadas no capítulo 9, e logo a seguir expõe-se a bibliografia utilizada. No apêndice A encontra-se um glossário com os diversos termos encontrados no decorrer do trabalho, dedica-se este apêndice ao leitor não familiarizado com a área. No apêndice B, são demonstradas matematicamente, a equivalência entre as fórmulas matemáticas alternativas para o cálculo do método GC.

Capítulo 2 - Diagramas de Voronoi

2.1 Introdução

Neste capítulo procura-se enfatizar à exposição e análise unificada das propriedades matemáticas e computacionais do DV. Uma pesquisa das técnicas de construção para geração dos DV é descrita, e é apresentado o gerador do DV adotado. Os elementos da matriz de coeficientes, equação (1.1), são gerados e armazenados segundo a estrutura de geração do DV, e esta estrutura é descrita na última seção.

2.2 Definição e propriedades elementares

O pesquisador Dirichlet, em 1850, propôs um método onde, um dado domínio pode ser sistematicamente decomposto em um conjunto de polígonos convexos. Dados dois pontos no plano, p_i e p_j , o bissetor perpendicular da linha unindo os dois pontos subdivide o plano em duas regiões $R(p_i)$ e $R(p_j)$. A região $R(p_i)$ é a região mais próxima de p_i que de p_j .

Estendendo estas idéias para um conjunto de pontos no plano, as regiões $R(p_i)$ são os territórios que devem ser atribuídos para cada ponto, tal que $R(p_i)$ representa o espaço mais próximo de p_i de qualquer outro ponto no conjunto. Esta construção geométrica de volumes é conhecida como: “tessellation of Dirichlet”, polígonos de Thiessen e células ou domínios de Wigner-Seitz (Maliska Jr., 1993).

Os polígonos convexos não sobrepostos, que cobrem todo o domínio, são também chamados de *regiões de Voronoi*. Esta definição pode ser estendida para dimensões maiores, por exemplo, para três dimensões, as regiões de Voronoi são poliedros convexos.

Seja $S = \{ p_1, \dots, p_n \}$ um conjunto de pontos distintos e não-colineares no plano \mathfrak{R}^2 e seja $d(p, p_i)$ a distância Euclideana entre dois pontos p e p_i , A região de Voronoi $R(p_i)$ gerada pelo ponto p_i é definida por:

$$R(p_i) = \{ p \in \mathfrak{R}^2; d(p, p_i) < d(p, p_j), \forall j \neq i \} \quad (2.1)$$

Note que a região $R(p_i)$, não pode ser vazia, pois ela contém todos os pontos p que estão mais próximos de p_i do que de qualquer outro ponto, em particular, p_i pertence a $R(p_i)$. As regiões de Voronoi também são denominadas de polígonos de Voronoi formados por não mais que $(n-1)$ lados. O conjunto de regiões de Voronoi $\{R(p_1), R(p_2), \dots, R(p_n)\}$ particiona o plano e forma o DV para S . Os elementos p_i , pertencentes a S , são chamados geradores de Voronoi (Preparata and Shamos, 1985; Weatherill, 1992).

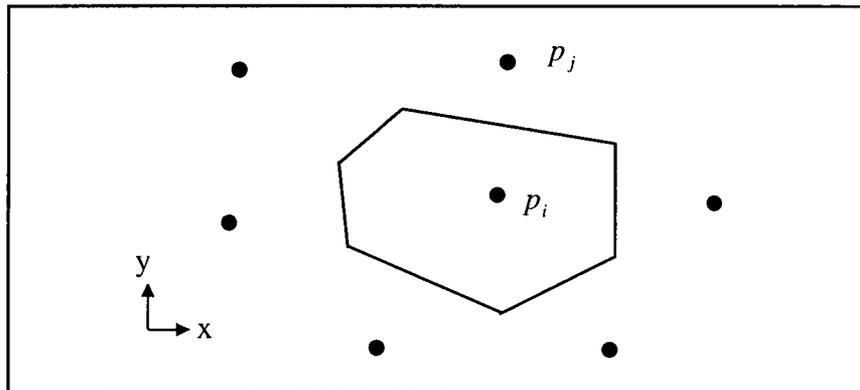


Figura 2.1 - Polígono de Voronoi.

Uma fronteira comum à duas regiões de Voronoi $R(p_i)$ e $R(p_j)$ é chamada face de Voronoi. Os pontos onde duas ou mais faces de Voronoi se encontram são denominados vértices de Voronoi, conseqüentemente o limite de uma região consiste de no máximo $(n-1)$ faces e vértices. No presente trabalho, a discretização numérica do domínio computacional é feita do modo que o domínio seja particionado em volumes de controle, representados por um ponto nodal gerador. As seguintes propriedades são conseqüências diretas da definição do DV (mostradas na figura 2.2):

- (i) a reta que une o ponto i ao seu vizinho j é sempre ortogonal a aresta comum a estes dois pontos, por exemplo \overline{ij} é perpendicular a \overline{ae} para $j=5$.
- (ii) a face comum ao ponto i e seu vizinho j , ou o seu prolongamento, está sobre a mediatriz da reta \overline{ij} que une estes dois pontos. Por exemplo, a reta \overline{de} passa no ponto médio da reta d_{i4} , ou seja, passa no ponto médio da reta \overline{ij} para $j=4$, isto é, cada ponto na face está equidistante de exatamente dois pontos.
- (iii) qualquer vértice de Voronoi é o circuncentro de três ou mais pontos geradores pertencentes a S . Por exemplo, o vértice d é o centro do círculo que passa pelos pontos $i, j=3, j=4$, isto é, cada vértice é equidistante de no mínimo três pontos.

(iv) a região de Voronoi $R(p_i)$ é um polígono convexo contendo o gerador correspondente p_i no seu interior.

(v) os polígonos de Voronoi particionam o plano Euclidiano.

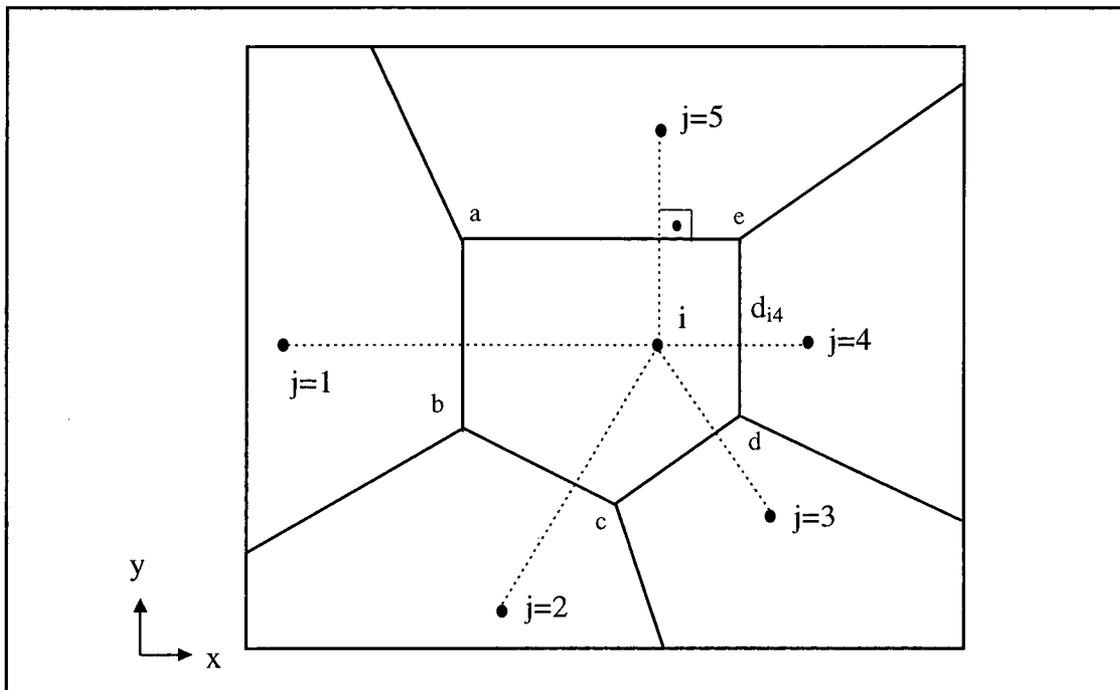


Figura 2.2 - Diagrama de Voronoi formado por seis polígonos convexos.

A literatura pertinente apresenta descrições detalhadas e informações adicionais sobre os DV, como exemplos pode-se citar Aurenhammer (1991), Sugihara e Iri (1992). Vem crescendo o número de áreas com interesse nos DV, em parte devido a suas propriedades. Verifica-se, nas últimas décadas, a utilização dos DV como uma ferramenta que possibilita o planejamento de trajetórias de robôs móveis, evitando colisões em obstáculos. Os obstáculos podem ser considerados como os geradores do DV, isto é, seus pontos nodais. As arestas de Voronoi podem ser interpretadas como o mapa de trajetórias. Considerando a presença de um robô tipo-ponto no espaço de trabalho, é fácil verificar que as arestas representam caminhos livres de colisão por onde o robô pode navegar (Canny and Donald, 1988; Roque, 1996).

2.3 Técnicas de construção

Nesta seção revisa-se as técnicas de construção fundamentais conhecidas para a geração computacional e representação de DV. O DV foi usado por décadas pelos cientistas, onde muitas

regras de construção intuitivas foram propostas, todavia o mais primitivo diagrama foi desenhado com lápis e régua com problemas de ambigüidade se muitos pontos surgissem em um mesmo círculo (Aurenhammer, 1991).

2.3.1 Técnica dividir para conquistar - “divide-and-conquer”

Considerando o problema do par mais próximo em uma dimensão, nota-se a existência de uma simples técnica para resolvê-lo eficientemente, para isto basta ordenar os pontos dados e procurar o par mais próximo, através de uma varredura linear destes pontos. Observa-se que o par de pontos mais próximo é necessariamente um par consecutivo no conjunto ordenado. A necessidade de generalização da procura de um par mais próximo para dimensões maiores leva ao surgimento do paradigma de *dividir para conquistar* (“divide-and-conquer”).

Entre as técnicas para a construção dos DV, a técnica de *dividir para conquistar* é a mais relevante na ciência da computação e na geometria computacional, proposta por Shamos e Hoey (Preparata and Shamos, 1985; Roos, 1991). A característica principal da técnica é a divisão do problema em subproblemas, que são versões pequenas do problema original, que podem ser combinados para uma solução global (Roos, 1991).

A aplicação desta técnica para construir o DV no plano Euclideano, separa o conjunto de pontos S em dois subconjuntos S_1 e S_2 , cada subconjunto tendo $n/2$ pontos e obtém-se o par mais próximo em cada um. O próximo passo é o cálculo do diagrama $DV(S_1)$ e $DV(S_2)$ dos subproblemas, onde a tarefa complexa é fundir os diagramas $DV(S_1)$ e $DV(S_2)$ dentro do diagrama global $DV(S_1 \cup S_2)$. Isto pode ser feito pelo cálculo da curva de bissecção,

$$B(S_1, S_2) = \{x \in \mathfrak{R}^2 \mid \min_{p_i \in S_1} d(x, p_i) = \min_{p_j \in S_2} d(x, p_j)\} \quad (2.2)$$

e pelo fechamento dos sub-diagramas nesta curva de bissecção, reporte-se na figura 2.3.

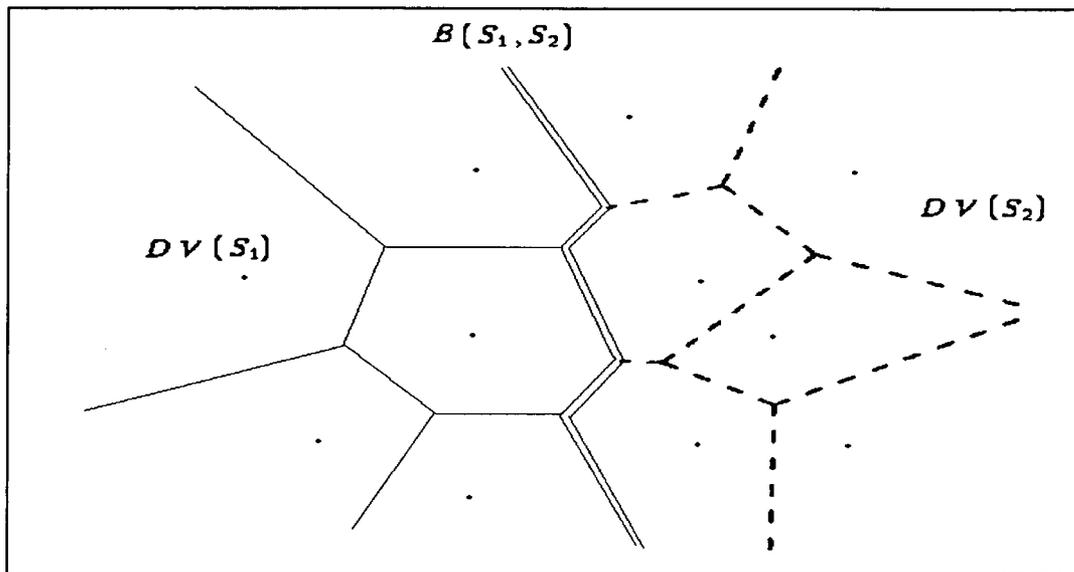


Figura 2.3 - União dos subdiagramas $DV(S_1)$ e $DV(S_2)$.

2.3.2 Técnica de varredura - “sweepline”

Segundo Fortune (1987), os algoritmos de varredura são frequentemente algoritmos incrementais simples, evitando a dificuldade na união dos passos. Para aplicar esta técnica move-se uma linha horizontal (ou vertical) no domínio geométrico mantendo, contudo, a interseção ordenada com as faces de Voronoi, conforme figura 2.4.

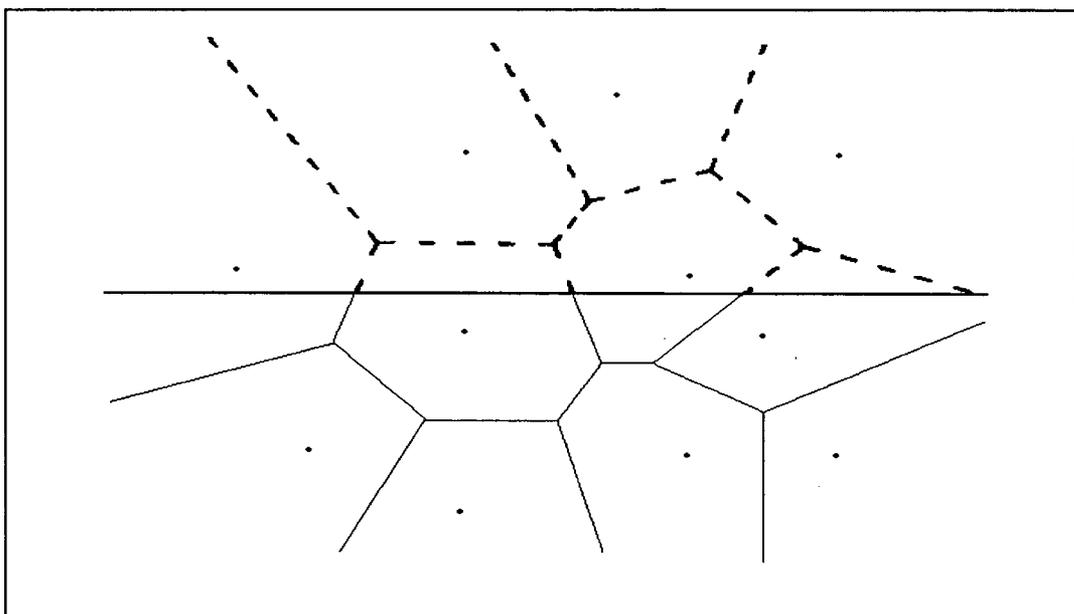


Figura 2.4 - Intersecção do diagrama de Voronoi com a varredura em linha horizontal.

2.3.3 Técnica incremental

Uma das técnicas importantes de projeto de algoritmos, especialmente quando os dados não são todos conhecidos antecipadamente, é a incremental. A técnica incremental calcula a estrutura geométrica pela adição de pontos básicos um por um. A vantagem desta técnica é a simplicidade de implementação, a rapidez e a estabilidade numérica do algoritmo. O primeiro passo é adicionar um ponto p ao DV já existente, assim determina-se o polígono de Voronoi $R(p)$ em que o ponto está, o bissetor $B(p_i, p)$ contribui com uma face de Voronoi para o polígono, como mostrado na figura 2.5.

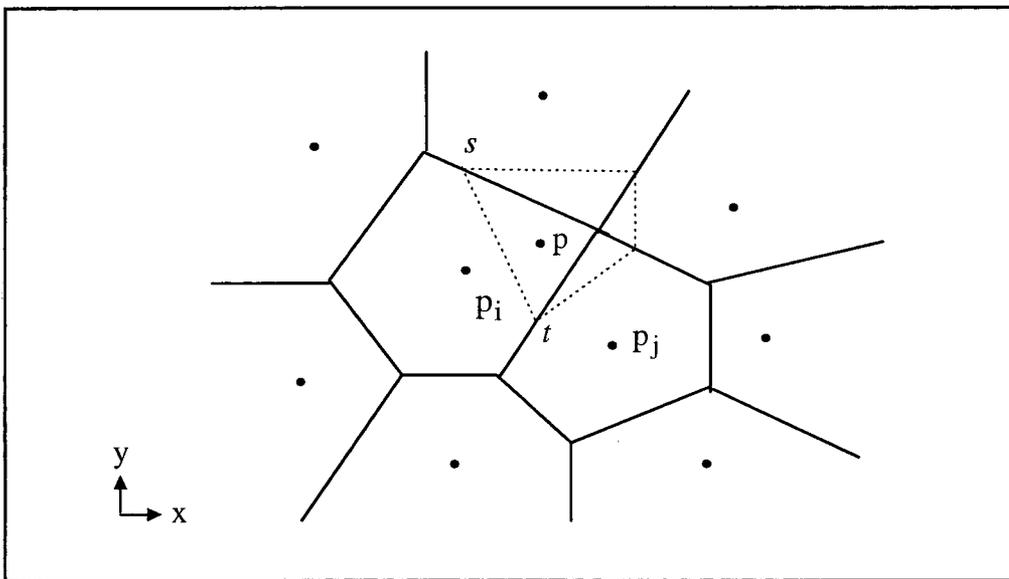
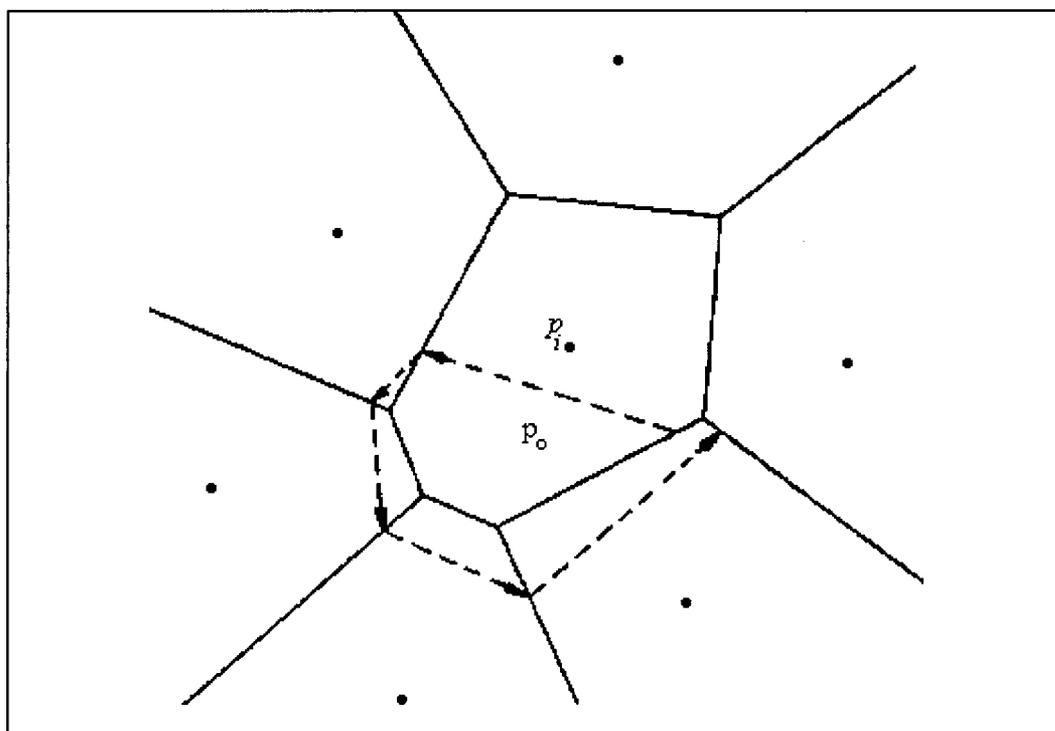


Figura 2.5 - Adicionando um novo ponto p ao diagrama de Voronoi.

A região $R(p)$ do novo gerador p é construída da seguinte forma:

- (i) encontra-se o gerador p_i , cuja região contém p ;
- (ii) traça-se o bissetor perpendicular a p e p_i , linha tracejada na figura 2.5. Este bissetor intercepta as faces da região $R(p_i)$ em dois pontos, s e t , onde os bissetores das próximas regiões devem passar;
- (iii) traça-se novamente o bissetor de p e p_j (iniciando em t), e assim sucessivamente, construindo-se uma seqüência de bissetores entre p e os seus vizinhos, até retornar a face inicial da região $R(p)$;
- (iv) removem-se as faces do interior da região fechada pela seqüência de bissetores, o que forma a região do novo gerador p .

Green e Sibson, em 1977, foram os primeiros na aplicação desta técnica para a construção do DV. O método incremental é simples em princípio, e é dito também robusto frente a erros numéricos, contudo analogamente aos outros, ele é instável quando encontra lugares de degenerescência. Uma situação onde a técnica incremental pode não adequar-se como esperado, ocorre quando o bissetor entre o novo gerador p e seus vizinhos, passa próximo de um desses pontos vizinhos, assim a região do novo gerador pode não formar um ciclo fechado, devido a erros de arredondamento (Sugihara and Iri, 1992). A figura 2.6 mostra um exemplo de inconsistência topológica na construção incremental do polígono $R(p)$ do DV.



fonte: Sugihara e Iri, (1992).

Figura 2.6 - Inconsistência topológica na geração de um polígono de Voronoi.

2.3.3.1 Gerador do diagrama de Voronoi pela técnica incremental

A geração do DV usada no presente trabalho, faz uso da técnica incremental. O gerador de malhas de Voronoi desenvolvido por Maliska Jr. (1993) tem base unicamente na definição geométrica do DV. O programa realiza a discretização do domínio geométrico, fornece a numeração e as conexões de cada ponto gerador com seus vizinhos, definindo a estrutura da matriz do sistema linear resultante do procedimento numérico (Maliska, 1993).

A potencialidade do DV, em função de suas propriedades, é abrangente, porém é importante lembrar que o diagrama surge de um conjunto de pontos geradores definidos pelo usuário, e esses pontos vão determinar a concentração de volumes de controle, a sua distribuição e o formato dos volumes na malha computacional, além da esparsidade da matriz de coeficientes gerada na discretização do domínio.

A característica importante na geração de DV é o aumento do tempo de processamento em função do aumento do tamanho da malha. O algoritmo apresentado neste trabalho consome para n geradores um tempo de processamento da $O(n)$ onde, geralmente, outros métodos gastam um tempo da $O(n \log(n))$. Isto ocorre porque no processo de geração de um volume o algoritmo realiza cálculos apenas com os volumes vizinhos, não sofrendo interferência do número total de volumes (Maliska Jr., 1993).

2.4 Estrutura computacional

A maneira de armazenar os elementos da matriz e dos vetores que formam o sistema linear esparsos encontrado na equação (1.1) faz uso da estrutura de geração do DV apresentada por Maliska Jr., mencionada na seção anterior. Este esquema é bastante eficiente sem dificuldades, para o usuário se familiarizar com ele.

O DV é uma lista de volumes, um vetor, onde cada posição deste vetor é uma estrutura do tipo `tipo_volume`. Cada volume que deseja-se encontrar possui um *número*, que é a sua *posição na lista* e a *estrutura tipo_volume* que possui as seguintes partes (Maliska Jr., 1993):

- (i) `(.cx,.cy)`- coordenadas do ponto gerador do volume de controle discreto i ;
- (ii) `(.x[j],.y[j])`- coordenadas dos vértices do ponto j relativos ao volume de controle discreto i ;
- (iii) `.viz1[j]`- vizinho direto do volume de controle discreto i pela aresta j ;

Esta estrutura define o domínio do espaço dado, como pode ser visto na figura 2.7, que são os volumes de Voronoi para estes geradores.

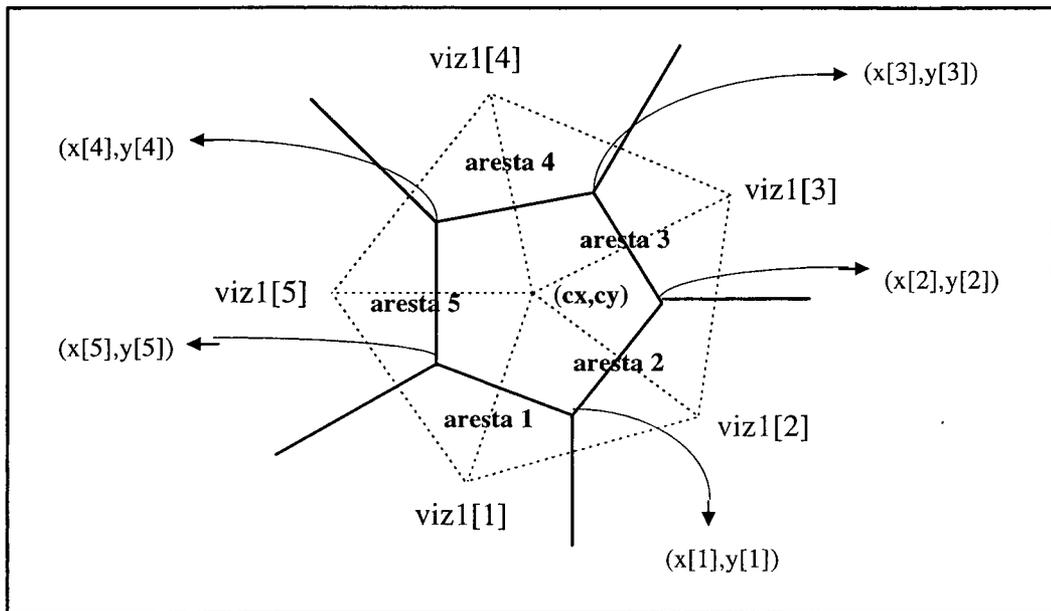


Figura 2.7 - Estrutura computacional do diagrama de Voronoi.

Além dessas definições para os elementos do volume discreto, tem-se as seguintes definições para os elementos da lista de volumes discretos e outros componentes auxiliares para a geração.

- (i) (px, py) - coordenadas do ponto gerador p (ponto que está sendo incrementado);
- (ii) i - volume discreto em que o ponto (px, py) está contido, isto é, são suas coordenadas;
- (iii) $volume[i]$ - volume de controle i do diagrama;
- (iv) $volume[k]$ - volume discreto do ponto nodal k .

Essas definições auxiliarão na montagem da estrutura computacional usada no armazenamento dos elementos da matriz esparsa de coeficientes. O $volume[i]$ do DV ocupará a linha i da matriz de coeficientes e os volumes vizinhos de i , identificados por $viz1[j]$, indicam as posições das colunas da matriz de coeficientes.

Na figura 2.8 mostra-se um DV formado por nove volumes geradores e a tabela 2.1 apresenta as ligações entre cada um destes volumes de acordo com a estrutura computacional usada na geração do DV.

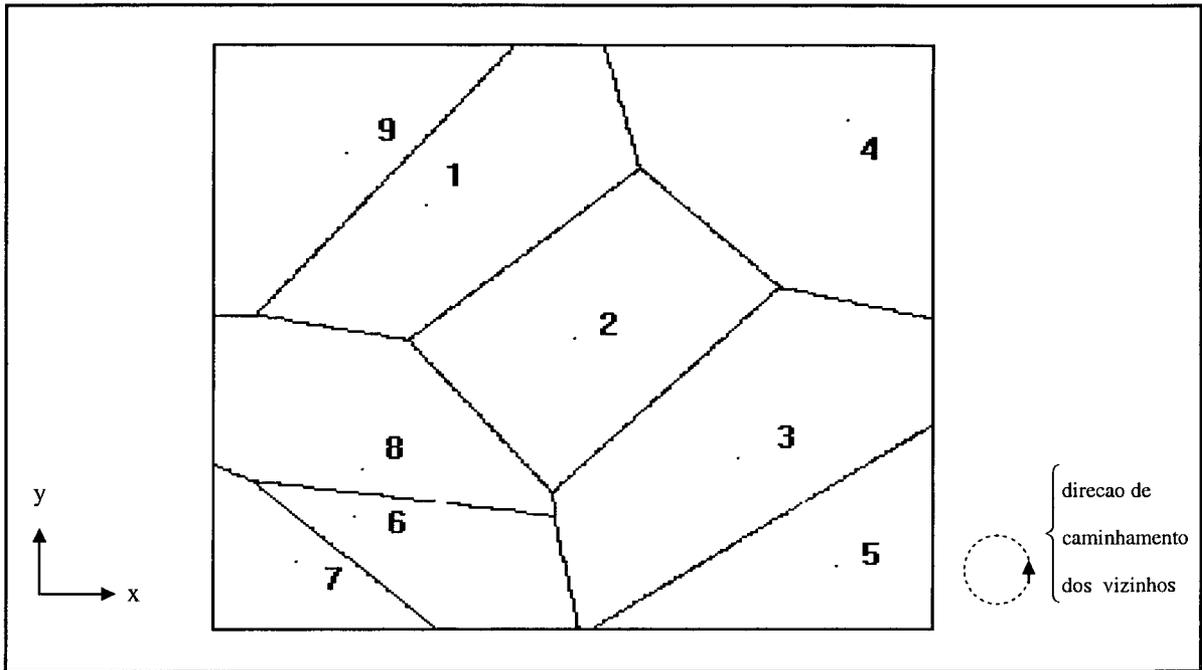


Figura 2.8 - Diagrama de Voronoi para 9 geradores.

Na figura 2.8 a direção de encaminhamento dos vizinhos é a ordem em que os vizinhos são utilizados, no sentido anti-horário, por exemplo.

Tabela 2.1 - Disposição dos vizinhos no diagrama de Voronoi para 9 geradores.

viz1[j]	i=1	i=2	i=3	i=4	i=5	i=6	i=7	i=8	i=9
viz1[1]	2	3	5	1	3	3	6	3	8
viz1[2]	4	4	4	2		8	8	2	1
viz1[3]	9	1	2	3		7		1	
viz1[4]	8	8	8					9	
viz1[5]		6	6					7	
viz1[6]								6	

Na figura 2.9 apresenta-se os coeficientes da matriz, por exemplo na linha 3, considerando assim o volume[i] para i=3, com a seqüência de vizinhos, viz1[j], que é o formato de armazenar os coeficientes utilizado no presente trabalho.

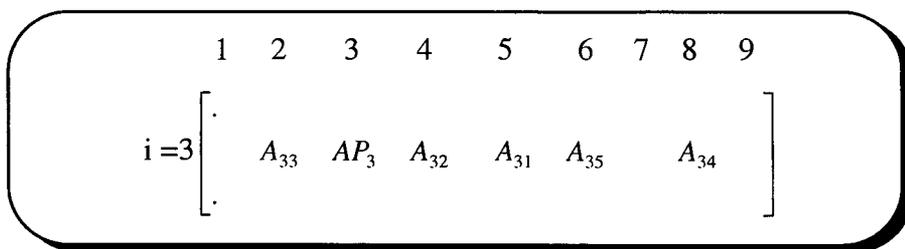


Figura 2.9 - Armazenamento da matriz de coeficientes.

Observa-se na figura 2.9 que a ligação por exemplo, do volume[3] com o volume[2] forma o coeficiente A_{33} e não o coeficiente A_{32} , isto ocorre porque na geração do DV o volume[2] é considerado o terceiro vizinho do volume[3], verifica-se ainda que só os elementos não-nulos são armazenados, usando assim um formato otimizado.

2.5 Conclusão

No presente capítulo apresentou-se o Diagrama de Voronoi, uma construção geométrica que divide o plano de acordo com a regra do *vizinho mais próximo*. Logo, cada ponto está associado com a região do plano mais próxima dele. Os DV têm interessantes propriedades matemáticas e computacionais, as quais foram descritas.

Na geração do DV existem três técnicas principais utilizadas, estas foram mostradas de forma sucinta. Neste trabalho é utilizada a técnica incremental na geração do DV, mostrando-se eficiente e robusta na discretização dos domínios adotados.

O gerador do DV ao final apresenta as conexões entre os volumes vizinhos, deste modo os elementos da matriz de coeficientes são armazenados fazendo uso desta estrutura, exigindo alguns cuidados adicionais por parte do usuário quando for trabalhar, por exemplo, com pré-condicionadores, apresentados no capítulo 5.

Cabe assinalar, antes de encerrar este capítulo, que os DV, tradicionalmente, têm sido empregados em diversas áreas de pesquisas conforme relatado em (Aurenhammer, 1991), além de ser uma vantajosa alternativa para a discretização não-estruturada em problemas de escoamento de fluidos. Porém, esta discretização só recentemente vem sendo difundida na área de CFD com o MVF, principalmente em trabalhos na área de simulação de reservatórios de petróleo (Marcondes, 1996).

Capítulo 3 - Método dos Volumes Finitos

3.1 Introdução

No capítulo 2 foram expostos aspectos gerais do DV juntamente, com uma breve descrição da malha não-estruturada gerada por estes diagramas. Neste capítulo apresenta-se as características básicas do MVF, bem como as vantagens e desvantagens do uso de malhas estruturadas e malhas não-estruturadas geradas pelo DV.

A integração numérica, da equação da condução de calor e das equações de Navier-Stokes, em cada volume de controle, proporciona a obtenção das equações algébricas do sistema linear, que será brevemente descrita no presente capítulo utilizando o MVF. As geometrias com suas respectivas condições de contorno são apresentadas ao final deste capítulo.

3.2 Características

Os DV vem ganhando mais destaque em simulações numéricas de fenômenos físicos, dadas as suas características de construção, permitindo que fluxos sejam facilmente aproximados nas fronteiras dos volumes de controle. O uso do MVF (Patankar, 1980) em malhas estruturadas é tradicional na Mecânica dos Fluidos, enquanto o uso em malhas não-estruturadas é mais recente, reporte-se aos trabalhos de Taniguchi *et al.* (1991), Taniguchi e Kobayashi (1991), Marcondes (1996) entre outros.

Neste trabalho são abordados problemas que envolvem escoamento de fluidos com ou sem transferência de calor. A solução desses problemas envolve o tratamento de um sistema de equações diferenciais parciais (EDPs) não lineares, as equações de Navier-Stokes, cuja solução aproximada está associada a uma discretização do tipo volumes finitos. As características principais deste método, são:

- (i) a discretização das equações diferenciais, formando um sistema de equações algébricas, é efetuada através de um processo de integração aproximada em volumes de controle finitos, convenientemente distribuídos sobre o domínio de solução;
- (ii) as equações aproximadas são obtidas através de balanços de conservação da grandeza física envolvida (massa, quantidade de movimento, entalpia, etc), nos volumes de controle;
- (iii) a avaliação das variáveis dependentes e suas derivadas nas faces dos volumes de controle, são obtidas por funções de interpolação, que podem ser baseadas na física do problema;

(iv) os termos não lineares das equações diferenciais são linearizados nas equações algébricas através de esquemas de separação simples, onde efetua-se o cálculo de coeficientes com campos estimados anteriormente;

(v) os sistemas de equações algébricas, originários da aplicação dos princípios de conservação a cada equação diferencial, são resolvidos simultaneamente ou segregadamente, com posterior acoplamento entre as variáveis dependentes.

3.3 Discretização em malha estruturada e não-estruturada

Para se obter o modelo matemático baseado nas equações diferenciais, que regem problemas conservativos, é necessário realizar um balanço em volume infinitesimal, ou seja, fazer o processo de limites com o volume tendendo a zero.

As equações algébricas no MVF são obtidas promovendo um balanço conservativo em volumes de controle finitos, garantindo a conservação das grandezas físicas em um estágio intermediário, entre o domínio físico global e o infinitesimal, das equações diferenciais. Portanto quanto mais refinadas forem as malhas, mais próximo da solução exata das equações diferenciais se chegará.

Existem dois tipos de discretizações do domínio de cálculo: a *discretização em malha estruturada* e em malha *não-estruturada*. Os volumes de controle da malha estruturada estão vinculados às linhas de um sistema de coordenadas, como mostra a figura 3.1.

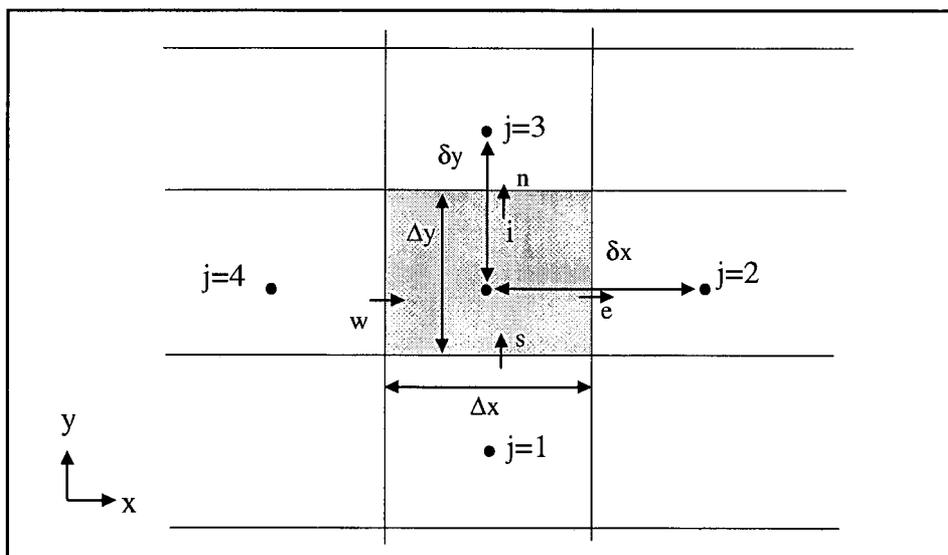


Figura 3.1 - Volume de controle elementar (i,j) para malha estruturada.

Na figura 3.1 os índices (e, w, n, s) representam as direções este, oeste, norte e sul, respectivamente, e são usados para identificar as faces do volume de controle na discretização numérica, na malha cartesiana $S_{ij} = \Delta x$ ou Δy conforme j , e $L_{ij} = \delta x$ ou δy conforme j .

As malhas estruturadas cartesianas tem um número fixo de vizinhos, e permitem que os balanços de massa, quantidade de movimento, energia, entre outros, também sejam estruturados (Peters, 1996). Os sistemas lineares resultantes com o uso destas malhas, apesar de serem esparsos, apresentam lei de formação definida, ou seja, apresentam comprimento de banda definida (Maliska, 1995).

Uma das importantes vantagens na utilização de malha não-estruturada é sua aplicabilidade em domínios geométricos complexos e o seu refinamento em regiões específicas do domínio, para retratar melhor os fluxos locais. Esta constatação deve-se ao fato de que nas malhas não-estruturadas os elementos e pontos podem ser adicionados e/ou retirados localmente de acordo com as características ditadas pela geometria ou o fluxo.

As malhas não-estruturadas, geradas por DV, dão origem a volumes de controle cujas faces são sempre ortogonais às retas que unem o ponto nodal gerador do volume de controle com cada ponto nodal dos volumes vizinhos. Esta característica, por sua vez, simplifica as aproximações numéricas existentes, facilitando a realização de balanços de massa, energia, entre outros (Marcondes *et al.*, 1995).

Todas as malhas localmente ortogonais podem ser classificadas como casos particulares de malhas de Voronoi, tais como: cartesianas, cilíndricas, hexagonais, etc (Weatherill, 1992; Marcondes, 1996). Na figura 3.2 é apresentado um volume de controle da malha não-estruturada gerada por DV.

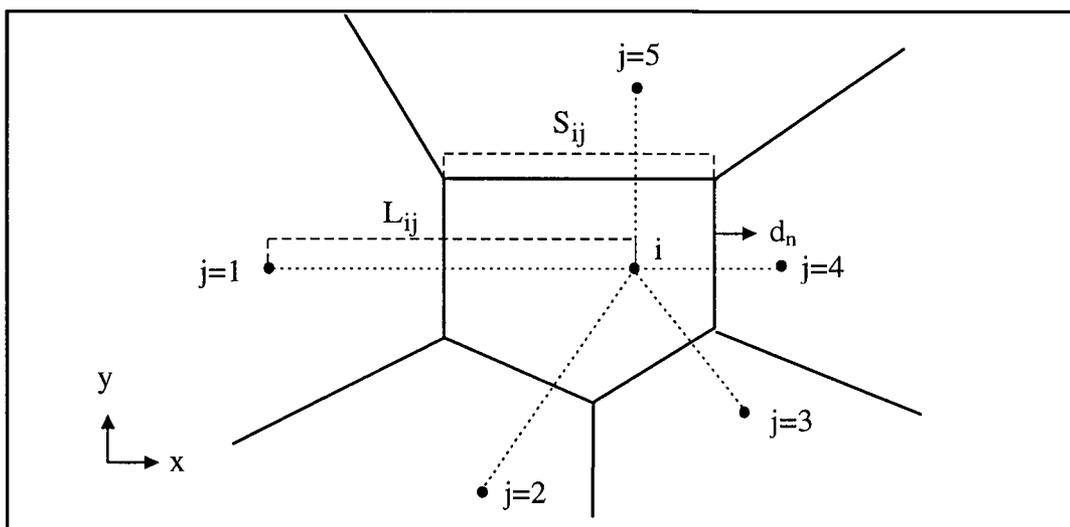


Figura 3.2 - Volume de controle elementar para malha não-estruturada de Voronoi.

Na figura 3.2, i é o índice do ponto p_i gerador do volume de controle, j são os índices dos vizinhos p_j do ponto p_i .

Os sistemas lineares resultantes nestas malhas são esparsos e sem lei de formação, pois o número de vizinhos pode variar em cada volume de controle, e não existe uma lei para a numeração de forma ordenada destes vizinhos, exigindo algoritmos robustos e eficientes para resolvê-los, no sentido de prever as diversas ligações possíveis. Contudo as malhas não-estruturadas geradas pelos DV aliam a versatilidade das malhas não-estruturadas com a simplicidade das malhas ortogonais.

3.4 Disposição das variáveis dependentes na malha computacional

Para a discretização das equações governantes define-se a priori a localização das variáveis dependentes na malha, aspecto importante para obter-se economia de memória e tempo de processamento. Diferentes estratégias de arranjo das variáveis na malha podem ser adotadas, mas, para sistemas coordenados ortogonais, dois deles são empregados: arranjo co-localizado e arranjo desencontrado (Maliska, 1995). As vantagens e desvantagens das diversas formas, para a localização das variáveis na malha estruturada, são encontradas em vários trabalhos entre eles pode-se citar os trabalhos de Silva (1991) e Maliska (1995).

O arranjo aqui adotado é caracterizado pela localização de todas as variáveis no centro do volume de controle, sobre o ponto nodal, e é denominado arranjo co-localizado (Silva, 1991; Peters, 1994). O arranjo co-localizado é a escolha natural, pela simplicidade no controle dos índices das variáveis na implementação computacional e por possuir um único volume de controle para realizar a integração de todas as equações.

A figura 3.3 ilustra a situação de discretização com o arranjo co-localizado. Este arranjo sempre foi aplicado à problemas de escoamentos com fluidos compressíveis, sem relatos na literatura de dificuldades associadas ao seu uso, exceto em problemas transientes (Peters, 1994).

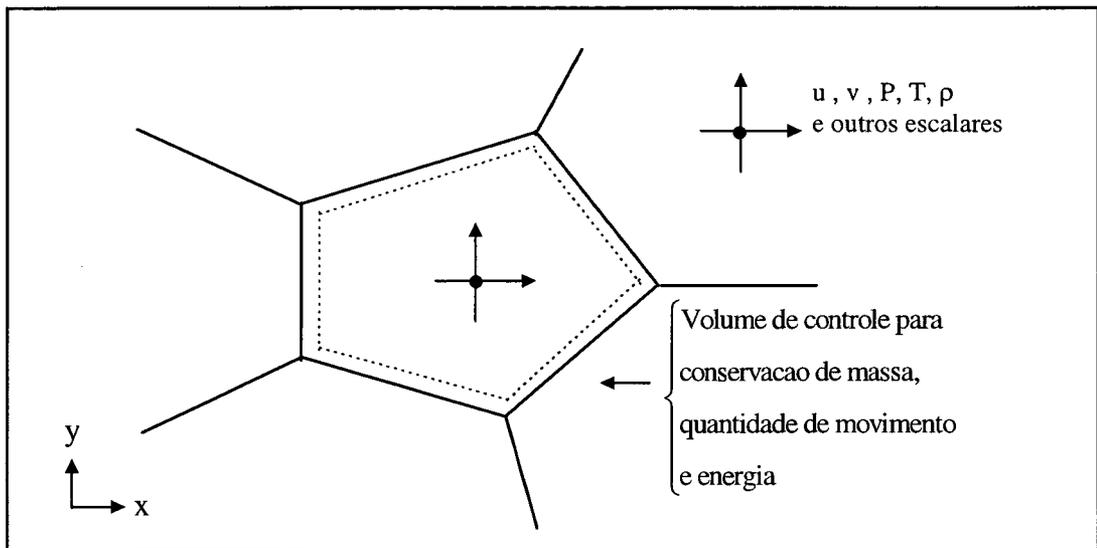


Figura 3.3 - Volume de controle para u, v, T, P e ρ no arranjo co-localizado de variáveis.

3.5 Equações governantes

As equações diferenciais parciais, que envolvem fluxos convectivos e difusivos, podem ser representadas genericamente para uma variável ϕ da seguinte forma vetorial,

$$\frac{\partial}{\partial t}(\rho \phi) + \vec{\nabla} \cdot (\vec{J}) = S^\phi \quad (3.1)$$

onde ϕ é uma variável dependente genérica, ρ é a massa específica do fluido, S indica um termo fonte por unidade de volume e \vec{J} é o fluxo total de ϕ (convectivo e difusivo) definido na equação (3.1.a) e \vec{V} é o vetor velocidade.

$$\vec{J} = \rho \vec{V} \phi - \Gamma^\phi \vec{\nabla} \phi \quad (3.1.a)$$

Na tabela 3.1 são apresentadas as equações de conservação, onde ϕ (na equação (3.1)) assume um valor específico em cada caso.

Tabela 3.1 - Representação de ϕ , S^ϕ e Γ^ϕ .

Tipo de equações	ϕ	S^ϕ	Γ^ϕ
Conservação da massa	1	0	0
Conservação da quantidade de movimento na direção x	u	$-\frac{\partial P}{\partial x}$	μ
Conservação da quantidade de movimento na direção y	v	$-\frac{\partial P}{\partial y}$	μ
Conservação da energia	T	$\frac{q'''}{c_p}$	$\frac{K}{c_p}$

Com os valores da tabela 3.1 substituídos na equação (3.1), obtém-se as equações de Navier-Stokes para fluidos Newtonianos, considerando as propriedades constantes em geometria bidimensional. As equações de Navier-Stokes correspondem apenas as equações de conservação da quantidade de movimento, no entanto, é comum usar a equação de conservação da massa e da energia, neste conjunto (De Bortoli, 1996).

(i) *Conservação da massa*

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \quad (3.2)$$

(ii) *Conservação da quantidade de movimento na direção x*

$$\frac{\partial}{\partial t}(\rho u) + \frac{\partial}{\partial x}(\rho u u) + \frac{\partial}{\partial y}(\rho u v) = -\frac{\partial P}{\partial x} + \frac{\partial}{\partial x}\left(\mu \frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(\mu \frac{\partial u}{\partial y}\right) \quad (3.3)$$

(iii) *Conservação da quantidade de movimento na direção y*

$$\frac{\partial}{\partial t}(\rho v) + \frac{\partial}{\partial x}(\rho u v) + \frac{\partial}{\partial y}(\rho v v) = -\frac{\partial P}{\partial y} + \frac{\partial}{\partial x}\left(\mu \frac{\partial v}{\partial x}\right) + \frac{\partial}{\partial y}\left(\mu \frac{\partial v}{\partial y}\right) \quad (3.4)$$

(iv) *Conservação da energia*

$$\frac{\partial}{\partial t}(\rho T) + \frac{\partial}{\partial x}(\rho u T) + \frac{\partial}{\partial y}(\rho v T) = \frac{\partial}{\partial x}\left(\frac{K}{c_p} \frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(\frac{K}{c_p} \frac{\partial T}{\partial y}\right) + \frac{q'''}{c_p} \quad (3.5)$$

3.5.1 Discretização das equações governantes

No presente trabalho a equação vetorial de ϕ será discretizada, onde ϕ assume as variáveis T e u (ou v) dentro do contexto do MVF, integrando as equações em todos os volumes de controle do domínio discretizado.

O sistema de equações lineares gerado a partir da equação (3.5), será resolvido adotando-se $\vec{V} = 0$, apenas difusão de calor em fluido estático ou sólido, sendo que as equações de movimento na direção x e y (Navier-Stokes) não têm solução analítica, exceto em casos especiais de situações unidimensionais (Peters, 1996) e serão discretizadas num segundo instante.

No processo de integração numérica sobre o volume de controle definido nos DV é mais simples trabalhar com as equações na forma vetorial. Integrando a equação (3.1), no volume de controle da figura 3.2, e sendo V o volume de integração para cada volume de controle, tem-se:

$$\int_v \frac{\partial(\rho\phi)}{\partial t} dV + \int_v (\vec{V} \cdot \vec{J}) dV = \int_v S^\phi dV \quad (3.6)$$

(a) (b) (c)

3.5.1.1 Discretização da equação da condução de calor

Aproximando o primeiro termo da equação (3.6) para a variável $\phi = T$ (equação da condução de calor), fazendo uso da diferença ascendente de primeira ordem no tempo tem-se:

$$\int_v \frac{\partial}{\partial t} (\rho T) dV = \frac{T_i - T_i^0}{\Delta t} \Delta V \quad (3.7)$$

onde T_i é a temperatura do ponto nodal central no instante de tempo atual.

No segundo termo, pelo *teorema da divergência*, pode-se reduzir uma integração volumétrica de um divergente para uma integração de superfície no contorno do volume de controle em questão, conforme representado na equação (3.8),

$$\int_v \vec{V} \cdot \left(\frac{K}{c_p} \vec{\nabla} T \right) dV = \int_s \frac{K}{c_p} \vec{\nabla} T \cdot d\vec{n} \quad (3.8)$$

onde $d\vec{n}$ é a superfície de integração normal de cada face do volume de controle. Conforme equação (3.8), as integrações de superfície, sobre cada face do volume de controle, quando discretizadas geram a expressão,

$$\int_s \frac{K}{c_p} \vec{\nabla} T \cdot d\vec{n} = \sum_{j=1}^{NV} \frac{K_{ij}}{c_p} \left(\frac{T_j - T_i}{L_{ij}} \right) S_{ij} \quad (3.9)$$

onde S_{ij} é a área normal em cada face do volume de controle i , ou seja, é a distância entre dois vértices consecutivos nas faces dos volumes de controle, L_{ij} é a distância entre o ponto nodal central do volume de controle i com cada vizinho j , ambas observadas na figura 3.2, NV é o número de vizinhos do ponto nodal i , T_j é a temperatura de cada vizinho j de i . Reorganizando a equação (3.6) tem-se,

$$AP_i T_i - \sum_{j=1}^{NV} (A_{ij} T_j) = b_i \quad (3.10)$$

sendo,

$$A_{ij} = \frac{k_{ij} S_{ij}}{c_p L_{ij}} \quad (3.11)$$

$$AP_i = \sum_{j=1}^{NV} A_{ij} + AP_i^0 \quad (3.12)$$

$$b_i = AP_i^0 T_i^0 + \frac{q'''}{c_p} \Delta V \quad (3.13)$$

$$AP_i^0 = \frac{\Delta V}{\Delta t}, \text{ com } \Delta t \neq 0. \quad (3.14)$$

onde AP_i é o coeficiente do ponto nodal central e A_{ij} são os coeficientes dos vizinhos de p_i e ΔV é o volume do volume de controle elementar.

3.5.1.2 Discretização das equações de Navier-Stokes

O procedimento de integração numérica da equação (3.6) para u (ou v) é encontrado no trabalho de Cardoso (Cardoso, 1997), e é realizado inicialmente pela integração do termo (a) no volume de controle do DV,

$$\int_v \frac{\partial(\rho\phi)}{\partial t} dV = \frac{\rho_i \phi_i - \rho_i \phi_i^0}{\Delta t} \Delta V_i \quad (3.15)$$

com uma aproximação de primeira ordem no tempo, pode-se organizar a equação (3.15) como,

$$\int_v \frac{\partial(\rho\phi)}{\partial t} dV = (\phi_i - \phi_i^0) A_i^0 \quad (3.16)$$

onde $A_i^0 = \frac{\rho_i \Delta V_i}{\Delta t}$

Nos demais termos é necessário estabelecer qual o instante de tempo que avalia-se as variáveis ϕ , normalmente é adotada a *formulação implícita*, onde todas as variáveis são estimadas no instante de tempo atual ($t + \Delta t$) que é denotado por 1, que por simplicidade são suprimidos, já que aparecem em grande número.

No segundo termo da equação (3.6) tem-se que avaliar o fluxo de ϕ nas faces dos volumes de controle, J_{ij} e para tal é necessário um esquema de interpolação.

$$\int_V (\bar{\nabla} J) dV = \int_{S_n} \bar{J} d\bar{S}_n = \sum_{j=1}^{NV} J_{ij} S_{ij} \quad (3.17)$$

O fluxo J líquido que passa na face ij é representado por $J_{ij} S_{ij}$. Considerando w como a velocidade normal a face ij e n a direção normal a cada face ij e usando a equação (3.1.a) tem-se,

$$\sum_{j=1}^{NV} J_{ij} S_{ij} = \sum_{j=1}^{NV} \left[\rho w \phi - \Gamma \frac{\partial \phi}{\partial n} \right]_{ij} S_{ij} \quad (3.18)$$

onde,

$$\rho_{ij} = \frac{\rho_i + \rho_j}{2} \quad \Gamma_{ij} = \frac{2\Gamma_i \Gamma_j}{\Gamma_i + \Gamma_j}$$

$w_{ij} = \frac{w_i + w_j}{2}$, normalmente opta-se por uma média simples onde as velocidades normais w_i e w_j são calculadas por projeção de u_i e v_i na direção normal \bar{n}_{ij}

$$\left(\frac{\partial \phi}{\partial n} \right)_{ij} = \frac{\phi_j - \phi_i}{L_{ij}}, \text{ componente do gradiente de } \phi \text{ normal a face } ij.$$

É necessária uma avaliação para ϕ_{ij} , cujo resultado exato seria a própria solução do sistema de equações. Porém isto não faz sentido, já que o objetivo da interpolação é justamente a resolução das equações diferenciais. Das propostas que surgiram, a mais consistente fisicamente é a proposta por Patankar (1980), que trata a resolução da equação de transporte unidimensional. Devido ao custo computacional para a avaliação desta proposta, usa-se uma simplificação mais conhecida, conforme equação (3.19),

$$J = - \left(\frac{\Gamma_{ij}}{L_{ij}} \right) F(Pe) (\phi_j - \phi_i) + \rho_{ij} w_{ij} \phi^{UP} \quad (3.19)$$

onde

$$\phi^{UP} = \begin{cases} \phi_i \Rightarrow w_{ij} > 0 \\ \phi_j \Rightarrow w_{ij} < 0 \end{cases}$$

$F(Pe)$ assume valores diferentes, conforme o esquema escolhido. Neste trabalho adota-se o esquema da Lei da Potência (Patankar, 1980),

$$F(Pe) = \max[0, (1 - 0,1|Pe|)^5] \quad (3.20)$$

Voltando na equação (3.17) tem-se,

$$\int_{S_n} \bar{J} dS_n = \sum_{j=1}^{NV} J_{ij} S_{ij} = \sum (-D_{ij} \cdot F(Pe)(\phi_j - \phi_i) + \max[0, +F_{ij}] \phi_i - \max[0, -F_{ij}] \phi_j) \quad (3.21)$$

onde

$F_{ij} = (\rho w S)_{ij}$ é a vazão mássica

$D_{ij} = \left(\frac{\Gamma S}{L} \right)_{ij}$ é o coeficiente de difusão

$Pe = \frac{F_{ij}}{D_{ij}}$ é o número de Peclet do volume de controle i em relação a face j .

O terceiro termo da equação (3.6) é o termo fonte das equações u e v , e possui avaliações de gradientes de pressão, que podem ser obtidas de formas diferentes. O vetor gradiente de pressão ($\vec{\nabla} p_i$) deve ser avaliado sobre cada ponto nodal central do domínio, através de suas componentes ∇p_x e ∇p_y ,

$$\vec{\nabla} p_i = \nabla p_x \hat{i} + \nabla p_y \hat{j} \quad (3.22)$$

O que tem-se disponível são apenas as componentes do gradiente nas direções normais às faces, avaliadas sobre elas,

$$\nabla p_{ij} = \frac{p_j - p_i}{L_{ij}} \quad (3.23)$$

No presente trabalho optou-se por usar três formas diferenciadas para o cálculo dos gradientes de pressão:

(i) Média ponderada entre os gradientes normais projetados

Uma das idéias mais simples para avaliação do gradiente de pressão é considerar as contribuições de cada componente normal nas faces, deste modo, o gradiente de pressão pode ser obtido por médias ponderadas de seus componentes sobre cada face.

$$\nabla p_x = \frac{\sum_{j=1}^{NV} \left[\left(\bar{\nabla} p_{ij} \cdot \hat{i} \right) L_{ij} \right]}{\sum_{j=1}^{NV} L_{ij}} \quad (3.24)$$

$$\nabla p_y = \frac{\sum_{j=1}^{NV} \left[\left(\bar{\nabla} p_{ij} \cdot \hat{j} \right) L_{ij} \right]}{\sum_{j=1}^{NV} L_{ij}} \quad (3.25)$$

onde $\bar{\nabla} p_{ij}$ é o vetor gradiente de pressão normal à interface ij , calculado por:

$$\bar{\nabla} p_{ij} = \left(\frac{p_j - p_i}{L_{ij}} \right) \cdot \hat{e}_{ij} = \left(\frac{p_j - p_i}{L_{ij}} \right) \cdot \left(e_{x_{ij}} \hat{i} + e_{y_{ij}} \hat{j} \right) \quad (3.26)$$

A equação (3.25) é uma proposta da primeira versão do livro de Maliska (Maliska, 1995), que ainda não tinha sido testada a exaustão; posteriormente Maliska verificou que esta proposta não era genérica e sugeriu uma forma de avaliação mais eficiente. No entanto, esta expressão permitiu que Cardoso fizesse uma reflexão sobre as mesmas, e considerasse apenas as influências das distâncias L_{ij} projetadas nas direções de cálculo das componentes do gradiente de pressão (Cardoso, 1997). Verifica-se que as projeções de L_{ij} em y nada acrescentam ao cálculo do gradiente de pressão na direção x , vide (Cardoso, 1997). Propôs-se, então, que para o cálculo do gradiente de pressão na direção x fossem consideradas apenas as distâncias L_{ij} projetadas na direção x . Analogamente propôs-se o cálculo do gradiente de pressão em y , conforme segue:

$$\nabla p_x = \frac{\sum_{j=1}^{NV} \left[\left(\bar{\nabla} p_{ij} \cdot \hat{i} \right) L_{ij} \right]}{\sum_{j=1}^{NV} (L_{ij} |e_{x_{ij}}|)} \quad (3.27)$$

$$\nabla p_y = \frac{\sum_{j=1}^{NV} \left[\left(\vec{\nabla} p_{ij} \cdot \hat{j} \right) L_{ij} \right]}{\sum_{j=1}^{NV} (L_{ij} |e_{y_{ij}}|)} \quad (3.28)$$

Os denominadores das equações. (3.27) e (3.28) consideram apenas as distâncias que contribuem com o gradiente projetado. Portanto, as projeções do gradiente em x, por exemplo, consideram apenas as distâncias L_{ij} projetadas em x. Este cálculo do gradiente de pressão é denominado média ponderada entre os gradientes normais projetados (MPGNP).

Tratamento especial foi dispensado aos pontos nodais ligados às fronteiras do domínio geométrico, cujo campo de pressão do contorno não é conhecido. Nestes pontos considerou-se apenas as contribuições do campo de pressão interno.

(ii) Mínimo resíduo quadrático

Existe uma diferença $(\delta p)_{ij}$ entre a avaliação sobre cada face e a projeção do gradiente na direção normal \hat{n}_{ij} , avaliado sobre o ponto nodal central. Esta diferença δ_{ij} é dada por:

$$(\delta p)_{ij} = (\nabla p)_{ij} - (\vec{\nabla} p_i) \cdot \hat{n}_{ij} \quad (3.29)$$

Conforme Taniguchi *et al.*(1991) e Taniguchi e Kobayashi (1991), propõe-se uma função global ψ_i de minimização das diferenças δp_{ij} , entre os vizinhos j, conforme equação (3.30),

$$\psi_i = \sum_{j=1}^{NV} (g_{ij} \delta p_{ij} \delta p_{ij}) \quad (3.30)$$

Substituindo a equação (3.29) e (3.26) na equação (3.30), obtém-se

$$\Psi_i = \sum_{j=1}^{NV} \left\{ g_{ij} \left[(\nabla p)_{ij} - \left(\nabla p_x \hat{i} + \nabla p_y \hat{j} \right) \cdot \hat{n}_{ij} \right]^2 \right\} \quad (3.31)$$

$$\Psi_i = \sum_{j=1}^{NV} \left\{ g_{ij} \left[(\nabla p)_{ij} - \left(\nabla p_x e_{x_{ij}} + \nabla p_y e_{y_{ij}} \right) \right]^2 \right\} \quad (3.32)$$

onde g_{ij} é um fator peso do gradiente normal de cada face. Neste caso, adotou-se g_{ij} como sendo proporcional ao ângulo de visão de cada face ij em relação ao ponto central i (Taniguchi *et al.*,

1991) e (Taniguchi and Kobayashi, 1991). Este valor é equivalente a razão entre a área de passagem S_{ij} e a distância L_{ij} . Portanto, g_{ij} é dado por

$$g_{ij} = \frac{S_{ij}}{L_{ij}} \quad (3.33)$$

Fazendo-se uma minimização de ψ_i em função de cada componente do vetor gradiente de pressão, ∇p_x e ∇p_y , obtém-se duas equações para as duas componentes incógnitas, quais sejam

$$\frac{\partial \Psi_i}{\partial (\nabla p_x)} = \sum_{j=1}^{NV} \left\{ 2g_{ij} \left[(\nabla p)_{ij} - \left((\nabla p)_x e_{x_{ij}} + (\nabla p)_y e_{y_{ij}} \right) \right] e_{x_{ij}} \right\} = 0 \quad (3.34)$$

$$\frac{\partial \Psi_i}{\partial (\nabla p_y)} = \sum_{j=1}^{NV} \left\{ 2g_{ij} \left[(\nabla p)_{ij} - \left((\nabla p)_x e_{x_{ij}} + (\nabla p)_y e_{y_{ij}} \right) \right] e_{y_{ij}} \right\} = 0 \quad (3.35)$$

A partir das equações (3.34) e (3.35) pode ser obtido um sistema de equações lineares para o cálculo do ∇p_x e ∇p_y , fazendo uso do mínimo resíduo quadrático (MRQ) da seguinte forma:

$$\begin{cases} \left[\sum_{j=1}^{NV} g_{ij} (e_{x_{ij}})^2 \right] \nabla p_x + \left[\sum_{j=1}^{NV} g_{ij} (e_{y_{ij}} e_{x_{ij}}) \right] \nabla p_y = \sum_{j=1}^{NV} \left[g_{ij} \bar{\nabla} p_{ij} e_{x_{ij}} \right] \\ \left[\sum_{j=1}^{NV} g_{ij} (e_{x_{ij}} e_{y_{ij}}) \right] \nabla p_x + \left[\sum_{j=1}^{NV} g_{ij} (e_{y_{ij}})^2 \right] \nabla p_y = \sum_{j=1}^{NV} \left[g_{ij} \bar{\nabla} p_{ij} e_{y_{ij}} \right] \end{cases} \quad (3.36)$$

Resolvendo-se o sistema dado pela equação (3.36), tem-se

$$\nabla p_x = \frac{\left[\sum_{j=1}^{NV} \left[g_{ij} \bar{\nabla} p_{ij} e_{x_{ij}} \right] \right] \left[\sum_{j=1}^{NV} g_{ij} (e_{y_{ij}})^2 \right] - \left[\sum_{j=1}^{NV} g_{ij} (e_{y_{ij}} e_{x_{ij}}) \right] \left[\sum_{j=1}^{NV} \left[g_{ij} \bar{\nabla} p_{ij} e_{y_{ij}} \right] \right]}{\left[\sum_{j=1}^{NV} g_{ij} (e_{x_{ij}})^2 \right] \left[\sum_{j=1}^{NV} g_{ij} (e_{y_{ij}})^2 \right] - \left[\sum_{j=1}^{NV} g_{ij} (e_{x_{ij}} e_{y_{ij}}) \right]^2} \quad (3.37)$$

e

$$\nabla p_y = \frac{\left[\sum_{j=1}^{NV} \left[g_{ij} \bar{\nabla} p_{ij} e_{y_{ij}} \right] \right] \left[\sum_{j=1}^{NV} g_{ij} (e_{x_{ij}})^2 \right] - \left[\sum_{j=1}^{NV} g_{ij} (e_{x_{ij}} e_{y_{ij}}) \right] \left[\sum_{j=1}^{NV} \left[g_{ij} \bar{\nabla} p_{ij} e_{x_{ij}} \right] \right]}{\left[\sum_{j=1}^{NV} g_{ij} (e_{x_{ij}})^2 \right] \left[\sum_{j=1}^{NV} g_{ij} (e_{y_{ij}})^2 \right] - \left[\sum_{j=1}^{NV} g_{ij} (e_{x_{ij}} e_{y_{ij}}) \right]^2} \quad (3.38)$$

Na avaliação foram consideradas apenas as contribuições dos gradientes internos ao domínio geométrico, sem considerar o campo de pressão sobre o contorno geométrico.

(iii) Média ponderada modificada entre os gradientes normais projetados

Neste trabalho propõe-se uma modificação no gradiente de pressão apresentado por Cardoso (1997) este novo gradiente denominado de média ponderada modificada entre os gradientes normais projetados (MPMGNP). Este gradiente utiliza o fator peso apresentado na equação (3.33), como fator de ponderação da influência de cada gradiente normal $\vec{\nabla}p_{ij}$, conforme equações (3.39) e (3.40) a seguir,

$$\nabla p_x = \frac{\sum_{j=1}^{NV} \left[\left(\vec{\nabla}p_{ij} \cdot \hat{i} \right) g_{ij} \right]}{\sum_{j=1}^{NV} (g_{ij} |e_{x_{ij}}|)} \quad (3.39)$$

e analogamente para y

$$\nabla p_y = \frac{\sum_{j=1}^{NV} \left[\left(\vec{\nabla}p_{ij} \cdot \hat{j} \right) g_{ij} \right]}{\sum_{j=1}^{NV} (g_{ij} |e_{y_{ij}}|)} \quad (3.40)$$

Obtidas numericamente as duas componentes do gradiente de pressão, segundo algum dos gradientes apresentados, pode-se avaliar as componentes de velocidade u e v a partir de um campo inicial u^* , v^* e p^* . Agrupando-se os termos e arrumando as variáveis, tem-se,

$$(\phi_i - \phi_i^0) A_i^0 + \sum_{j=1}^{NV} \left[-D_{ij} \cdot F(Pe) (\phi_j - \phi_i) + \max[0, +F_{ij}] \phi_i - \max[0, -F_{ij}] \phi_j \right] = S^\phi \Delta V_i \quad (3.41)$$

Somando a equação discretizada da continuidade (para escoamento incompressível, $\rho = \text{cte}$) multiplicada por ϕ_i à equação acima, e alocando os termos em ϕ_i no lado direito tem-se,

$$\left\{ A_i^0 + \sum_{j=1}^{NV} \left[\max[0, -F_{ij}] + D_{ij} F(Pe) \right] \right\} \phi_i - \sum_{j=1}^{NV} \left\{ \left[\max[0, -F_{ij}] + D_{ij} F(Pe) \right] \phi_j \right\} = S^\phi \Delta V_i + \phi_i^0 A_i^0 \quad (3.42)$$

Logo,

$$AP_i \phi_i - \sum_{j=1}^{NV} (A_{ij} \phi_j) = b_i^\phi \quad (3.43)$$

onde,

$$A_{ij} = [\max[0, -F_{ij}] + D_{ij} \cdot F(Pe)] \quad (3.44)$$

$$AP_i = \sum_{j=1}^{NV} A_{ij} + A_i^0 \quad (3.45)$$

$$b_i^\phi = A_i^0 \phi_i^0 + S^\phi \Delta \mathcal{V}_i \quad (3.46)$$

Para $\phi = u$ ou v , encontra-se um termo fonte auxiliar de gradiente de pressão.

$$AP_i u_i = \sum_{j=1}^{NV} (A_{ij} u_j) + b_i^u - \Delta \mathcal{V}_i (\nabla p)_x$$

$$AP_i v_i = \sum_{j=1}^{NV} (A_{ij} v_j) + b_i^v - \Delta \mathcal{V}_i (\nabla p)_y \quad (3.47)$$

Verifica-se que existe apenas uma equação evolutiva para u e outra para v , ou para um ϕ que seja regido por uma equação de transporte do tipo convectiva e difusiva. A pressão aparece como uma variável das equações gerais, sem uma equação evolutiva para esta variável.

Uma maneira computacionalmente *barata*, de resolver este problema, é usar a equação da conservação da massa como equação para avaliação da pressão, adotando um acoplamento entre pressão e velocidade, resolvendo segregadamente as variáveis envolvidas: u , v e p .

O método aqui adotado é o SIMPLEC, que é o SIMPLE (“Semi-Implicit Method for Pressure Linked Equations”) consistente, modificado para malhas co-localizadas. A idéia principal do método é encontrar um campo de pressão que satisfaça a equação da conservação da massa (Van Doormaal and Raithby, 1984). O método SIMPLEC difere do SIMPLE, apenas nas equações de correção das velocidades. Seguindo as etapas encontradas em Cardoso (1997) obtém-se um campo de pressão, conforme equação (3.48) que segue,

$$AP_i p_i = \sum_{j=1}^{NV} A_{ij} p_j - \sum_{j=1}^{NV} \rho_{ij} w_{ij}^* S_{ij} \quad (3.48)$$

onde

$$A_{ij} = \frac{\rho_{ij} S_{ij} d_{ij}^w}{L_{ij}} \quad (3.49)$$

$$AP_i = \sum_{j=1}^{NV} A_{ij} \quad (3.50)$$

$$p = p^* + p' \quad (3.51)$$

Obtidos os valores de p' , através da resolução do sistema linear dado pela equação (3.48), pode-se corrigir p^* , através da equação (3.51) e obter o campo correto de pressão p . Então, toma-se p correto como sendo o p^* do próximo nível iterativo. Maiores detalhes sobre a discretização numérica podem ser encontrados em Patankar (1980), Maliska (1995), Peters (1996) e Cardoso (1997). Assim, os sistemas resultantes da discretização das equações de Navier-Stokes, para as velocidades u , v e para a correção da pressão, são apresentados nas equações (3.52), (3.53) e (3.54), respectivamente,

$$AP_i u_i = \sum_{j=1}^{NV} (A_{ij} u_j) + b_i^u - \Delta \mathcal{O}_i (\nabla p)_x \quad (3.52)$$

$$AP_i v_i = \sum_{j=1}^{NV} (A_{ij} v_j) + b_i^v - \Delta \mathcal{O}_i (\nabla p)_y \quad (3.53)$$

$$AP_i p'_i = \sum_{j=1}^{NV} A_{ij} p'_j - \sum_{j=1}^{NV} \rho_{ij} w_{ij}^* S_{ij} \quad (3.54)$$

Na figura (3.4) é apresentado um fluxograma definindo os passos a serem seguidos na implementação do algoritmo para resolução das equações de Navier-Stokes, algumas das equações, descritas nas equações anteriores estão presentes neste fluxograma.

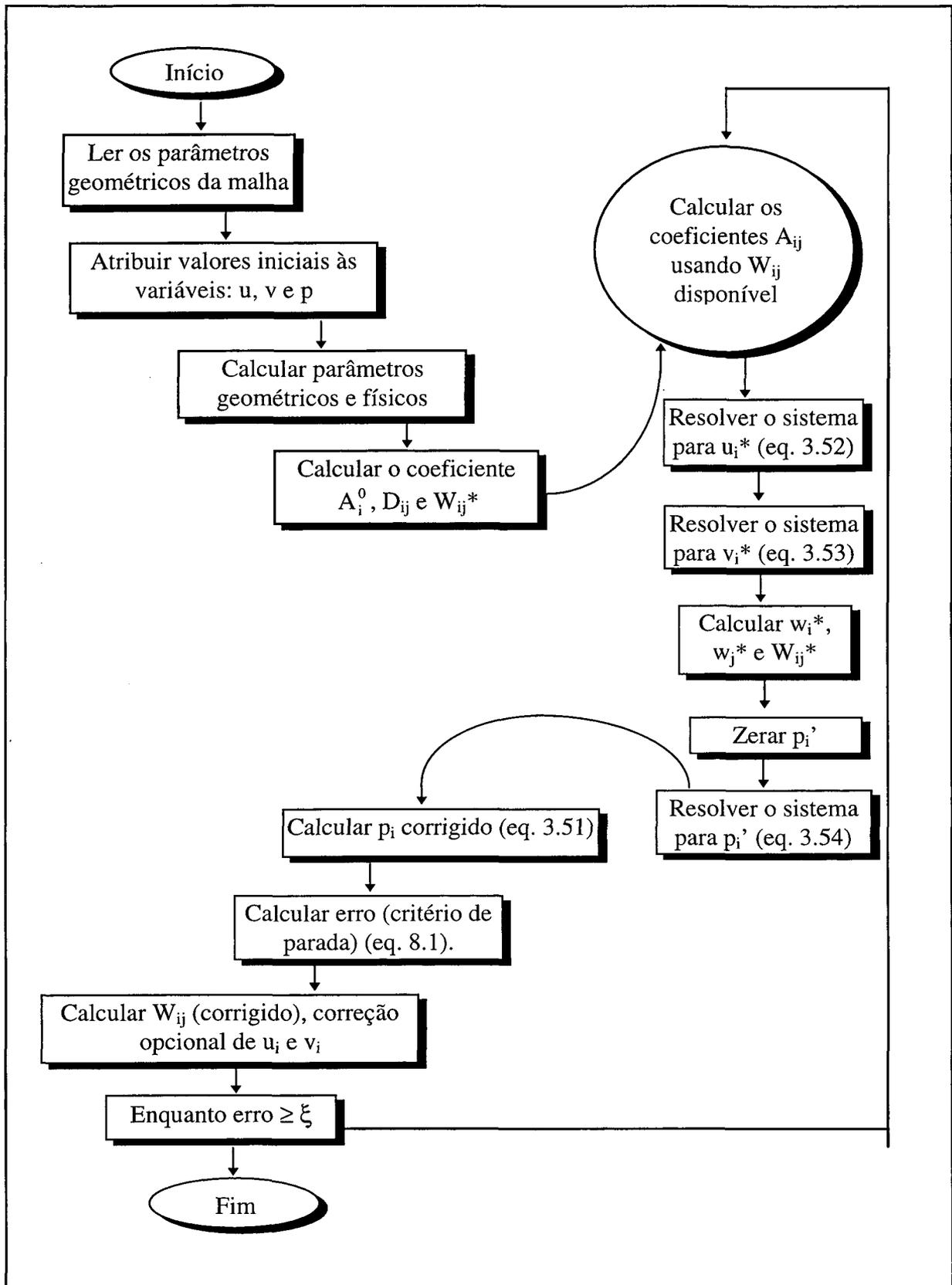


Figura 3.4 - Fluxograma para resolução das equações de Navier-Stokes.

3.6 Condições de contorno e apresentação das geometrias

Os domínios computacionais bidimensionais de solução escolhidos são apresentados na figura 3.5, e correspondem a uma cavidade quadrada, um sólido em forma de L e um sólido em forma triangular, cujas dimensões são definidas em u.c. (unidades de comprimento).

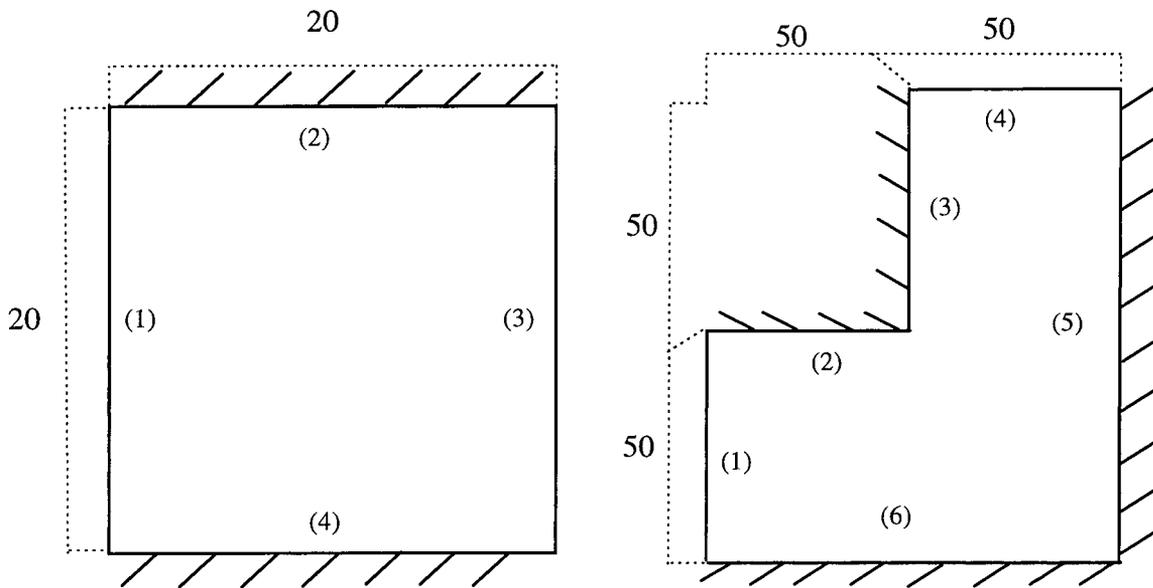


Figura 3.5.a - Domínio geométrico em formato quadrado e L.

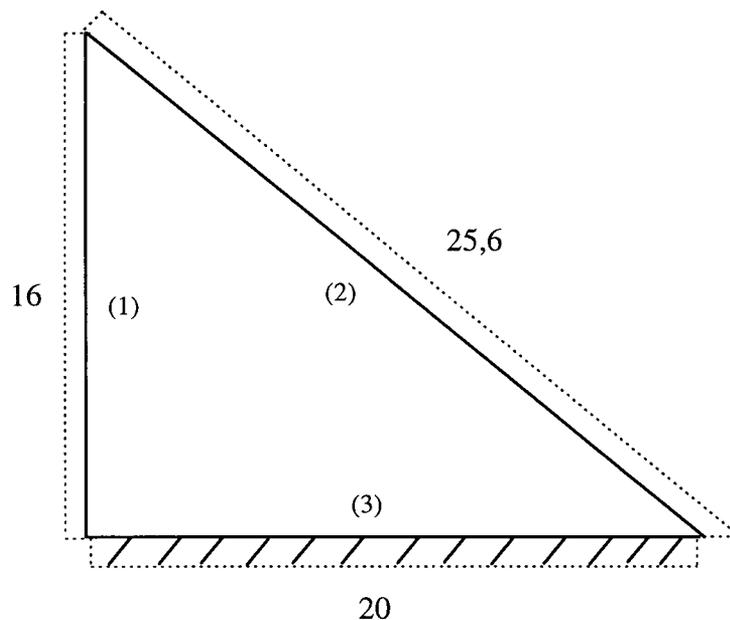


Figura 3.5.b - Domínio geométrico em formato triangular.

Na tabela 3.2 são apresentadas as condições de contorno para a cavidade quadrada, sólido em forma de L e em forma triangular.

Tabela 3.2 - Condições de contorno.

Posições na fig. 3.5	cavidade quadrada	formato L	formato triangular
(1)	$T(0, y) = 0, (0 \leq y < 20)$	$T(0, y) = 0, (0 < y < 50)$	$T(0, y) = 0, (0 < y < 16)$
(2)	$\frac{\partial T}{\partial y}(x, 20) = 0, (0 \leq x \leq 20)$	$\frac{\partial T}{\partial y}(x, 50) = 0, (0 \leq x \leq 50)$	$\frac{\partial T}{\partial y}(x, 10) = 100, (0 < x < 20;$ $y = 16\left(1 - \frac{x}{20}\right))$
(3)	$T(20, y) = 100, (0 < y < 20)$	$\frac{\partial T}{\partial x}(50, y) = 0, (50 \leq y \leq 100)$	$\frac{\partial T}{\partial y}(x, 0) = 0, (0 \leq x \leq 20)$
(4)	$\frac{\partial T}{\partial y}(x, 0) = 0, (0 \leq x \leq 20)$	$T(x, 100) = 100, (50 < x < 100)$	-
(5)	-	$\frac{\partial T}{\partial x}(100, y) = 0, (0 \leq y \leq 100)$	-
(6)	-	$\frac{\partial T}{\partial y}(x, 0) = 0, (0 \leq x \leq 100)$	-

3.7 Conclusão

Neste capítulo, ilustrou-se a possibilidade de resolver as equações de Navier-Stokes, altamente não-lineares, fazendo uso, por exemplo, do método numérico baseado em volumes finitos, com uma configuração de malha co-localizada para todas as variáveis.

No MVF o processo de discretização das equações diferenciais consiste em dividir o domínio físico em um número de volumes de controle não sobrepostos. As equações diferenciais são integradas sobre cada volume de controle. Assumem-se perfis entre os pontos nodais, situados nos centros dos volumes, para avaliar os fluxos convectivos e difusivos através das faces dos volumes.

Dois perspectivas surgem para discretizações do domínio de cálculo: a *discretização estruturada* e a *não-estruturada*, ambas tiveram suas características apresentadas neste capítulo. A malha não-estruturada, adotada neste trabalho, é gerada por DV, que alia a versatilidade das malhas não-estruturadas com a simplicidade das malhas ortogonais.

O esquema SIMPLEC é usado para solucionar o problema do acoplamento pressão-velocidade nas equações de Navier-Stokes. Como esquema de interpolação escolheu-se o da Lei da Potência. O gradiente de pressão será obtido utilizando três formas diferentes de cálculo. Após a discretização das equações obteve-se um sistema de equações algébricas lineares, cujos métodos para resolução serão apresentados no próximo capítulo.

Capítulo 4 - Métodos da família do Gradiente Conjugado

4.1 Introdução

Neste capítulo serão apresentados alguns métodos iterativos não-estacionários: GC, CGS, Bi-CGSTAB, GMRES e o TFQMR, empregados na solução dos sistemas lineares, além de uma descrição abrangente do comportamento de convergência.

4.2 Gradiente Conjugado

Um dos métodos clássicos não-estacionários³ é o método GC, que foi proposto por Hestenes e Stiefel, em 1952, como um método direto. Reid, em 1970, mostrou que o GC poderia ser usado como um método iterativo, com maior velocidade de convergência que *Chebyshev*, e superior ao método SOR, fazendo uso do melhor fator de relaxação (Reid, 1971; Golub and O'Leary, 1989; O'Leary, 1995).

O GC está entre os mais eficientes métodos para solução de sistemas lineares no qual a matriz A de coeficientes (equação (1.1)) é positiva definida e simétrica. Este método é baseado na idéia de minimizar o funcional, que pode ser escrito de formas alternativas, tal como (Campos, 1995),

$$\varphi(x^k) = \frac{1}{2} (x^k)^T A x^k - (x^k)^T b \quad (4.1)$$

$$\varphi(x^k) = \frac{1}{2} (x^k - x^*)^T A (x^k - x^*) \quad (4.2)$$

$$\varphi(x^k) = \frac{1}{2} \left((x^k)^T A x^k - 2b^T x^k + (x^*)^T A x^* \right) \quad (4.3)$$

$$\varphi(x^k) = \frac{1}{2} (r^k)^T A^{-1} r^k \quad (4.4)$$

onde x^* é o vetor solução, r^k é o vetor resíduo do sistema de equações, dado por $r^k = b - Ax^k$ e A é uma matriz positiva definida e simétrica de ordem $n \times n$.

³ Um método iterativo não-estacionário pode ser escrito como $x^{k+1} = G_k x^k + N_k b$, onde G_k , N_k e b são parâmetros que variam de acordo com o processo iterativo, de acordo com Barrett *et al.*, 1994.

O funcional $\varphi(x^k)$ tem um único mínimo global, onde o gradiente se anula, no ponto $x^* = A^{-1}b$, que é a solução do sistema linear $Ax = b$. O valor mínimo do funcional, dado pelo ponto x^* , é $\frac{-b^T A^{-1}b}{2}$. Consequentemente, minimizar o funcional $\varphi(x)$ e resolver o sistema linear $Ax = b$ são problemas equivalentes (Golub and Van Loan, 1996).

A convergência do método, pela minimização da função $\varphi(x^k)$, pode não ser conseguida se a matriz dos coeficientes é semi-definida positiva, porém o método pode convergir para matrizes simétrica indefinida, o mesmo pode não ocorrer se a matriz é assimétrica (Campos, 1995).

Segundo Golub e Van Loan, uma estratégia para minimizar o funcional $\varphi(x)$ é o método dos passos descendentes, onde para qualquer ponto x^k a função $\varphi(x)$ decresce mais rapidamente na direção oposta ao gradiente ($\nabla\varphi(x^k)$), isto é, $-\nabla\varphi(x^k) = -(Ax^k - b)$.

O gradiente negativo é chamado de resíduo de x^k , $r^k = b - Ax^k$. Se o resíduo no ponto x^k é não-nulo, então existe um número α de tal forma que $\varphi(x^k + \alpha r^k) < \varphi(x^k)$. Para x^k e r^k descritos e, desconsiderando o sobrescrito k , obtém-se a seguinte função em α , usando a equação (4.1),

$$f(\alpha) = \varphi(x + \alpha r) = \frac{1}{2}(x + \alpha r)^T A(x + \alpha r) - (x + \alpha r)^T b \quad (4.5)$$

No método dos passos descendentes a função quadrática $f(\alpha)$, apresentada na equação (4.5), assume o seu ponto mínimo quando a primeira derivada em relação a α é nula, $\frac{\partial f(\alpha)}{\partial \alpha} = 0$, desta forma obtém-se o valor da variável independente α , tal como:

$$\alpha = \frac{r^T r}{r^T A r} \quad (4.6)$$

Logo encontra-se o algoritmo GC para minimização do funcional $\varphi(x)$ ou para a solução do sistema linear $Ax = b$, que é frequentemente expresso pelo pseudocódigo descrito na figura 4.1.

Algoritmo Gradiente Conjugado

Atribuir um valor inicial x^0 para a solução x^* .

Faça **para** $i = 1, 2, \dots, n$ (número de equações do sistema)

$$r^0 = b - Ax^0 \quad (4.7)$$

$$p^0 = r^0 \quad (4.8)$$

Fim **para** i

Repita enquanto $k = 1, \dots, k_{\text{máx}}$

Faça **para** $i = 1, 2, \dots, n$

$$\alpha^k = \frac{\left(r^k \right)^T r^k}{\left(p^k \right)^T A p^k} \quad (4.9)$$

$$x^{k+1} = x^k + \alpha^k p^k \quad (4.10)$$

$$r^{k+1} = r^k - \alpha^k A p^k \quad (4.11)$$

$$\beta^k = \frac{\left(r^{k+1} \right)^T r^{k+1}}{\left(r^k \right)^T r^k} \quad (4.12)$$

$$p^{k+1} = r^{k+1} - \beta^k p^k \quad (4.13)$$

Fim **para** i

Testar o critério de convergência, continuar se não convergiu ainda.

$k = k+1$ (incremento do contador de iterações)

Fim **repita**

Figura 4.1 - Algoritmo Gradiente Conjugado.

onde p^k é a direção vetorial de minimização, α^k é o escalar que define a distância de movimento na direção do resíduo e β^k é um escalar.

A versão pré-condicionada do método GC usa um subespaço diferente para a construção iterativa, mas satisfaz as mesmas propriedades de minimização sobre este subespaço. A matriz de pré-condicionamento também deve ser simétrica e positiva definida (Barrett *et al.*, 1994).

Surgem variantes do algoritmo com diferentes escolhas para o cálculo de α , r^k e β no método GC (Reid, 1971; Golub and O'Leary, 1989), por exemplo, o resíduo é calculado direto da definição como,

$$r^{k+1} = b - Ax^{k+1} \quad (4.14)$$

e usam-se as fórmulas alternativas para os parâmetros que seguem,

$$\alpha^k = \frac{\left(r^k \right)^T p^k}{\left(p^k \right)^T A p^k} \quad (4.15)$$

$$\beta^k = -\frac{(r^{k+1})^T A p^k}{(p^k)^T A p^k} \quad (4.16)$$

No apêndice B serão demonstradas matematicamente as equivalências entre estas fórmulas (Reid, 1971; Hageman and Young, 1981) fazendo uso das equações que descrevem o método GC e algumas de suas importantes propriedades, que são resumidas no seguinte teorema (Yang, 1994).

Teorema:

Seja A positiva definida e simétrica, r^i são os vetores resíduos e p^i são os vetores direcional da i-ésima iteração do GC. Então:

- (i) $(p^i, A p^j) = 0$ para $\forall i \neq j$ (*conjugacidade*)
- (ii) $(r^i, A p^j) = 0$ para $\forall i > j \geq 0$ e $i \neq j+1$ (*A-conjugado*)
 $(r^i, r^j) = 0$ para $\forall i > j \geq 0$.

A prova que é feita através de indução é encontrada em (Fletcher, 1976)

Uma versão equivalente do algoritmo GC é formada pela eliminação dos vetores p nas equações do algoritmo da figura 4.1, tal como (Hageman and Young, 1981; Golub and O'Leary, 1989):

$$x^{k+1} = x^{k-1} + \delta^{k+1} \left(\frac{r^k}{\partial^k + x^k - x^{k-1}} \right) \quad (4.17)$$

$$r^{k+1} = r^{k-1} + \delta^{k+1} \left(\frac{-A r^k}{\partial^k + r^k - r^{k-1}} \right) \quad (4.18)$$

onde

$$\delta^{k+1} = \left[1 - \frac{(r^k)^T r^k}{(r^{k-1})^T r^{k-1}} \frac{\partial^{k-1}}{\partial^k} \frac{1}{\delta^k} \right]^{-1} \quad e \quad \partial^k = \frac{(r^k)^T A r^k}{(r^k)^T r^k} \quad (4.19)$$

com $x^{-1} = x^0$ e $\delta^0 = 0$.

4.2.1 Convergência do Gradiente Conjugado

O algoritmo GC converge para aritmética infinita, no máximo em n iterações, sendo n a dimensão da matriz de coeficientes A (equação (1.1)), na maioria das vezes com um tempo de processamento superior ao conseguido com os métodos diretos. A velocidade de convergência do método GC será acelerada se a matriz A tiver t autovalores distintos ($t < n$), então o algoritmo do GC convergirá em, no máximo, t iterações, o que equivale aos autovalores de A estarem agrupados em “clusters”.

Uma previsão precisa da convergência do método GC, assim como os demais métodos iterativos, é difícil de se fazer, contudo uma estimativa da razão de convergência do método GC é limitada em termos do número de condição (κ_2) da matriz de coeficientes A (equação (1.1)) (Barrett *et al.*, 1994). Se x^* é a solução exata do sistema $Ax = b$ e a matriz A é simétrica positiva definida com $\lambda_{\text{maior}} \geq \dots \geq \lambda_{\text{menor}} > 0$, então a estimativa da razão de convergência é denotada por:

$$\|x^k - x^*\|_A \leq 2\alpha^k \|x^0 - x^*\|_A \quad (4.20)$$

onde:

$$\alpha = \frac{(\sqrt{\kappa_2} - 1)}{(\sqrt{\kappa_2} + 1)}$$

$$\kappa_2 = \text{cond}(A) = \frac{\text{maior}|\lambda(A)|}{\text{menor}|\lambda(A)|} \text{ (quociente entre o maior e menor autovalores).}$$

Analisando a expressão para o cálculo de α verifica-se que:

(i) se $\kappa_2=1 \Rightarrow \alpha=0$, isto é, quanto menor o valor da condição κ_2 mais rápida é a velocidade de convergência do método GC, dada pela equação (4.20).

(ii) e $\lim_{\kappa_2 \rightarrow \infty} \alpha = 1$, ou seja, quanto maior o valor da condição κ_2 mais lenta é a velocidade de convergência do método GC.

O método GC envolve um produto vetor-matriz, a atualização de três vetores e dois produtos internos, por iteração, tornando-o atrativo computacionalmente. Uma visão geral, a respeito dos vários artigos publicados nos primeiros 25 anos de existência do método GC, é encontrada em Golub e O’Leary (1989).

No método GC o vetor resíduo satisfaz os três termos recursivos, devido a simetria da matriz de coeficientes, sendo esta uma característica *poderosa* e atraente do método. No entanto, não é possível manter estas propriedades para sistemas lineares assimétricos, surgindo diversas

formas de trabalhar esta situação, entre elas, consideram-se as equações normais no lugar das equações do sistema original (Yang, 1994).

Nas equações normais a matriz de coeficientes do sistema linear é multiplicada pela sua transposta, desta forma a matriz de coeficientes do novo sistema é positiva definida e simétrica, e o algoritmo GC pode ser aplicado. A desvantagem de usar estas equações é que o número de condição cresce significativamente. Logo, nestes algoritmos a convergência esperada é muito lenta, tornando-se importante pré-condicionar o sistema. Contudo, não é necessário modificar os sistemas assimétricos para solucioná-los pois existem vários métodos para resolvê-los, sem considerar as equações normais.

As matrizes, dos sistemas lineares, quando assimétricas podem ser resolvidas por métodos tais como o Bi-CG, que resolve adequadamente uma grande classe de equações onde outros métodos podem parar ou até mesmo oscilar. O Bi-CG, desta forma, poderia ser visto como um método ideal, porém ele tem algumas desvantagens (Sleijpen and Fokkema, 1993):

- (i) a transposta (complexa ou real) de A é frequentemente inviável, um dos fatores que podem ser responsáveis por esta inviabilidade é a forma como as variáveis são armazenadas.
- (ii) o custo computacional é baixo em termos de produto vetor-vetor, porém cada passo iterativo requer duas multiplicações com matrizes, tendo o dobro de custo do CG;
- (iii) frequentemente converge com irregularidades, o que pode tornar lenta a convergência ou até mesmo parar sem ter encontrado a solução.

Deste modo, o Bi-CG não será analisado neste trabalho. Alguns dos métodos iterativos não-estacionários que solucionam sistemas assimétricos são descritos nas seções seguintes.

4.3 CGS

O método CGS é semelhante ao GC e tem origem no método Bi-CG, proposto por Sonneveld em 1989, resolve sistemas de equações lineares reais assimétricas não-positivas. A convergência é geralmente mais rápida que o método Bi-CG evitando o cálculo da transposta da matriz de coeficientes A (Sonneveld, 1989).

O CGS pode ser visto como um método no qual o operador “contração” do Bi-CG é aplicado duas vezes, o que pode reduzir o resíduo inicial r^0 e o vetor resíduo r^k , aumentando a sua velocidade de convergência (Barrett *et al.*, 1994). Uma descrição algorítmica do método é dada na figura 4.2.

Algoritmo CGS

Atribuir um valor inicial x^0 para a solução x^* .

Faça **para** $i=1, 2, \dots, n$ (número de equações do sistema linear)

$$r^0 = b - Ax^0$$

$$\tilde{r}^0 = r^0, \text{ tal que } (r^0, \tilde{r}^0) \neq 0 \quad q^0 = p^0 = 0 \quad \rho^0 = 1$$

Fim **para** i

Repita para $k = 1, \dots, k_{\text{máx}}$

Faça **para** $i=1, 2, \dots, n$

$$\rho^k = (\tilde{r}^0)^T r^{k-1}$$

$$\beta^k = \frac{\rho^k}{\rho^{k-1}}$$

$$u^k = r^{k-1} + \beta^k q^{k-1}$$

$$p^k = u^k + \beta^k (q^{k-1} + \beta^k p^{k-1})$$

$$v^k = Ap^k$$

$$\sigma^k = (\tilde{r}^0)^T v^k$$

$$\alpha^k = \frac{\rho^k}{\sigma^k}$$

$$q^k = u^k - \alpha^k v^k$$

Calcular o novo resíduo e o novo valor para a solução

$$r^k = r^{k-1} - \alpha^k A(u^k + q^k)$$

$$x^k = x^{k-1} + \alpha^k (u^k + q^k)$$

Fim **para** i

Testar o critério de convergência, continuar se necessário

$k = k+1$ (incremento do contador de iterações)

Fim **repita**

Figura 4.2 - Algoritmo CGS

4.3.1 Convergência do CGS

Cada passo iterativo do algoritmo CGS requer duas vezes a quantidade de operações necessárias pelo algoritmo GC, e o mesmo número de operações por iteração que o Bi-CG, mas não envolve o cálculo com A^T só utilizando multiplicações com a matriz A . Em circunstâncias onde o cálculo com A^T é impraticável o CGS torna-se atrativo (Deng *et al.*, 1994).

Frequentemente observa-se uma velocidade de convergência duas vezes mais rápida no CGS do que no Bi-CG, porém o CGS pode ter uma convergência altamente irregular ocasionando em perdas da precisão na solução (Barrett *et al.*, 1994). Para matrizes gerais, o

algoritmo da figura 4.2 pode terminar prematuramente com $\sigma_k = 0$ ou $\rho_k = 0$, sem a solução da equação (1.1) ser obtida. Uma continuação do processo iterativo leva a uma divisão por zero, e conseqüentemente uma parada do algoritmo (Freund, 1993).

4.4 Bi-CGSTAB

O método Bi-CGSTAB, proposto por Van der Vorst em 1992, foi desenvolvido para resolver sistemas lineares assimétricos, evitando os padrões de *frequentes irregularidades* na convergência do método CGS, para isto o Bi-CGSTAB utiliza um tipo de minimização local do vetor resíduo, isto é, ao menos localmente o vetor resíduo é minimizado, o qual conduz a uma convergência suave (Van der Vorst, 1992).

No algoritmo que segue calcula-se ω^k e o resíduo $r^k = Q_k(A)P_k(A)r_i^0$ é minimizado na norma-2 como uma função de ω^k , onde Q_k e P_k são polinômios em A de grau k , maiores detalhes são encontrados em Van der Vorst (1992). A figura 4.3 descreve um algoritmo para o método Bi-CGSTAB.

Algoritmo Bi-CGSTAB

Atribuir um valor inicial x^0 para a solução x^* .

Faça **para** $i=1, 2, \dots, n$ (número de equações do sistema linear)

$$r^0 = b - Ax^0; \tilde{r}^0 = r^0, \text{ tal que } (r^0, \tilde{r}^0) \neq 0; v^0 = p^0 = 0; \alpha^0 = \omega^0 = \rho^0 = 1$$

Fim **para** i

Repita para $k = 1, \dots, k_{\text{máx}}$

Faça **para** $i=1, 2, \dots, n$

$$\rho^k = (\tilde{r}^0)^T r^{k-1}; \beta^k = (\rho^k / \rho^{k-1})(\alpha^{k-1} / \omega^{k-1}); p^k = r^{k-1} + \beta^k (p^{k-1} + \omega^{k-1} v^{k-1})$$

$$v^k = Ap^k; \alpha^k = \rho^k / \left((\tilde{r}^0)^T v^k \right)$$

$$s^k = r^{k-1} - \alpha^k v^k; t^k = As^k; \omega^k = (t^k)^T s^k / (t^k)^T t^k$$

Calcular o novo valor para a solução

$$x^k = x^{k-1} + \alpha^k p^k + \omega^k s^k$$

Fim **para** i

Testar o critério de convergência, continuar se necessário

$k = k+1$ (incremento do contador de iterações)

$$r^k = s^k - \omega^k t^k$$

Fim **repita**

Figura 4.3 - Algoritmo Bi-CGSTAB.

4.4.1 Convergência do Bi-CGSTAB

O Bi-CGSTAB requer, para a solução do sistema (1.1), a avaliação de dois produtos matriz-vetor e quatro produtos internos, isto é, dois produtos internos a mais que o Bi-CG e o CGS. Em situações práticas, contudo, os dois produtos internos adicionais levam a um pequeno crescimento do *esforço* computacional, o que é prontamente compensado pela redução no número de passos iterativos, especialmente em computadores vetoriais em que os produtos internos são, usualmente, operações rápidas.

Além da locação de memória para x , b , r e A o Bi-CGSTAB necessita de espaço para quatro vetores adicionais de dimensão n , sejam: \tilde{r}^0 , p , v e t , nota-se que o vetor r pode ser sobrescrito pelo vetor s , pois ambos são usados em situações distintas (Van der Vorst, 1992; Deng *et al.*, 1994).

Pela propriedade de ortogonalidade $(P_k(A)r^0, Q_k(A^T)\tilde{r}^0) = 0$, para $j < i$, segue que Bi-CGSTAB é, também, um método finito, isto é, em aritmética exata ele pode convergir no máximo $t \leq n$ passos iterativos. Neste caso tem-se $s_i = 0$ no t -ésimo passo iterativo e o vetor ω^t é então indefinido, isto representa um critério de parada do algoritmo. Para evitar este tipo de parada existem algoritmos com dois testes de convergência, como encontrado em Barrett *et al.* (1994), no primeiro teste é checada a norma de s_i e no segundo é necessário $\omega \neq 0$, para continuar o processo iterativo. Para maiores detalhes sobre a análise de convergência do CGS e Bi-CGSTAB pode-se recorrer ao trabalho de Fujino e Zhang (Fujino and Zhang, 1992).

Os problemas de convergência com o método Bi-CGSTAB, em particular, podem ocorrer pela escolha do valor não apropriado para o vetor \tilde{r}^0 , pelo parâmetro ρ ou (\tilde{r}^0, v^k) serem nulos (ou muito pequenos). Nestas situações deve-se recomeçar com um valor diferente para o vetor \tilde{r}^0 ou escolher outro método. Contudo existem algumas variantes do método, que são equivalentes e têm comportamento diferente em precisão aritmética finita. Uma variante é obtida pelas seguintes mudanças no código da figura 4.3 (Van der Vorst, 1992).

(i) incluir $\rho^1 = (\tilde{r}^0, r^0)$ na parte inicial do algoritmo;

(ii) omitir o cálculo de ρ^k na primeira linha da parte iterativa;

(iii) adicionar o cálculo de $\rho^{k+1} = -\omega^k (\tilde{r}^0, t^k)$ imediatamente depois do cálculo de ω^k .

O algoritmo do método Bi-CGSTAB com as modificações apresentadas anteriormente é descrito na figura 4.4.

Algoritmo Bi-CGSTAB modificado

Atribuir um valor inicial x^0 para a solução x^* .

Faça **para** $i=1, 2, \dots, n$ (número de equações do sistema linear)

$$r^0 = b - Ax^0$$

$$\tilde{r}^0 = r^0, \text{ tal que } (r^0, \tilde{r}^0) \neq 0$$

$$v^0 = p^0 = 0$$

$$\alpha^0 = \omega^0 = 1$$

$$\rho^0 = 1$$

$$\rho^1 = (\tilde{r}^0, r^0)$$

Fim **para** i

Repita para $k = 1, \dots, k_{\text{máx}}$

Faça **para** $i=1, 2, \dots, n$

se ($k \neq 1$)

$$\rho^k = -w^{k-1}(\tilde{r}^0, t^{k-1})$$

fim se

$$\beta^k = (\rho^k / \rho^{k-1})(\alpha^{k-1} / \omega^{k-1})$$

$$p^k = r^{k-1} + \beta^k (p^{k-1} + \omega^{k-1} v^{k-1})$$

$$v^k = Ap^k$$

$$\alpha^k = \rho^k / \left((\tilde{r}^0)^T v^k \right)$$

$$s^k = r^{k-1} - \alpha^k v^k$$

$$t^k = As^k$$

$$\omega^k = (t^k)^T s^k / (t^k)^T t^k$$

Calcular o novo valor para a solução

$$x^k = x^{k-1} + \alpha^k p^k + \omega^k s^k$$

Fim **para** i

Testar o critério de convergência, continuar se necessário

$k = k+1$ (incremento do contador de iterações)

$$r^k = s^k - \omega^k t^k$$

Fim **repita**

Figura 4.4 - Algoritmo Bi-CGSTAB modificado.

Existem outras variantes do método, que podem ser mais apropriadas a determinadas classes de problemas. Algumas investigações devem ser levadas em consideração, ao se adotar qualquer variante.

O Bi-CGSTAB(l), é outro método iterativo atrativo para solução de sistemas lineares assimétricos, quando $l=1$ o método encontra a mesma aproximação, para o x^k , que o Bi-CGSTAB. O método combina as vantagens do Bi-CG e o GMRES(m) e converge mais rápido que qualquer um deles, o trabalho computacional e o requerimento de memória são modestos com relação ao Bi-CG, CGS e Bi-CGSTAB, contudo este método não será analisado neste trabalho (Sleijpen and Fokkema, 1993).

4.5 TFQMR

O método TFQMR, proposto por R. Freund em 1993, resolve sistemas lineares cuja matriz de coeficientes é não-singular e não-Hermitiana. Quase minimiza a norma resíduo obtida pelo CGS, é uma variante do Bi-CG livre do cálculo da transposta, razão pela qual é chamado “transpose-free”. O TFQMR é implementado, facilmente, pela mudança em poucas linhas do algoritmo CGS. Enquanto a norma do resíduo CGS tipicamente oscila, o TFQMR demonstra um comportamento de convergência regular, mas *paradas* podem ser possíveis (Freund, 1993; Lin *et al.*, 1995).

Em contraste ao CGS, o TFQMR cria duas iterações por passo, explorando consequentemente todas as direções viáveis. Cada passo iterativo de ambos algoritmos envolvem somente dois produtos vetor-matriz utilizando a matriz de coeficientes A . O algoritmo apresentado na figura 4.5 é semelhante ao CGS sendo efetuadas apenas algumas modificações, este algoritmo é formado por pesos gerais $\omega > 0$. (Bücker and Basermann, 1994).

Segundo Freund pode-se escolher ω como sendo:

$$\omega^k = \begin{cases} \|r^k\|, & \text{se } m = 2n + 1 \text{ (impar)} \\ \sqrt{\|r^{k-1}\| \cdot \|r^k\|}, & \text{se } m = 2n \text{ (par)} \end{cases}$$

Semelhante ao básico CGS o TFQMR pode parar prematuramente, porém estas paradas são muito raras na prática, mais detalhes sobre o comportamento de convergência do método podem ser encontrados em Freund (1993).

Algoritmo TFQMR

Atribuir um valor inicial x^0 para a solução x^* .

Faça **para** $i=1, 2, \dots, n$ (número de equações do sistema linear)

$$r^0 = b - Ax^0 \quad \tilde{r}^0 = r^0, \text{ tal que } (r^0, \tilde{r}^0) \neq 0$$

$$u^0 = p^0 = r^0 \quad v^0 = Ap^0 \quad d^0 = 0 \quad \rho^0 = (r^0, \tilde{r}^0)$$

Fim **para** i

$$\tau^0 = \omega^1, \vartheta^0 = \eta^0 = 0$$

Repita para $k = 1, \dots, k_{\text{máx}}$

Faça **para** $i=1, 2, \dots, n$

$$\sigma^{k-1} = (\tilde{r}^0, v^{k-1})$$

$$\alpha^{k-1} = \frac{\rho^{k-1}}{\sigma^{k-1}}$$

$$q^k = u^{k-1} - \alpha^{k-1} v^{k-1}$$

$$r^k = r^{k-1} - \alpha^{k-1} A(u^{k-1} + q^k)$$

Fim **para** i

Faça **para** $m = 2k-1; 2k$.

Faça **para** $i=1; 2, \dots, n$

$$\vartheta^k = \omega^{k+1}/\tau^{k-1}$$

$$c^k = \frac{1}{\sqrt{1 + (v^k)^2}}$$

$$\tau^k = \tau^{k-1} \vartheta^k c^k$$

$$\eta^k = (c^k)^2 \alpha^{k-1}$$

$$d^k = y^k + ((\vartheta^{k-1})^2 \eta^{k-1}/\alpha^{k-1}) d^{k-1}$$

$$\text{onde } y^k = \begin{cases} u^{k-1}, & \text{se } m = 2n - 1 \\ q^k, & \text{se } m = 2n \end{cases}$$

$$x^k = x^{k-1} + \eta^k d^k$$

Fim **para** i

Fim **para** m

Testar o critério de convergência, continuar se necessário

Faça **para** $i=1; 2, \dots, n$

$$\rho^k = (r^0, \tilde{r}^0)$$

$$\beta^k = \frac{\rho^k}{\rho^{k-1}}$$

$$u^k = r^k + \beta^k q^k$$

$$p^k = u^k + \beta^k (q^k + \beta^k p^{k-1})$$

$$v^k = Ap^k$$

Fim **para** i

$k = k+1$ (incremento do contador de iterações)

Fim **repita**

Figura 4.5 - Algoritmo TFQMR.

4.6 GMRES

O método GMRES, proposto por Saad e Schultz em 1986, tem a propriedade de minimizar a cada passo iterativo a norma do vetor resíduo $\|b - A(x^0 + z^k)\|$ onde z é um membro do subespaço de Krylov $K_k = \{r^0, Ar^0, \dots, A^{k-1}r^0\}$. Na prática a dimensão k do subespaço de Krylov é determinada no sentido de reduzir o custo do algoritmo e restringir o espaço de memória utilizado. No processo de minimização busca-se uma base ortonormal do subespaço de Krylov e, posteriormente, resolve-se um problema de mínimos quadrados (Marcondes, 1996).

O algoritmo é derivado do processo de Arnoldi para construção de uma base l_2 -ortogonal do subespaço de Krylov, pode ser considerado como uma generalização do método MINRES e é teoricamente equivalente ao GC nas equações normal (GCR) e ORTHODIR, apresentando algumas mudanças nestes dois últimos. Os resultados deste método correspondem a utilização da ortogonalização de Gram-Schmidt modificada, sucedida por rotações de Givens no plano.

Nos sistemas em que a matriz não é positiva real ou é assimétrica, provém a principal motivação para o desenvolvimento do GMRES. O GMRES requer só a metade da armazenagem requerida pelo método GCR e menos que 1/3 das operações aritméticas (Saad and Schultz, 1986).

No GMRES todos os vetores calculados, previamente, na seqüência ortogonal têm que ser armazenados, e por esta razão, versões recomeçadas do método são usadas. Quando o número de iterações k cresce o requerimento de armazenagem para o número de vetores tem crescimento semelhante a k e o número de multiplicações semelhante a $(1/2)k^2n$.

Para remediar esta dificuldade, pode-se usar o algoritmo iterativamente, isto é, pode-se recomeçar o algoritmo a cada m passos, onde m é algum parâmetro inteiro fixo. A versão recomeçada do GMRES denotada por GMRES(m) é descrita na figura 4.6.

Algoritmo GMRES

Atribuir um valor inicial x^0 para a solução x^* .

passo 1) Inicialização

Faça **para** $i=1, 2, \dots, n$ (número de equações do sistema linear)

$$r^0 = b - Ax^0; \quad v^0 = r^0 / \|r^0\|; \quad e^1 = \|r^0\|$$

Fim **para** i

passo 2) Ortogonalização de Gram-Schmidt Modificado

Faça **para** $k=1, \dots, m$

Faça **para** $i = 1, \dots, k$

$$h_{i,k} = (Av^k, v^i)$$

Fim **para** i

$$\bar{v}^{k+1} = Av^k - \sum_{i=1}^k h_{i,k} v^i$$

$$h_{k+1,k} = \|\bar{v}^{k+1}\|$$

$$v^{k+1} = \bar{v}^{k+1} / h_{k+1,k}$$

$$\bar{h}^k = \{h_{1,k}, \dots, h_{k+1,k}\}^T$$

passo 3) Algoritmo QR (rotações de Givens)

Faça **para** $i=1, 2, \dots, k-1$

$$\begin{Bmatrix} h_{i,k} \\ h_{i,k+1} \end{Bmatrix} \leftarrow \begin{bmatrix} c_i & s_i \\ -s_i & c_i \end{bmatrix} \begin{Bmatrix} h_{i,k} \\ h_{i,k+1} \end{Bmatrix}$$

$$\gamma = [h_{k,k}^2 + h_{k,k+1}^2]^{1/2}$$

$$c^k = h_{k,k} / \gamma$$

$$s^k = h_{k,k+1} / \gamma$$

Rotação aplicada ao \bar{h}

$$h_{k,k} \leftarrow \gamma$$

$$h_{k,k+1} \leftarrow 0$$

Rotação aplicada ao vetor e

$$e^{k+1} \leftarrow -s^k e^k \quad e^k \leftarrow -c^k e^k$$

Se $|e^{k+1}| / |e^1| < \text{tolerância}$ então saia do laço k

passo 4) Resolver y

$$\begin{Bmatrix} y_1 \\ M \\ y_k \end{Bmatrix} \leftarrow \begin{bmatrix} h_{1,1} & \Lambda & h_{k,k} \\ M & M & M \\ 0 & \Lambda & h_{k,k} \end{bmatrix}^{-1} \begin{Bmatrix} e_1 \\ M \\ e_k \end{Bmatrix}$$

passo 5) Resolver x

$$x = x^0 + V^m y^m = x^0 + v_1 y_1 + \dots + v_m y_m$$

Se convergiu pare for k , senão

passo 6) Recomeçar

$$r^k = b - Ax^k, \quad x^0 = x^k, \quad v^1 = r^k / \|r^k\|$$

Volte ao **passo 2**

Figura 4.6 - Algoritmo GMRES recomeçado.

4.6.1 Convergência do GMRES

A solução x^j produzida pelo GMRES no passo j -ésimo é exata se, e somente se, as quatro seguintes condições, equivalentes, são satisfeitas:

- (i) o algoritmo para no passo j ;
- (ii) $\beta^{j+1} = 0$;
- (iii) $h_{j+1,j} = 0$;
- (iv) o grau do polinômio minimal do vetor resíduo inicial r^0 é igual ao j .

O laço iterativo constrói o espaço de Krylov e projeta o sistema de equações original para este espaço, produzindo a matriz superior de Hessenberg $H = \{h_{ij}\}$. A ortogonalização de Gram-Schmidt, dos vetores bases, é tipicamente a parte mais dispendiosa computacionalmente do algoritmo, para valores moderados de m (Behr and Tezduyar, 1994).

O GMRES, como os demais métodos iterativos, é extremamente sensível às propriedades numéricas da matriz de coeficientes do sistema.

4.7 Comparativo entre os métodos não-estacionários

Os métodos iterativos não-estacionários apresentados, não resolvem eficientemente todos os tipos de problemas, logo o conhecimento das propriedades da matriz de coeficientes é um dos principais critérios para seleção de um método iterativo, que seja eficiente para resolução do problema em estudo. A tabela 4.1 apresenta o número aproximado de operações para a iteração k em cada um dos métodos iterativos analisados.

Tabela 4.1 - Operações aproximadas dos métodos iterativos para a iteração k .

métodos	(x,y)	$y+\alpha x$	$y=Ax$	$y=P^{-1}x$
GC	2	3	1	1
CGS	2	6	2	2
Bi-CGSTAB	4	6	2	2
TFQMR	2	6	2	2
GMRES	$k+1$	$k+1$	1	1

Na tabela 4.1 a expressão (x,y) representa o produto interno entre dois vetores, $y+\alpha x$ representa a soma entre vetores após um deles ter sido multiplicado por um escalar, e as duas últimas colunas da tabela apresentam produtos de matriz por vetor.

O número de operações, apresentado em cada passo iterativo dos métodos, é responsável pelo custo de processamento individual, logo esta análise é também importante na escolha do método mais adequado pelo usuário. Nota-se na tabela 4.1 que o método GC é o que possui menor número de operações por laço iterativo, o CGS e o TFQMR apresentam o mesmo número de operações, sendo quase o dobro do GC, a menos no número de produtos internos. O Bi-CGSTAB apresenta dois produtos internos a mais que os dois métodos anteriormente citados, já no GMRES o número de (x, y) e $x+\alpha y$ aumentam de acordo com o número de iterações, motivo pelo qual são utilizadas versões recomçadas deste método.

4.8 Conclusão

Neste capítulo apresentou-se alguns métodos iterativos não-estacionários para a resolução dos sistemas lineares simétricos e assimétricos, resultantes da discretização de EDPs no método dos volumes finitos.

O método GC é aplicável para sistemas cuja matriz é simétrica positiva definida, a velocidade de convergência é dependente do número de condição, e podem ser facilmente paralelizáveis. Os demais métodos da família do GC, apresentados neste capítulo, são aplicados na resolução de sistemas cuja matriz é assimétrica podendo ser indefinida, para alguns métodos.

O método CGS melhora a convergência do Bi-CG e não requer o cálculo da transposta da matriz de coeficientes, mas a quantidade de memória utilizada é comparável ao Bi-CG. O Bi-CGSTAB similar a ambos, Bi-CG e CGS, evita os padrões de convergência irregular do último, além de não usar a transposta da matriz de coeficientes.

No método GMRES o resíduo torna-se pequeno em um número fixo de passos iterativos, mas estes passos causam aumento de processamento e armazenamento, assim versões recomçadas do método são uma necessidade (Hribersek and Skerget, 1996). No presente trabalho o método TFQMR apresentou convergência lenta, se comparado aos demais métodos, o que poderá ser verificado no capítulo 8.

Capítulo 5 - Pré-condicionamento

5.1 Introdução

Os métodos iterativos podem ter sua velocidade de convergência acelerada através do uso de pré-condicionadores. Desta forma o sistema linear $Ax = b$ é transformado em outro sistema equivalente, que possui a mesma solução e o número de condição espectral e distribuição dos autovalores melhorada, de tal forma que ficam em “clusters” ou se aproximam da unidade.

A eficiência do pré-condicionador depende do quanto P^{-1} , a inversa da matriz de pré-condicionamento, aproxima-se de A^{-1} . Os pré-condicionamentos descritos neste capítulo são: diagonal, SSOR, por fatoração de Cholesky incompleta, por Série truncada de Maclaurin, por Série truncada de Neumann e polinomial.

5.2 Pré-condicionamento explícito e implícito

A velocidade de convergência dos métodos iterativos, em especial da família do GC, pode ser melhorada fazendo uso de um pré-condicionador apropriado. O objetivo do pré-condicionador é reduzir o número de condição κ_2 do sistema $Ax = b$ e isto é possível por meio de uma transformação de congruência na matriz de coeficientes A , da forma que segue (Barrett *et al.*, 1994),

$$P^{-1}Ax = P^{-1}b \quad (5.1)$$

que, pela completa consistência tem a mesma solução da equação (1.1). O sistema equivalente será resolvido por,

$$Cv = u \quad (5.2)$$

com

$$C = P^{-1}A \quad v = x \quad u = P^{-1}b \quad (5.3)$$

onde P é a matriz de pré-condicionamento.

O objetivo do pré-condicionamento é reduzir o número de condicionamento, ou condição espectral da matriz A , o quanto possível. Se $P=A$ tem-se a melhor escolha para P , pois $\kappa_2(A)=\kappa_2(P^{-1}A)=1$, porém, desta forma a solução é calculada diretamente, através da inversa da matriz de coeficientes A , aumentando a complexidade algorítmica.

O pré-condicionador mais simples é uma matriz diagonal, que muitas vezes não reduz o número de condicionamento tornando-se inadequado, e do outro lado matrizes de pré-condicionamento complicadas podem destruir a esparsidade da matriz de coeficientes original (Borges *et al*, 1996). Assim a escolha do pré-condicionador é importante para o sucesso do sistema equivalente e, de acordo com Campos (1995), deve ter algumas características desejáveis:

- (i) a matriz de pré-condicionamento P deve ser facilmente construída;
- (ii) os autovalores de $[P^{-1}A]$ devem estar próximos de um ou em grupos;
- (iii) se A é esparsa P deve ser esparsa como A ;
- (iv) P deve ser uma boa aproximação da matriz de coeficientes;

As três diferentes maneiras de aplicar o pré-condicionamento são:

- (i) a *orientação dos dois lados*, à esquerda e à direita, também denominado pré-condicionamento dividido ou simétrico, organizado como, $[P_E^{-1}AP_D^{-1}][P_Dx]=[P_E^{-1}b]$;
- (ii) a *orientação à esquerda* $[P_E^{-1}Ax]=[P_E^{-1}b]$; e
- (iii) a *orientação à direita* $[AP_D^{-1}y]=b \Leftrightarrow y=[P_Dx]$.

onde P_E é a matriz de pré-condicionamento à esquerda e P_D é a matriz de pré-condicionamento à direita, desta forma a matriz de pré-condicionamento P fica dividida em $P = P_E P_D$. Os pré-condicionamentos à esquerda e à direita são considerados casos especiais, podendo ser representados conforme figura 5.1.

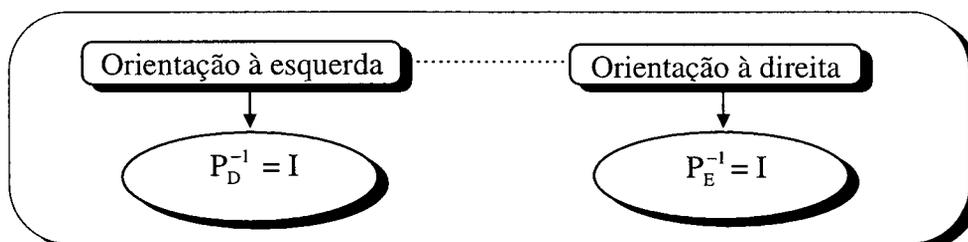


Figura 5.1 - Representação do pré-condicionamento à direita e à esquerda.

Na aplicação de um pré-condicionamento obtém-se a solução de um sistema linear permutado, ou transformado, que pode ser resolvido com maior rapidez. O sistema linear $Ax = b$

pode ser pré-condicionado implicitamente (Campos and Rollet, 1995), ou seja, o vetor resíduo é modificado, assim na k -ésima iteração $(r_i, r_j) \neq 0$ para $\forall i \neq j$, e $(r_i, P^{-1}r_j) = 0$ para $\forall i \neq j$. Tal sistema resolvido pelo método GC com pré-condicionamento implícito resulta no algoritmo mostrado na figura 5.2.

valores iniciais $\{x^0, A, b, \xi, k_{\max}\}$
 $r^0 = b - Ax^0$; $p^0 = 0$; $\rho^0 = 1$; $k = 0$
 repita enquanto{
 $k = k+1$; $z^k = P^{-1}r^k$; $\rho^k = (r^{k-1}, z^{k-1})$;
 $\beta^k = \rho^k/\rho^{k-1}$; $p^k = z^{k-1} + \beta^k p^{k-1}$;
 $\alpha^k = \rho^k/(p^k, Ap^k)$
 $x^k = x^{k-1} + \alpha^k p^k$; $r^k = r^{k-1} - \alpha^k Ap^k$
 } $\|r^k\|_2/\|r^0\|_2 < \xi$ ou $k = k_{\max}$

Figura 5.2 - Gradiente Conjugado pré-condicionado implicitamente.

Particularmente quando A é simétrica positiva definida o sistema linear $Ax = b$, fazendo uso do GC, pode ser pré-condicionado explicitamente, ou seja, no lugar do resíduo, a matriz dos coeficientes que é modificada. Desta forma a equação (1.1) é resolvida pelo algoritmo GC apresentado na figura 5.3.

valores iniciais $\{x^0, A, b, \xi, k_{\max}\}$
 $r^0 = P_E^{-1/2} (b - AP_E^{-T/2} x^0)$
 $p^0 = 0$; $\rho^0 = 1$; $k = 0$
 repita enquanto {
 $k = k+1$; $\rho^k = (r^{k-1}, r^{k-1})$; $\beta^k = \rho^k/\rho^{k-1}$
 $p^k = r^{k-1} + \beta^k p^{k-1}$;
 $\alpha^k = \rho^k/(p^k, P_E^{-1/2} AP_E^{-T/2} p^k)$
 $x^k = x^{k-1} + \alpha^k p^k$;
 $r^k = r^{k-1} - \alpha^k P_E^{-1/2} AP_E^{-T/2} p^k$
 } $\|r^k\|_2/\|r^0\|_2 < \xi$ ou $k = k_{\max}$
 $x^k = P_E^{-T/2} x^k$

Figura 5.3 - Gradiente Conjugado pré-condicionado explicitamente.

No caso simétrico a matriz $P_D = P_E^T$, assim mantém-se a simetria necessária da matriz dos coeficientes. Cabe ressaltar, que alguns trabalhos usam a matriz de pré-condicionamento tal como P^{-T} e P^{-1} , assim deve-se ter cuidado ao adotá-las e observar a forma como o autor usa o pré-condicionamento em seu trabalho (Ajiz and Jennings, 1984; Papadrakakis, 1991). Os tipos mais conhecidos de pré-condicionadores serão descritos nos itens a seguir.

5.3 Pré-condicionamento diagonal

O pré-condicionamento diagonal⁴ ou Jacobi, é uma das formas mais simples de pré-condicionar um sistema linear (Barrett *et al.*, 1994; Dracopoulos and Crisfield, 1995). Obtido usando-se os recíprocos da diagonal da matriz de coeficientes A , de acordo com a seguinte formulação,

$$\begin{aligned}
 (i) \text{ à esquerda - } P_E &= D = \text{diag}(a_{ii}) \\
 (ii) \text{ à direita - } P_D &= D = \text{diag}(a_{ii}) \\
 (iii) \text{ simétrico - } P_D &= P_E = D^{1/2} = \text{diag}(\sqrt{a_{ii}})
 \end{aligned}
 \tag{5.4}$$

No caso especial da técnica de escalonamento à esquerda tem-se:

$$[D^{-1}A]x = [D^{-1}b] \Rightarrow Cv = u \tag{5.5}$$

A matriz de coeficientes C é a matriz do sistema transformado pelo pré-condicionamento e tem a diagonal da matriz formada por elementos unitários.

5.4 Pré-condicionamento por “Symmetric Successive Overrelaxation”

O método “Symmetric Successive Overrelaxation” (SSOR) foi apresentado em 1955 por Sheldon e após foi usado como pré-condicionamento para o método GC. O método SSOR representa a multiplicação por uma matriz simétrica, o que não ocorre no SOR. Segundo o que foi proposto por Campos (1995) um método iterativo pode ser escrito na forma:

$$x^{k+1} = x^k + e^{k+1} \tag{5.6}$$

onde e^{k+1} é o vetor diferença entre a solução atual e a anterior no processo iterativo.

$$P(x^{k+1} - x^k) = r^k \tag{5.7}$$

ou

$$Pe^{k+1} = r^k = b - Ax^k \tag{5.8}$$

Tem-se,

⁴ Em certos casos verifica-se que o pré-condicionamento de Jacobi é ótimo ou próximo do ótimo, reduzindo o número de condição. Isto é verificado nos trabalhos de Forsythe e Strauss (1955) para matrizes com propriedades A e por Van der Sluis (1969) para matrizes com esparsidade geral, conforme encontrado em Barrett *et al.*, 1994.

$$\begin{aligned}
 x^{k+1} - x^k &= P^{-1} (b - Ax^k) \\
 x^{k+1} &= x^k + P^{-1} (b - Ax^k) \\
 x^{k+1} &= (I - P^{-1}A)x^k + P^{-1}b
 \end{aligned}
 \tag{5.9}$$

Mostrando que o alcance de convergência é dado pela condição da matriz $(I - P^{-1}A)$.
Dividindo a matriz A em

$$A = D + B_L + B_U \tag{5.10}$$

onde D é formada pelos elementos diagonal de A , B_L é a matriz triangular estritamente inferior formada pelos elementos de A com $l_{ii} = 0 \forall i$ e B_U é a matriz triangular superior formada pelos elementos de A , então a equação da técnica clássica SOR, é tal que,

$$(D + \omega B_L)x^{k+1} = ((1-\omega)D - \omega B_U)x^k + \omega b \tag{5.11}$$

O método SSOR é um processo de dois passos iterativos, o primeiro passo é SOR ascendente. O segundo passo consiste do SOR “backward” em que as equações são feitas na ordem reversa tal como,

$$(D + \omega B_U)x^{k+1} = ((1-\omega)D - \omega B_L)x^k + \omega b \tag{5.12}$$

O pré-condicionamento usado na equação (5.9) é o explícito, logo o método SSOR escrito como pré-condicionador explícito tem as duas etapas seguintes:

(i) primeira etapa

Fazendo uso das equações (5.8) e (5.11)

$$(D + \omega B_L)x^{k+1/2} = ((1-\omega)D - \omega B_U)x^k + \omega(r^k + Ax^k) \tag{5.13.a}$$

Aplicando a equação (5.10) na equação (5.13.a) encontra-se:

$$(D + \omega B_L)(x^{k+1/2} - x^k) = \omega r^k \tag{5.13.b}$$

(ii) segunda etapa

Fazendo uso das equações (5.8) e (5.12) obtém-se:

$$(D + \omega B_U)x^{k+1} = ((1 - \omega)D - \omega B_L)x^{k+1/2} + \omega(r^{k+1/2} + Ax^{k+1/2}) \quad (5.14.a)$$

Aplicando a equação (5.10) na equação (5.14.a) encontra-se:

$$(D + \omega B_U)(x^{k+1} - x^{k+1/2}) = \omega(b - Ax^{k+1/2}) \quad (5.14.b)$$

Para eliminar $x^{k+1/2}$ primeiro isola-se x^{k+1} na equação (5.13.b), obtendo a equação que segue,

$$(D + \omega B_U)x^{k+1} = (D + \omega B_U - \omega A)x^{k+1/2} + \omega b \quad (5.15)$$

Aplicando as equações (5.7), (5.8) e (5.13.b) na equação (5.15) vem,

$$(D + \omega B_U)x^{k+1} = (D + \omega B_U - \omega A)(x^k + \omega(D + \omega B_L)^{-1}r^k) + \omega(r^k + Ax^k)$$

$$(D + \omega B_U)x^{k+1} = (D + \omega B_U)x^k + \omega((D + \omega B_U - \omega A)(D + \omega B_L)^{-1} + I)r^k$$

$$(D + \omega B_U)(x^{k+1} - x^k) = \omega(2 - \omega)D(D + \omega B_L)^{-1}r^k$$

$$\frac{1}{\omega(2 - \omega)}(D + \omega B_L)D^{-1}(D + \omega B_U)(x^{k+1} - x^k) = r^k$$

Assim a matriz de pré-condicionamento P para $0 < \omega < 2$, pela equação (5.7) é,

$$P = \frac{1}{\omega(2 - \omega)}(D + \omega B_L)D^{-1}(D + \omega B_U) \quad (5.16)$$

ou

$$P = ((D + \omega B_L)N^{-1})((D + \omega B_L)N^{-1})^T \quad (5.17)$$

onde

$$N = (\omega(2 - \omega)D)^{1/2} \quad (5.18)$$

P é simétrica e o sistema linear *explicitamente pré-condicionado pelo SSOR* leva a,

$$(N(D + \omega B_L)^{-1}A(D + \omega B_L)^{-T}N)(N^{-1}(D + \omega B_L)^T x) = N(D + \omega B_L)^{-1}b \quad (5.19)$$

ou pelo *pré-condicionamento implícito* tem-se,

$$A [((D + \omega B_L)N^{-1})((D + \omega B_L)N^{-1})^T]^{-1} x = [((D + \omega B_L)N^{-1})((D + \omega B_L)N^{-1})^T]^{-1} b \quad (5.20)$$

Segundo Papadrakakis, o fator escalar $1/[\omega(2-\omega)]$ tem sido abandonado desde que ele não tem influência no número de condição de $P^{-1}A$, onde ω é um parâmetro de relaxação tomado na vizinhança de um (Papadrakakis, 1993).

Recentemente Papadrakakis e Dracopoulos propuseram uma modificação para o pré-condicionamento SSOR, que é conhecido como pré-condicionamento parcial. Esta técnica combina simplicidade na formação da matriz de pré-condicionamento P evitando complicados endereçamentos, que são requeridos no caso da fatoração de Cholesky Incompleta, com alta demanda de armazenagem (Papadrakakis and Dracopoulos, 1991). Consequentemente P é obtido como,

$$P = P_E \cdot P_D^T \quad (5.21)$$

onde

$$P_E = (D + \omega B)D^{-1/2} \quad (5.22)$$

com D formada pelos elementos diagonais da matriz de coeficientes A , B pelos elementos restantes, fora da diagonal da matriz de coeficientes A , e ω o parâmetro de relaxação. O critério que especifica os elementos de A que devem permanecer em B é definido por um parâmetro ψ controlado pelo usuário. Logo todos elementos de A satisfazendo

$$A_{ij}^2 < \psi \cdot AP_i \cdot AP_j \quad (5.23)$$

são considerados pequenos diminuindo o efeito do número de condição de A e não são incluídos em B . O caso de $\psi = 1$ corresponde ao frequentemente usado e facilmente aplicado escalonamento diagonal, e o caso de $\psi = 0$ corresponde ao método iterativo SSOR. Importante notar, que apesar de certos elementos da matriz A serem rejeitados a matriz P é sempre positiva definida, portanto a convergência do método iterativo é garantida (Papadrakakis and Dracopoulos, 1991; Papadrakakis, 1993).

5.5 Pré-condicionamento por fatoração de Cholesky incompleta

As fatorações incompletas são derivadas da solução direta de $Ax = b$, onde a matriz A simétrica positiva definida têm a fatoração completa de Cholesky representada por $A = LL^T$ onde L é a matriz triangular inferior determinada recursivamente. Assim, os produtos $L^{-1}z$ e $(L^T)^{-1}z$, onde z é um vetor qualquer, são facilmente calculados. Uma simples modificação na fatoração

LL^T em que as raízes quadradas são evitadas é apresentado como segue, $A = LDL^T$, onde L e D são determinadas recursivamente. Infelizmente, para matrizes esparsas de grandes dimensões, o cálculo de L requer tempo e armazenagem extra, e muitos elementos da matriz de coeficientes A que eram nulos tornam-se não-nulos e assim a esparsidade da matriz original é perdida. Para evitar este problema faz-se uma fatoração incompleta da matriz A , que assume a seguinte forma,

$$A = \mathbf{L}\mathbf{L}^T - R. \quad (5.24)$$

onde R é uma matriz resíduo formada pelos elementos descartados na fatoração de Cholesky e \mathbf{L} é a matriz triangular inferior modificada, ou seja, com alguns elementos ignorados na fatoração. Se $L_{ii} = 0$ o algoritmo para, e se $L_{ii} < 0$ o método GC pode não encontrar a solução esperada. Meijerink e Van der Vorst mostraram que se A é uma M-matriz ($A_{ij} \leq 0$ se $i \neq j$) a fatoração de Cholesky incompleta sempre encontra $L_{ii} > 0$, contudo se a matriz é simétrica positiva e definida arbitrária nem sempre encontra-se $L_{ii} > 0$. O trabalho e armazenagem envolvidos no cálculo da aproximada \mathbf{L} é muito menor que o envolvido no cálculo da matriz triangular exata (Kershaw, 1978). Quando a fatoração incompleta é obtida o sistema de equações a ser resolvido pelo método GC tem a forma:

$$\mathbf{L}^{-1}\mathbf{A}\mathbf{L}^{-T}y = \bar{b} \quad (5.25)$$

onde $y = \mathbf{L}^T x$ é o vetor variável transformado e $\bar{b} = \mathbf{L}^{-1}b$ é o vetor independente transformado. O método do GC com pré-condicionamento por fatoração de Cholesky incompleta (GCPCI) pode ser descrito de acordo com o pseudo-código da figura 5.4.

valores iniciais $\{x^0, A, b, \xi, k_{\max}\}$ $y^0 = 0; \quad p^0 = r^0 = \bar{b};$ repita enquanto { $z^k = \mathbf{L}^{-1}\mathbf{A}\mathbf{L}^{-T}p^k;$ $\rho^k = (r^k, r^k);$ $\alpha^k = \rho^k / (p^k, z^k);$ $y^{k+1} = y^k + \alpha^k p^k;$ $r^{k+1} = r^k - \alpha^k z^k;$ $\beta^k = (r^{k+1}, r^{k+1}) / \rho^k;$ $p^{k+1} = r^{k+1} + \beta^k p^k;$ } $\ r^k\ _2 / \ r^0\ _2 < \xi$ ou $\ r^k\ _2 / \ \bar{b}\ _2$ ou $k = k_{\max}$
--

Figura 5.4 - Algoritmo GCPCI.

Na figura 5.4, p^k é o vetor direção conjugado; r^k é o vetor resíduo; y^k é o vetor solução; u^k é o produto da matriz transformada e o vetor direção conjugado.

O passo de pré-multiplicação, sem a formação da matriz transformada explicitamente, é obtido pelas três operações seguintes (Ajiz and Jennings, 1984):

Substituição “backward”: $\mathbf{L}^T v^k = p^k$

Pré-multiplicação: $w^k = Av^k$

Substituição “forward”: $\mathbf{L}z^k = w^k$

O sucesso do GCPCI pode depender de quão boa é a aproximada inversa $(\mathbf{L}\mathbf{L}^T)^{-1}$, isto é, do quanto se aproxima da inversa da matriz A de coeficientes. De acordo com Kershaw o GCPCI e suas generalizações devem ser adequados para matrizes que surgem de EDPs, contudo para matrizes arbitrárias o fato de vizinhos distantes serem descartados durante a fatoração faz com que o método torne-se, provavelmente, inadequado (Kershaw, 1978).

O refinamento iterativo pode ser usado para resolver a equação (1.1) de acordo com a fórmula,

$$\mathbf{L}\mathbf{L}^T x^{k+1} = b + R x^k \quad (5.26)$$

O sistema linear $Ax = b$ pré-condicionado pela fatoração de Cholesky incompleta pode ser expresso, de acordo com a equação (5.10) tal como,

$$P x^{k+1} = P x^k + \psi_k r^k \quad (5.27)$$

onde $P = A + R$, $r^k = b - Ax^k$ e ψ é um parâmetro de tolerância. A matriz de pré-condicionamento $P = \mathbf{L}\mathbf{L}^T$, cuja matriz triangular inferior têm elementos não-nulos só nas posições correspondendo aos elementos não-nulos de A , foi proposta por Stone em 1968 (Campos and Rollett, 1995).

Mais tarde, surgiu a classe de pré-condicionamento chamada de decomposição incompleta \mathbf{LU} da matriz A , proposta em 1977 por Meijerink e Van der Vorst (Axelsson, 1985; Simon, 1988; Paolini and Brozolo, 1989), em tais matrizes os elementos não-nulos eram determinados a priori através da adição de diagonais de “fill-ins”. Sob os elementos não-nulos da matriz A são efetuadas operações correspondendo a uma fatoração *falsa* \mathbf{LU} .

Como a matriz A é esparsa deve ser feita uma importante consideração quanto ao custo do processo de fatoração incompleta, realizada a medida em que o número de novos elementos não-

nulos, gerados pela fatoração, são controlados por um ou mais parâmetros. Os parâmetros são usualmente uma constante inteira, por exemplo, indicando quantas diagonais novas podem ser formadas. Quando todos os elementos da matriz \mathbf{L} coincidem com a estrutura original da matriz \mathbf{A} têm-se o nível zero, ou pode-se dizer, que o pré-condicionador não têm nenhuma diagonal extra.

O conceito de diagonais extras é usado só dentro do contexto de matrizes naturalmente ordenadas que surgem, geralmente, da discretização através do MDF. As matrizes com esparsidade variadas, têm o conceito de diagonais extras generalizado para o conceito de níveis de “fill-in”. Este método de fatoração simbólica, requer um passo para determinar a posição de elementos não-nulos e a quantidade de espaço e memória requeridos para armazená-los, será analisado a seguir.

5.5.1 Pré-condicionamento com “fill-ins”

Na equação (5.24) se $R = 0$ então $P = LL^T = A$ e a equação (5.26) torna-se $LL^T x = b$, portanto a solução do sistema é encontrada pelo método direto Cholesky. Se $R \approx 0$ então $P = \mathbf{LL}^T \approx A$. A escolha de P entre a matriz de coeficientes A e a matriz identidade I , resulta em diferentes métodos iterativos pré-condicionados. O fator \mathbf{L} pode, em geral, ser menos esparsa que a matriz A , devido aos “fill-ins” durante a fase de fatoração.

Uma alternativa para fixar o número de “fill-ins” é o parâmetro de tolerância. Nesta estratégia, elementos não-nulos são incluídos no fator incompleto se eles são maiores que este parâmetro de tolerância. Desta forma nem o padrão nem o número de elementos não-nulos no fator incompleto podem ser determinados a priori e sim durante a fatoração (Jones and Plassmann, 1995). Os diversos modos de escolha e contagem destes termos descartados configura diferentes fatorações incompletas,

- (i) nos métodos de fatoração de Cholesky incompleta clássica força-se os fatores a serem esparsos, pela permissão de elementos não-nulos só em um conjunto específico de lugares, podendo ser denominada como fatoração incompleta pela posição;
- (ii) outra forma de realizar uma fatoração incompleta é pelo valor, ou seja, os elementos não-nulos só são aceitos se seus valores absolutos são *grande o bastante*, onde elementos muito pequenos podem tornar-se desprezíveis e serem descartados;

Quanto mais alto o nível maior será o “fill-in”. Marcondes, em 1996, empregando malhas de Voronoi para a solução das equações oriundas do escoamento bifásico (óleo-água) em meios porosos, observou que usualmente níveis 0, 1 e 2 são os mais indicados, pois não comprometem o custo da iteração individual do algoritmo, níveis maiores podem destruir a esparsidade da matriz A (Marcondes, 1996).

Ultimamente, a meta é reduzir o número de condição da matriz $L^{-1}AL$. Baseado nesta idéia, Jones e Plassmann desenvolveram uma fatoração que segue um número fixo de elementos não-nulos em cada linha ou coluna de L , deste modo a armazenagem dos elementos da matriz L é fixada e conhecida antes da fatoração. O fato de manter só os maiores elementos permite com que se possa ter os mesmos benefícios do esquema de um parâmetro de tolerância, sem usá-lo (Jones and Plassmann, 1995).

5.6 Pré-condicionamento por série truncada de Maclaurin

A matriz A do sistema (1.1) pode ser decomposta em:

$$A = D + B \quad (5.28)$$

onde D é uma matriz diagonal que contém os elementos diagonais de A e B é uma matriz formada pelos elementos não diagonais de A (Azevedo, 1996). Substituindo a equação (5.29) na equação (1.1) e isolando a variável x obtém-se a seguinte expressão:

$$x = (I + D^{-1}B)^{-1} \cdot D^{-1} \cdot b \quad (5.29)$$

onde I é uma matriz identidade de dimensão igual a da matriz A .

Na equação (5.29) o termo $(I + D^{-1}B)^{-1} \cdot D^{-1}$ representa a inversa da matriz A , substituindo a equação (5.30) na mesma equação tem-se:

$$G = D^{-1} B \quad (5.30)$$

$$x = (I + G)^{-1} \cdot D^{-1} \cdot b \quad (5.31)$$

A Série de Maclaurin é descrita por:

$$(I + G)^{-1} = (I - G + G^2 - G^3 + \dots - G^{2k-1} + G^{2k} - \dots), \text{ para } \forall k. \quad (5.32)$$

A inversa da matriz A será:

$$A^{-1} = (I - G + G^2 - G^3 + \dots - G^{2k-1} + G^{2k} - \dots)D^{-1} \quad (5.33)$$

A matriz de pré-condicionamento P ficará mais próxima de A dependendo do número de termos utilizados na Série de Maclaurin. O sistema auxiliar $P.z = r$, utilizado no método pré-condicionado ficará com a seguinte forma:

$$z = (I - G + G^2 - G^3 + \dots - G^{2k-1} + G^{2k} - \dots)D^{-1} r \quad (5.34)$$

Substituindo-se $r_{aux} = D^{-1}.r$ na equação (5.34) obtém-se a seguinte expressão:

$$z = (r_{aux} - (G.r_{aux}) + G.(G.r_{aux}) - G(G.(G.r_{aux})) + \dots) \quad (5.35)$$

O número de operações matriz-vetor utilizados no pré-condicionamento é dependente do número de termos em G utilizados na Série de Maclaurin truncada. Para que a Série de Maclaurin venha a convergir é necessário que o raio espectral de R

$$R = -D^{-1}B \quad (5.36)$$

seja menor que a unidade, isto é $\rho = \text{maior } |\lambda_i| < 1$ para $i=\{1, 2, \dots, n\}$, sendo n a dimensão do sistema e ρ o raio espectral definido pelo módulo do autovalor de maior módulo (Azevedo, 1996).

5.7 Pré-condicionamento por série truncada de Neumann

Decompondo-se a matriz A da equação (1.1) em $A = G - E$, G não-singular, logo a inversa de A será dada por:

$$A^{-1} = \left[I + G^{-1}E + (G^{-1}E)^2 + (G^{-1}E)^3 + \dots \right] \cdot G^{-1} \quad (5.37)$$

onde o módulo do autovalor de maior módulo de $G^{-1}E$ deve ser menor que um. Definindo-se $S = G^{-1}E$ tem-se:

$$A^{-1} = \left[\sum_{k=0}^{\infty} S^k \right] \cdot G^{-1} \quad (5.38)$$

O sistema auxiliar $P.z = r \Rightarrow z = P^{-1} \cdot r$ para este caso ficará:

$$z = \left[I + S + S^2 + S^3 + \dots \right] \cdot G^{-1}r \quad (5.39)$$

onde o grau de proximidade que a matriz de pré-condicionamento P terá com relação a matriz de coeficientes A , depende do número de termos utilizados na Série de Neumann.

5.8 Pré-condicionamento polinomial

O pré-condicionamento polinomial consiste em escolher um polinômio $s(A)$ que assemelhe-se a inversa da matriz de coeficientes, apresentado na equação (5.40).

$$s(A) = A^{-1} = [I - V]^{-1} = \eta_0 I + \eta_1 V + \eta_2 V^2 + \dots + \eta_k V^k \quad (5.40)$$

A mais simples expansão polinomial da equação (5.40) é dada pela série de Neumann, com $\{\eta_0, \eta_1, \dots, \eta_k\} = 1$, conforme equação (5.41):

$$s(A) = I + V + V^2 + \dots \quad (5.41)$$

Eficientes polinômios podem ser encontrados, tornando o espectro da matriz pré-condicionada mais próximo, quanto possível, da matriz identidade. Uma maneira de se conseguir isto é minimizando a norma $\|I - s(A)\|$, onde $\|\cdot\|$ representa a norma L_2 . Alguns pré-condicionamentos polinomiais são encontrados em Johnson *et al.* (1983) e Papadrakakis (1993)

5.9 Conclusão

Uma das dificuldades encontradas na resolução de equações simétricas, utilizando os métodos iterativos clássicos, é a velocidade de convergência lenta. A convergência está ligada ao número de condição da matriz de coeficientes, quando o número de condição é grande a tendência é aumentar o tempo de processamento. A condição pode ser melhorada com as técnicas de pré-condicionamento. Neste capítulo apresentou-se alguns pré-condicionamentos encontrados na literatura, sobre estes pode-se destacar:

- (i) o pré-condicionamento diagonal é simples de implementar, adequado ao ambiente seqüencial e paralelo (Barrett *et al.*, 1994). Na resolução de alguns sistemas lineares simétricos obteve-se resultados bastante promissores utilizando-se este pré-condicionamento.
- (ii) o pré-condicionamento SSOR necessita de um fator de relaxação (ω) (Campos, 1995) e para o SSOR modificado é necessário outro fator ψ , que determina os elementos da matriz de coeficientes que devem permanecer na matriz B , usada no cálculo do pré-condicionamento SSOR (Papadrakakis and Dracopoulos, 1991). Neste pré-condicionamento ambos os parâmetros são controlados pelo usuário, cabendo a ele encontrar os parâmetros mais adequados para resolver seu problema.
- (iii) o pré-condicionamento por fatoração LL^T é encontrado em diversos trabalhos (Ajiz and Jennings, 1984; Campos and Rollett, 1995; Jonnes and Plassman, 1995) com muitas formas para descartar elementos que se formam durante a fatoração. Neste trabalho será implementada a fatoração LL^T sem “fill-in”, isto é, após a fatoração só permanecem na matriz de pré-condicionamento os elementos que estão nas mesmas posições dos elementos da matriz de coeficientes.
- (iv) no pré-condicionamento por série truncada de Maclaurin não é necessário realizar o produto $D^{-1}B$ explicitamente e assim pode-se trabalhar com a matriz esparsa, contudo este pré-condicionamento torna-se mais adequado se usado com processamento paralelo.
- (v) o pré-condicionamento por série truncada de Neumann é semelhante ao de Maclaurin e não será implementado neste trabalho. No pré-condicionamento polinomial é necessário encontrar um parâmetro η_i de forma a otimizar o polinômio matricial, o parâmetro é escolhido de acordo com o problema a ser resolvido. Embora, algumas escolhas bastante simples para o parâmetro sejam sugeridas na literatura (Papadrakakis, 1993), este pré-condicionamento não é utilizado no presente trabalho.

Cabe assinalar que nas últimas décadas surgiram alguns trabalhos aplicando as técnicas de pré-condicionamentos aos sistemas assimétricos, pode-se citar os trabalhos de Brussino e Sonnad (1989), Venkatakrishnan (1990), Mchugh and Knoll, (1994). Neste trabalho serão utilizados os pré-condicionamentos diagonal, SSOR e por fatoração de CI no método GC, o que pode ser verificado no capítulo 7.

Capítulo 6 - Ordenação

6.1 Introdução

Neste capítulo serão descritas algumas ordenações das variáveis e proposta uma ordenação em linha contínua. A ordenação proposta utiliza as características da malha geométrica (Cardoso *et al.*, 1996).

Os métodos GC, “Tridiagonal Matrix Algorithm” e Gauss-Seidel são apresentados e aplicados na solução do sistema de equações algébricas lineares. A performance é avaliada em termos de tempo de processamento e número de iterações necessárias para o alcance da precisão estipulada para a convergência. Estes métodos são bastante conhecidos e frequentemente empregados na área de Mecânica dos Fluidos Computacional (Tanyi and Thatcher, 1996).

A obtenção dos resultados numéricos, mostrados no presente capítulo, tem como objetivo principal averiguar a eficácia dos métodos com a ordenação proposta e verificar a que custo computacional produz-se uma solução satisfatória. Porém, os resultados são preliminares e testes mais rigorosos e completos se fazem necessários, onde um refino da malha seria um aspecto importante a ser abordado, assim como outras ordenações encontradas na literatura (Saint-Georges *et al.*, 1996).

6.2 Definições preliminares

Pode-se observar que a estrutura da matriz de coeficientes está intimamente associada à malha geométrica, ou seja, a maneira pela qual os diversos pontos nodais estão interligados independe, completamente, dos valores numéricos dos parâmetros dessas interconexões.

Os segmentos de linha, orientados ou não-orientados, são denominados *ramos* ou *ligações* e seus terminais são designados *nós*. Ao conjunto de nós e elementos que descrevem a estrutura topológica da malha estruturada ou não-estruturada dá-se o nome de *grafo*.

Um *subgrafo* é qualquer subconjunto de elementos de um grafo. Um *caminho* é um subgrafo de elementos conexos com não mais de dois elementos conectados a qualquer um dos seus nós. A *distância* $d(i, k)$ entre dois nós i e k é o comprimento do caminho mais curto entre i e k . Por definição $d(i, i) = 0$. A *vizinhança* de um nó i , $V(i)$, é o conjunto de todos os nós k tais que $d(i, k) > 0$ e exista a linha ligando i a k . Se for atribuído um sentido no grafo conexo este é

chamado *grafo orientado*. Logo o grafo da malha não-estruturada da figura 2.8 pode ser representado conforme indicado na figura 6.1.

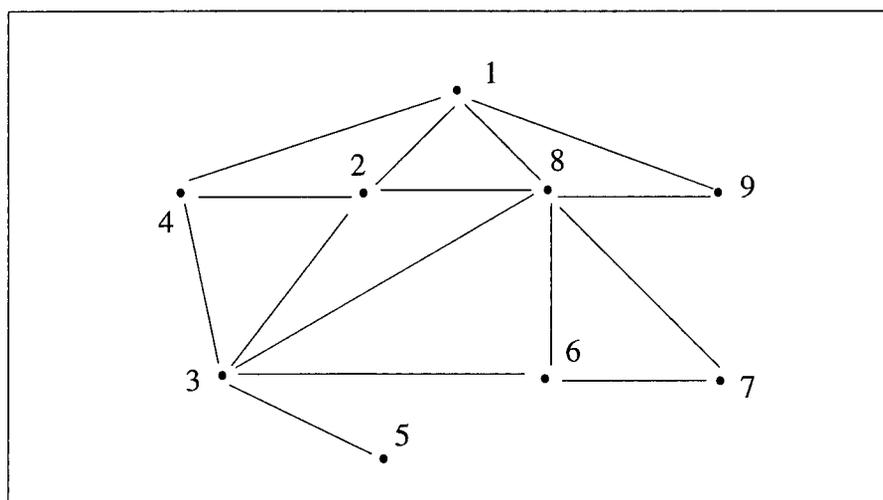


Figura 6.1 - Grafo representativo da malha com nove pontos nodais.

Matrizes de incidência simétricas podem ter sua estrutura corretamente descritas por grafos não-orientados, pois, nestes casos, como $a_{ij} \neq 0$, sempre que houver um segmento orientado de $i \rightarrow j$, necessariamente há outro segmento orientado de $j \leftarrow i$, fato que pode ser perfeitamente descrito por um segmento não orientado $i \leftrightarrow j$ (Christofides, 1975; Rich, 1988).

Os métodos de solução de sistemas de equações algébricas lineares, cujas matrizes são esparsas, relacionam-se com a disposição dos elementos não nulos dentro da matriz, e esta estrutura é convenientemente descrita e manipulada em termos da teoria de grafos. Fazendo uso de algumas definições desta teoria pretende-se descrever dois tipos de ordenações já existentes na literatura e propor a ordenação em linha contínua. Para atingir este objetivo foi realizado um estudo das diversas formas de ligação entre os pontos nodais, com o objetivo de agrupá-los ao máximo, promovendo uma minimização da esparsidade dos índices dos pontos formadores da matriz de coeficientes. Tal estudo exige uma análise gráfica, objetivando a otimização das ligações nodais.

6.3 Técnicas de Ordenação

A ordenação das variáveis pode ter um efeito significativo na convergência dos métodos do Gradiente Conjugado pré-condicionado, existem alguns estudos sobre o uso de várias técnicas de

ordenação acopladas com os pré-condicionadores por fatoração incompleta LU (ou ILU) (D’Azevedo *et al.*, 1992), assim como existem consideráveis trabalhos sobre o efeito da ordenação em problemas envolvendo malhas não-estruturadas (Maliska Jr. e Bezerra, 1994; Marcondes, 1996).

A maior parte das técnicas de ordenação estão baseadas na estrutura da matriz, e não usam os valores numéricos de seus elementos. Em geral as diversas técnicas podem ser classificadas conforme D’Azevedo *et al.* tais como:

- (i) ordenações randômicas normalmente não são adequadas;
- (ii) ordenações em linha natural, isto é, de acordo com a geração da malha geométrica, os métodos de resolução tem bom desempenho;
- (iii) ordenações que reduzem os “fill-ins”, tal como “minimum degree” e “nested dissection” não têm uma performance tão satisfatória na resolução do sistema;
- (iv) as ordenações que reduzem o sistema são muito efetivas, tal como:
 - a) ordenação “red/black” de um grafo bipartite; e
 - b) eliminação exata dos nós vermelhos.
- (v) ordenações que reduzem os “fill-ins” descartados são bastante adequadas, tal como “reverse Cuthill-McKee” (Liu and Sherman, 1976).

6.3.1 Ordenação geométrica

A ordenação natural é a ordenação oriunda do gerador da malha não-estruturada. As malhas geradas pelo DV podem ter volumes de controle hexagonais, retangulares, triangulares ou aleatórios, com os volumes podendo ser distorcidos ou irregulares. Após a geração da malha não-estruturada os pontos e suas ligações com a *vizinhança* definem a matriz esparsa de coeficientes. Vale ressaltar que o termo ordenação natural não é o mesmo utilizado em malhas cartesianas, onde a resolução dos sistemas é efetuada em linhas horizontais ou verticais (Marcondes, 1996).

A ordenação vertical ou horizontal, proposta por Palagi em 1992, é realizada de forma semelhante a ordenação natural empregada em malhas cartesianas. Para o caso da ordenação vertical, por exemplo na malha da figura 6.2, localiza-se o ponto gerador com menor valor de x e passa-se uma linha vertical com largura igual a x mínimo. Todos os pontos geradores que estiverem sobre esta linha serão renumerados, o próximo passo é localizar, entre os pontos não renumerados, o próximo ponto com menor valor de x e repete-se o processo para os pontos restantes. A ordenação horizontal é realizada de forma semelhante a ordenação vertical, a diferença está na forma com que as linhas são traçadas, que são no sentido horizontal, onde usa-se a cota igual a y mínimo. Na figura 6.2 apresenta-se também a matriz de coeficientes obtida com a ordenação.

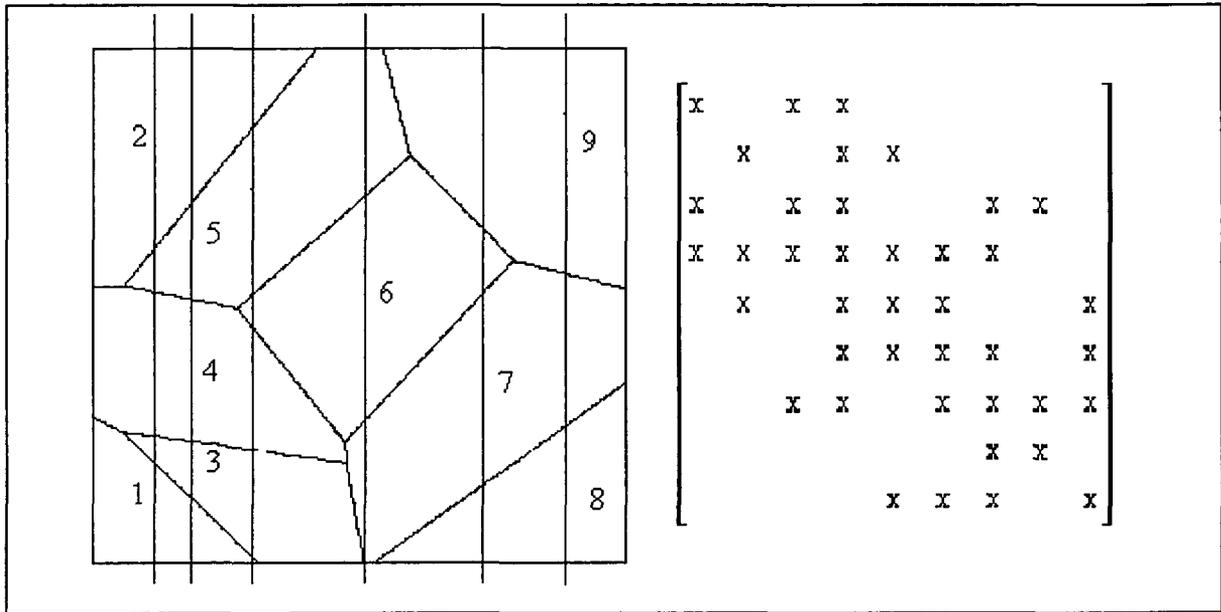


Figura 6.2 - Malha aleatória de Voronoi e matriz de coeficientes - ordenação vertical.

Observando a estrutura da matriz resultante da ordenação, é evidente a concentração dos elementos não-nulos em torno da diagonal principal. A concentração torna-se mais explícita com maior número de pontos na malha computacional.

6.3.2 Ordenação em linha contínua

Para malhas não-estruturadas não se tem uma linha definida, porém, é possível trabalhar com zigue-zagues curtos (Maliska Jr e Bezerra, 1994). Propõe-se, no presente trabalho, um zigue-zague contínuo em malha bidimensional, que é encontrado em Cardoso *et al.* (1996), ou seja um caminho passando por todos os pontos e seguindo alguns critérios, apresentados conforme segue:

- (1) Tomar o ponto inicial i , como sendo o mais próximo da origem (0,0), pode-se escolher outro ponto diferente da origem, mas dá-se preferência a pontos que estejam próximos a um dos contornos da geometria;
- (2) verificar quais são os vizinhos do ponto i , escolhido como ponto inicial, por exemplo: j_1, j_2, \dots, j_n ;
- (3) verificar o número de vizinhos de cada um dos vizinhos do ponto i , por exemplo:
 - j_1 tem j_{11}, j_{12} e j_{13}
 - j_2 tem j_{21}, j_{22}, j_{23} e j_{24}
 - j_3 tem j_{31}, j_{32} , onde j_{il} são os vizinhos de cada um dos vizinhos de i ;

- (4) escolher o vizinho de i , que tiver o menor número de vizinhos, como sendo o próximo ponto no caminho, por exemplo: j_3 será o próximo ponto do caminho pois ele tem o menor número de vizinhos
- (5) caso os vizinhos de um ponto i , tenham o mesmo número de vizinhos será usado como critério de desempate a menor distância destes pontos até a origem. Os vizinhos que já fazem parte do caminho não são mais testados e nem contados como vizinhos;
- (6) Após acrescentar o novo ponto ao caminho, este passa a ser o ponto inicial. Caso todos os pontos não tenham sido visitados voltar ao passo (2).

A varredura pode ser observada na figura 6.3 para uma malha em formato L, com 100 pontos nodais.

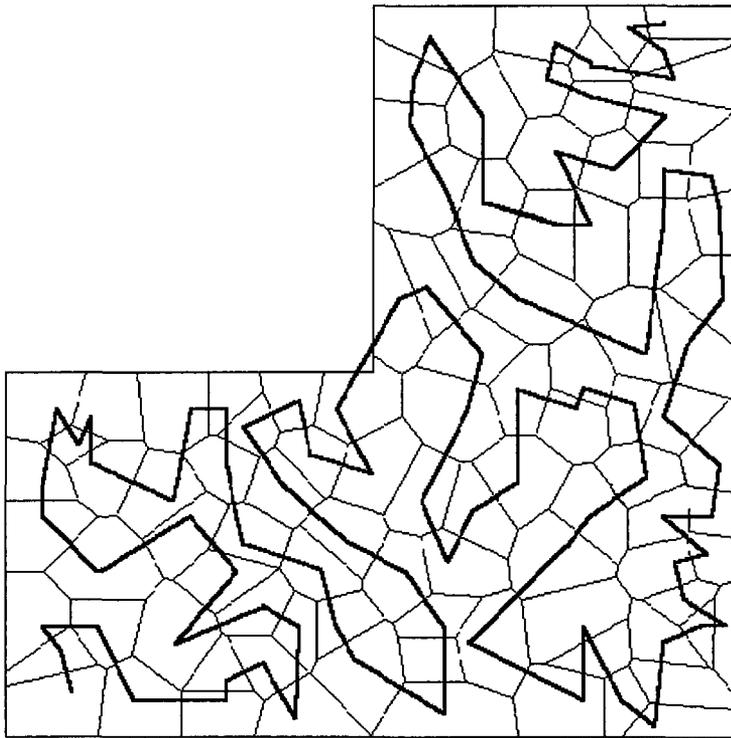


Figura 6.3 - Varredura em linha contínua, para a malha em formato L.

Trabalhos com zigue-zagues curtos foram investigados por Maliska Jr. e Bezerra (1994), obtendo resultados promissores, contudo uma linha com zigue-zague contínuo para malhas não-estruturadas é necessária para a aplicação do método TDMA.

O mais conhecido método na área de CFD para resolver sistemas que tenham sofrido uma ordenação é o TDMA. Como o próprio nome diz, é um método linha por linha que resolve diretamente uma linha em problemas unidimensionais. Já em problemas bidimensionais e tridimensionais o TDMA é iterativo com a varredura se processando linha por linha, ou coluna

por coluna, logicamente os sistemas lineares resultantes após a ordenação geram matrizes tridiagonais, que podem ser resolvidas de forma direta.

6.4 TDMA

O método “Tridiagonal Matrix Algorithm” (TDMA) resolve diretamente um problema unidimensional, já os problemas bidimensionais e tridimensionais são iterativos com a varredura se processando linha por linha, ou coluna por coluna (Patankar, 1980). Uma varredura, por exemplo, na linha horizontal é apresentada para uma malha computacional retangular, conforme figura 6.4.

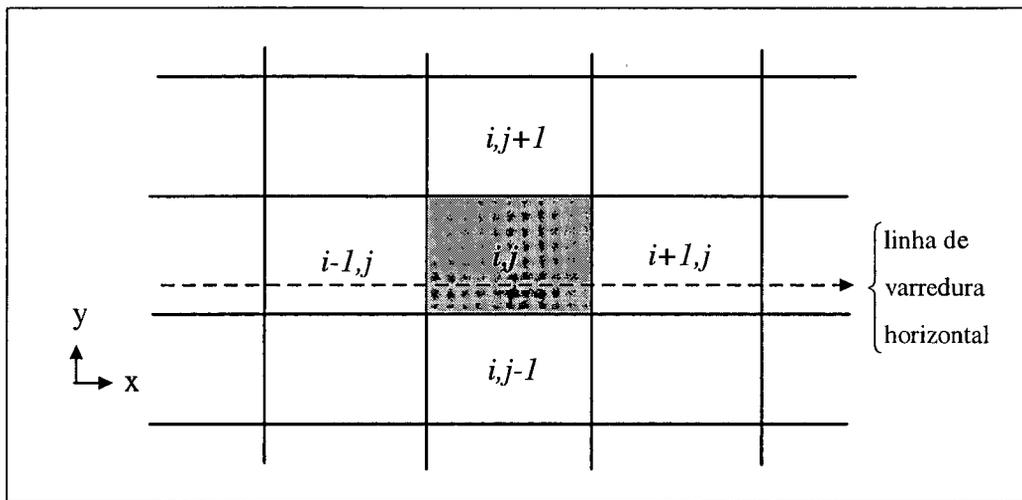


Figura 6.4 - Varredura horizontal em uma malha retangular.

Assim a equação (3.1) pode ser estruturada pela equação (6.1) conforme segue,

$$AP_i x_i - A_{i-1,j} x_{i-1,j} - A_{i+1,j} x_{i+1,j} - A_{i,j-1} x_{i,j-1} - A_{i,j+1} x_{i,j+1} = b \tag{6.1}$$

O sistema linear ao ser resolvido pelo TDMA, atribui às parcelas que estão fora da linha horizontal valores calculados em uma iteração anterior, que inicialmente seriam valores inseridos pelo usuário, desta forma o método TDMA reduz a matriz de coeficientes original do sistema linear, para uma matriz tridiagonal, tal como,

$$AP_i x_i^{k+1} - A_{i-1,j} x_{i-1,j}^{k+1} - A_{i+1,j} x_{i+1,j}^{k+1} = A_{i,j-1} x_{i,j-1}^k + A_{i,j+1} x_{i,j+1}^k + b \tag{6.2}$$

onde AP_i , $A_{i-1,j}$, $A_{i+1,j}$, $A_{i,j-1}$, $A_{i,j+1}$ são os coeficientes do ponto nodal central e pontos vizinhos que estão na mesma linha horizontal de AP_i e fora da linha horizontal, respectivamente.

Analogamente, a descrição da matriz tridiagonal, encontrada na equação (6.2), é adotada como sendo,

$$a_i x_i = c_i x_{i-1} + b_i x_{i+1} + d_i \quad (6.3)$$

onde d_i incorpora todos os termos que estejam fora da diagonal principal, isto é, da linha contínua de varredura. Para resolver o sistema linear formado por esta matriz, propõe-se que as equações sejam reduzidas a seguinte forma (matriz triangular superior).

$$x_i = P_i x_{i+1} + Q_i \quad (6.4)$$

que possibilita varrer a linha em um sentido determinando P_i e Q_i (triangularização) e voltar determinando os valores da variável x_i (retrosubstituição). Subtraindo um índice da equação 6.4 obtém-se,

$$x_{i-1} = P_{i-1} x_i + Q_{i-1} \quad (6.5)$$

Substituindo a equação 6.5 na equação 6.3 vem,

$$a_i x_i = c_i (P_{i-1} x_i + Q_{i-1}) + b_i x_{i+1} + d_i \quad (6.6)$$

Isolando a variável x_i na equação 6.6, tem-se

$$x_i = \frac{b_i}{(a_i - c_i P_{i-1})} x_{i+1} + \frac{(c_i Q_{i-1} + d_i)}{(a_i - c_i P_{i-1})} \quad (6.7)$$

Comparando a equação (6.7) com a equação (6.4), determina-se P_i e Q_i .

$$P_i = \frac{b_i}{(a_i - c_i P_{i-1})} \quad (6.8)$$

$$Q_i = \frac{(c_i Q_{i-1} + d_i)}{(a_i - c_i P_{i-1})} \quad (6.9)$$

A figura 6.5 descreve um algoritmo para o método TDMA.

Algoritmo TDMA

```

Atribuição de valores iniciais  $\{x_1, x_n, n, k_{\max}\}$ 
Calcular  $P_1$  e  $Q_1$  (baseado em uma condição de contorno)
Faça para  $k = 1, \dots, k_{\max}$ 
  Faça para  $i = 2, \dots, n$ 
    Calcular  $a_i^k, b_i^k, c_i^k, d_i^k$ 

$$P_i^k = \frac{b_i^k}{(a_i^k - c_i^k P_{i-1}^k)}$$


$$Q_i^k = \frac{(c_i^k Q_{i-1}^k + d_i^k)}{(a_i^k - c_i^k P_{i-1}^k)}$$

  Fim para  $i$ 
  Faça para  $i = n-1, n-2, \dots, 2$ 

$$x_i^k = P_i^k x_{i+1}^k + Q_i^k$$

  Fim para  $i$ 
  Se teste de convergência não é satisfeito
 $k = k+1$  (incrementar contador de iterações)
Senão fim para  $k$ 

```

Figura 6.5 - Algoritmo TDMA.

Este método tende a aumentar sua eficiência para malhas com dimensões maiores (De Bortoli, 1996). Nos DV formados por volumes aleatórios, aplica-se o método TDMA, geralmente, ao longo de uma linha em zigue-zague, unindo os pontos centrais dos volumes de controle através de segmentos de retas. Tem-se, então, um problema unidimensional ao longo de cada linha, onde o ponto i está conectado aos elementos $(i-1)$ e $(i+1)$, e as demais ligações do elemento i com os outros vizinhos são reunidas no termo fonte. O processo iterativo é necessário para a atualização das variáveis que estão no termo fonte e, que contém a influência dos vizinhos.

6.5 Gauss-Seidel

No método Gauss-Seidel, também conhecido como método dos deslocamentos sucessivos ou por passos simples, usado por Gauss e descoberto por Seidel em 1874, são usados os valores disponíveis $x_i^{(k+1)}$, conforme segue,

$$x_i^{(k+1)} = \left(\frac{b_i - \sum_{j < i} (a_{ij} x_j^{(k+1)}) - \sum_{j > i} (a_{ij} x_j^{(k)})}{a_{ii}} \right), \text{ para } i = 1, 2, \dots, n \quad (6.10)$$

Maiores detalhes do método GS são encontrados em Barrett *et al.*, 1994. Uma descrição do algoritmo é apresentada na figura 6.6.

Algoritmo Gauss-Seidel

```

Atribuir um valor inicial  $x^0$  para a solução  $x^*$ .
Repita enquanto  $k = 1, 2, \dots, k_{\max}$ 
  Faça para  $i = 1, 2, \dots, n$  (número de equações do sistema linear)
     $\bar{x}_i = 0$ 
    Faça para  $j = 1, 2, \dots, i-1$ 
       $\bar{x}_i = \bar{x}_i + a_{ij}x_j^{k+1}$ 
    Fim para
    Faça para  $j = i+1, \dots, n$ 
       $\bar{x}_i = \bar{x}_i + a_{ij}x_j^k$ 
    Fim para
     $x_i^{k+1} = \frac{(b_i - \bar{x}_i)}{a_{ii}}$ 
  Fim para
  Testar o critério de convergência e continuar se não convergiu ainda.
   $k = k+1$  (incremento do contador de iterações)
Fim repita

```

Figura 6.6 - Algoritmo Gauss-Seidel.

Se a matriz A for *diagonal dominante* e for *irredutível*, ou tiver *diagonal estritamente dominante*, para qualquer b e x^0 o método GS converge.

Com o objetivo de analisar as ordenações descritas no presente capítulo aplicam-se os métodos GS, TDMA e GC.

6.6 Análise da ordenação

Nas figura 6.7 e 6.8 são apresentadas uma malha hexagonal formada por 2475 pontos e uma retangular formada por 2401 pontos, respectivamente.

Na malha hexagonal, o sistema linear, resultante com a discretização da equação da condução de calor, é resolvido aplicando dois tipos de ordenações das variáveis. A primeira é a ordenação proposta por Cardoso *et al.* (Cardoso *et al.*, 1996) representada pelo *sistema 1* e denominada de malha hexagonal com ordenação. Na segunda ordenação a numeração dos volumes de controle é horizontal (iniciando na origem da cavidade quadrada indo até o lado

oposto, após repete-se este procedimento) e o sistema resultante é denotado por sistema b. Na malha retangular, encontrada na figura 6.8, a numeração dos volumes de controle é efetuada em linhas diagonais, a partir da origem da cavidade quadrada.

O domínio geométrico é formado por uma cavidade quadrada, com temperatura prescrita conforme tabela 3.3.

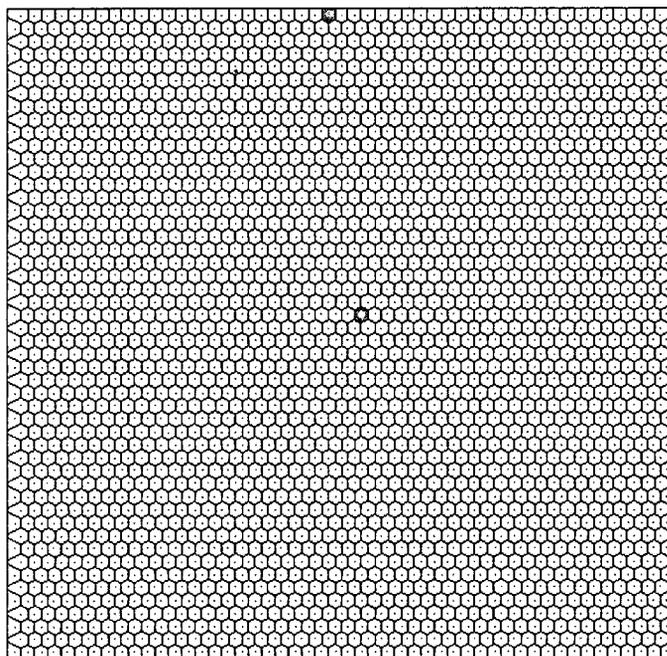


Figura 6.7 - Malha hexagonal formada por 2475 volumes.

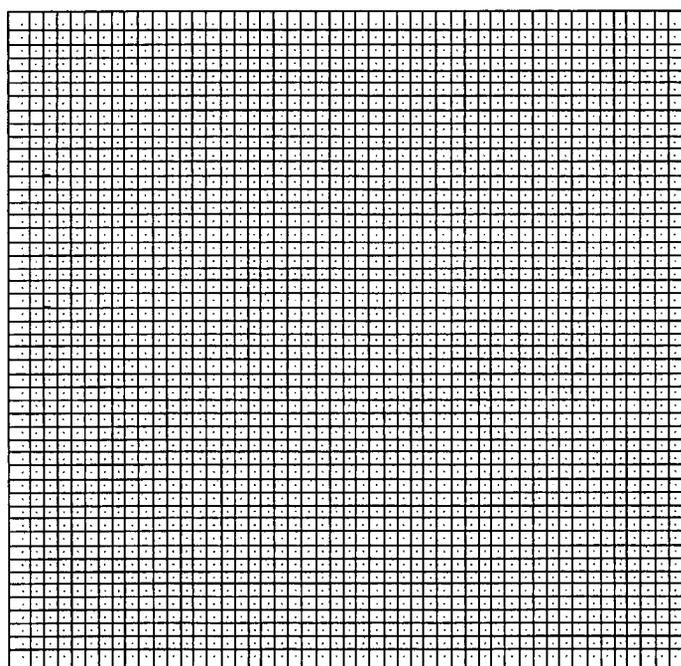


Figura 6.8 - Malha retangular formada por 2401 volumes.

A tabela 6.1 exprime as características encontradas nas malhas apresentadas anteriormente.

Tabela 6.1 - Características dos sistemas resultantes nas ordenações.

Tipo de geometria/sistema	ordem	ne	λ_{\max}	λ_{\min}	$\kappa_2 = \lambda_{\max}/\lambda_{\min}$
hexagonal com ordenação/(<i>sistema a</i>)	2475	16929	7,8925	1,0025	7,97
hexagonal original ⁵ /(<i>sistema b</i>)	2475	16929	7,8925	1,0025	7,97
retangular em diagonal/(<i>sistema c</i>)	2401	11809	8,4947	0,040	212,37

onde ne é o número de coeficientes não nulos no sistema e κ_2 é o número de condição.

Quando a condição da matriz é pequena, os pré-condicionadores poden não ser eficazes, tornando-os desnecessários.

As figuras 6.9 e 6.10 apresentam o formato da matriz de coeficientes da malha hexagonal e retangular com geração diagonal.

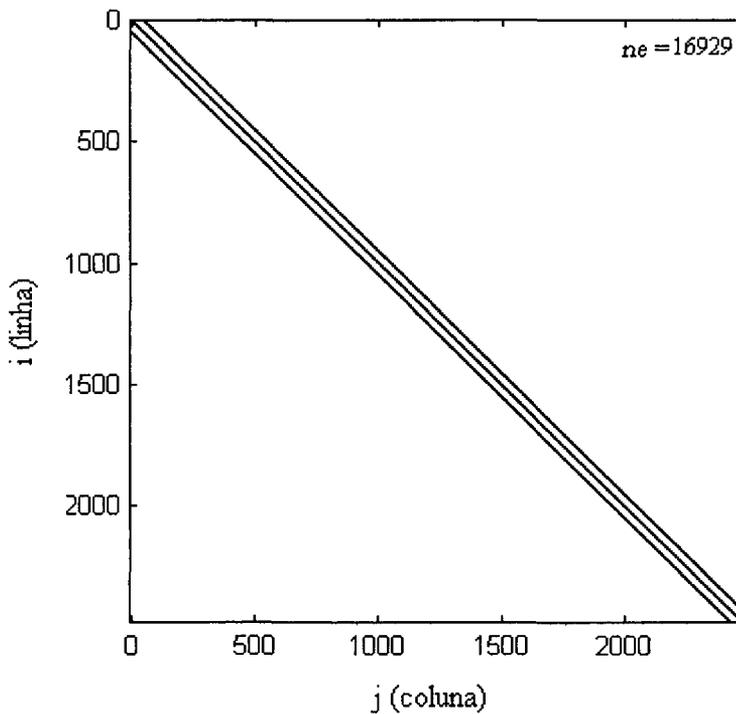


Figura 6.9 - Representação matricial (i, j) do *sistema a* e *sistema b*, ($n = 2475$).

⁵ A ordenação original na malha hexagonal é igual a ordenação horizontal, pois a geração da malha foi feita no sentido horizontal usando cota igual a y mínimo.

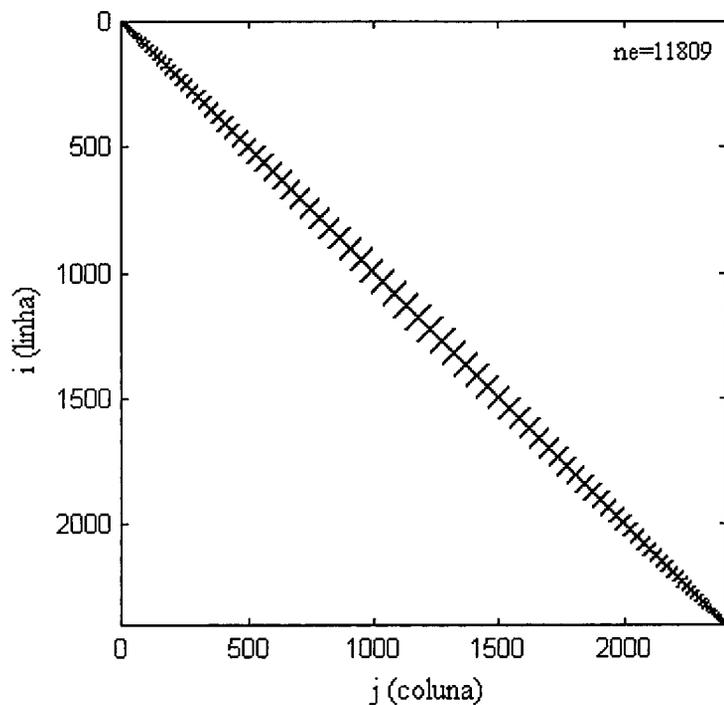


Figura 6.10 - Representação matricial (i, j) do sistema c, (n = 2401).

A figura 6.11 expõe os autovalores do sistema a e b, utilizando os coeficientes do sistema original. Na figura 6.12 são mostrados os autovalores do sistema c com a matriz original.

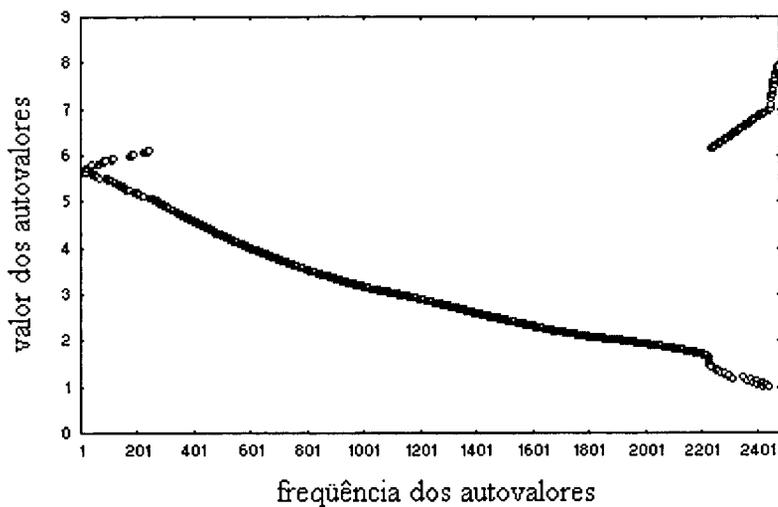


Figura 6.11 - Autovalores para o sistema a e b com a matriz de coeficientes original.

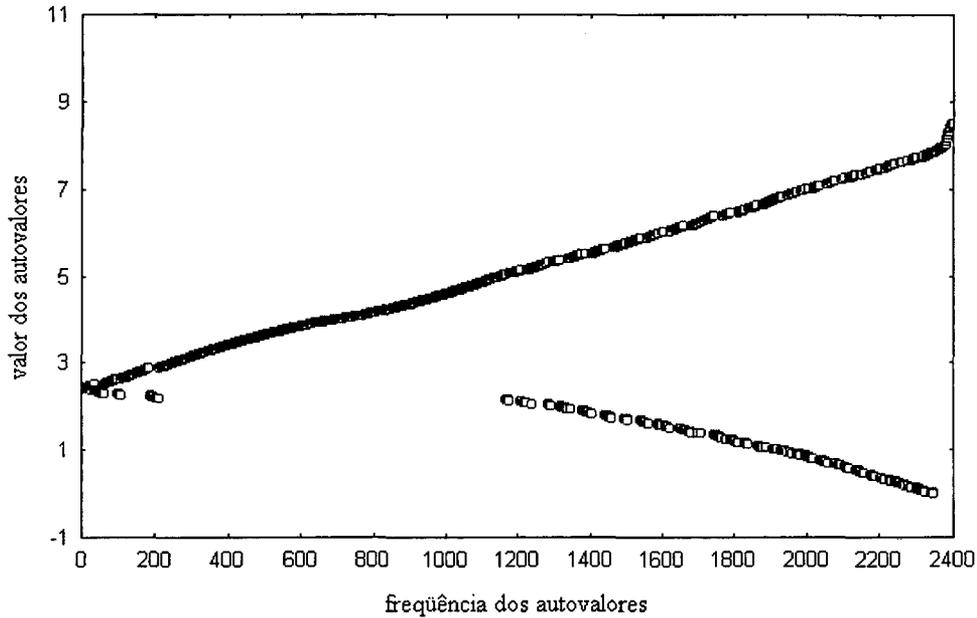


Figura 6.12 - Autovalores para o *sistema c* com a matriz de coeficientes original.

Na tabela 6.2 são exibidos os resultados encontrados, para a solução da equação da condução de calor (equação (3.10)), utilizando:

- (i) a ordenação em linha contínua (*sistema a*) e ordenação original (*sistema b*)
- (ii) a ordenação em diagonal realizada na própria geração da malha retangular (*sistema c*).

As ordenações são analisadas em termos de tempo de processamento e número de iterações, utilizando os métodos TDMA, GS, métodos tradicionais na solução de problemas de escoamento de fluidos, e o GC (Tanyi and Thatcher, 1996). A convergência é obtida utilizando o critério,

$$\text{erro}_i^{k+1} = \left\| \frac{T_i^{k+1} - T_i^k}{T_i^{k+1}} \right\| < \xi \tag{6.11}$$

onde T_i^{k+1} é a temperatura na iteração atual, T_i^k é a temperatura na iteração anterior e ξ é a precisão, aqui utilizada (10^{-5}), os resultados foram obtidos em uma SUN SPARCstation 10.

Tabela 6.2 - Resultados obtidos com o GS, TDMA e GC utilizando as ordenações.

Sistemas/métodos	Gauss-Seidel		TDMA		Gradiente Conjugado	
	iterações	tempo (s)	iterações	tempo (s)	iterações	tempo (s)
<i>sistema a</i>	801	22,70	1053	64,64	58	2,12
<i>sistema b</i>	803	17,31	830	37,09	56	1,71
<i>sistema c</i>	915	22,02	914	47,05	84	2,01

Na figura 6.13 exibem-se o comportamento dos métodos utilizados na solução do *sistema a*. O primeiro gráfico apresenta o valor para a temperatura, nas 50 primeiras iterações, em um ponto próximo a parede com temperatura prescrita de 100°C. O gráfico seguinte mostra a variação no erro (equação (6.12)). Observa-se que o GC é o método mais adequado entre os métodos usados para a resolução do *sistema a*, sendo aproximadamente 14 vezes mais rápido que o GS e 18 vezes mais rápido que o TDMA.

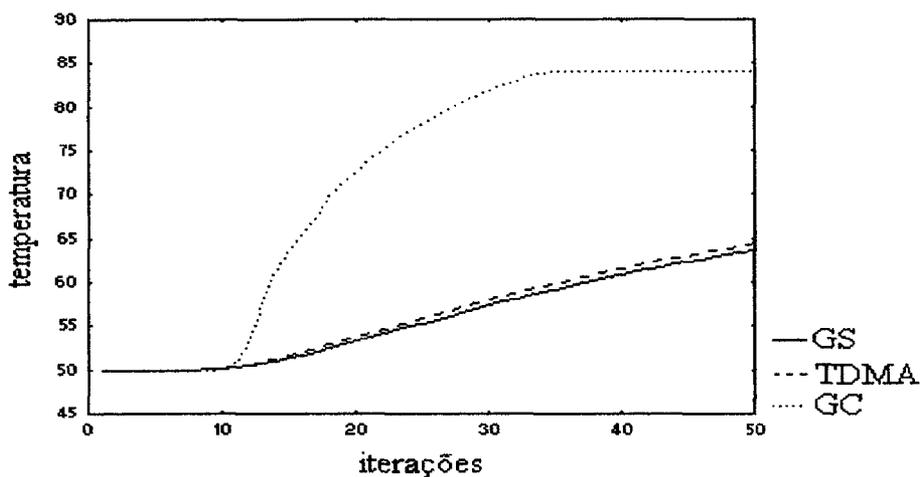


Figura 6.13 - Convergência da temperatura no *sistema a*.

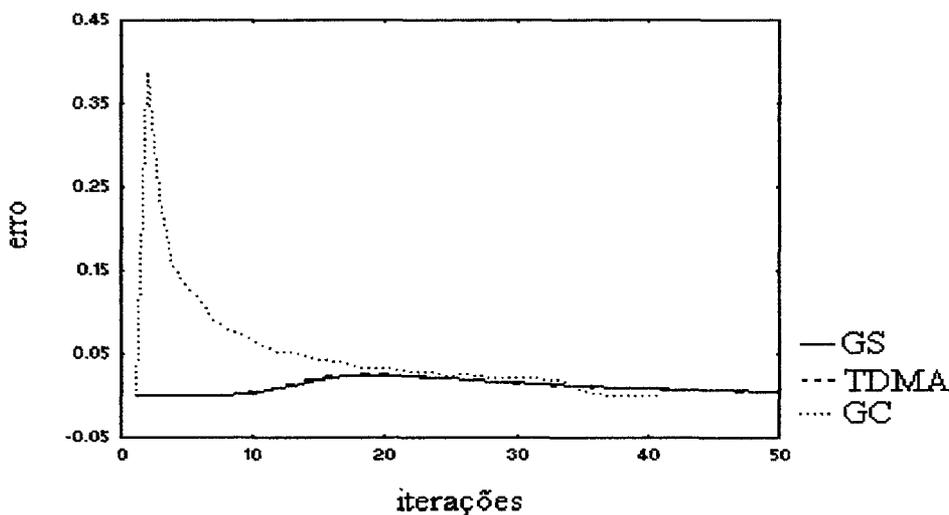


Figura 6.14 - Variação do critério de convergência no *sistema a*.

A figura 6.15 expõe os métodos utilizados no *sistema b*, o primeiro gráfico apresenta o comportamento da temperatura em um ponto próximo a parede com temperatura prescrita de 100°C. A figura 6.16, que segue, mostra a variação no erro. Verifica-se novamente que o GC é o método mais adequado, sendo aproximadamente 14 vezes mais rápido que os outros métodos, respondendo conforme as observações feitas por Tanyi and Thatcher (1996).

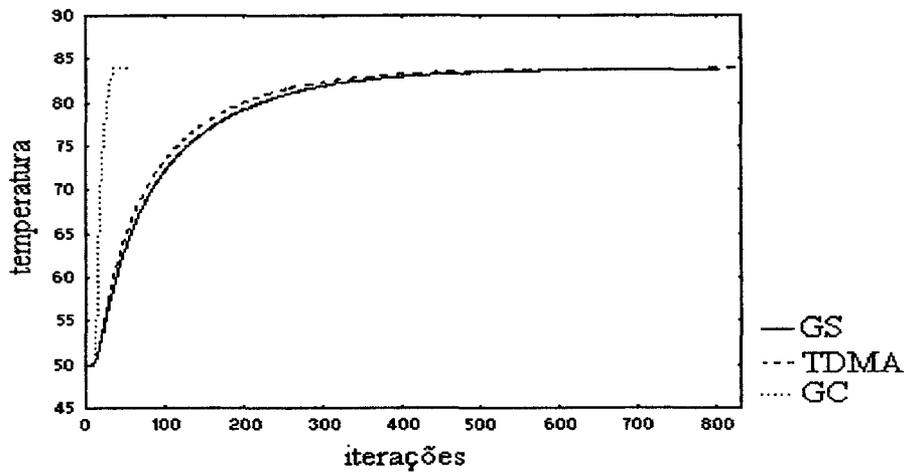


Figura 6.15 - Convergência da temperatura no *sistema b*.

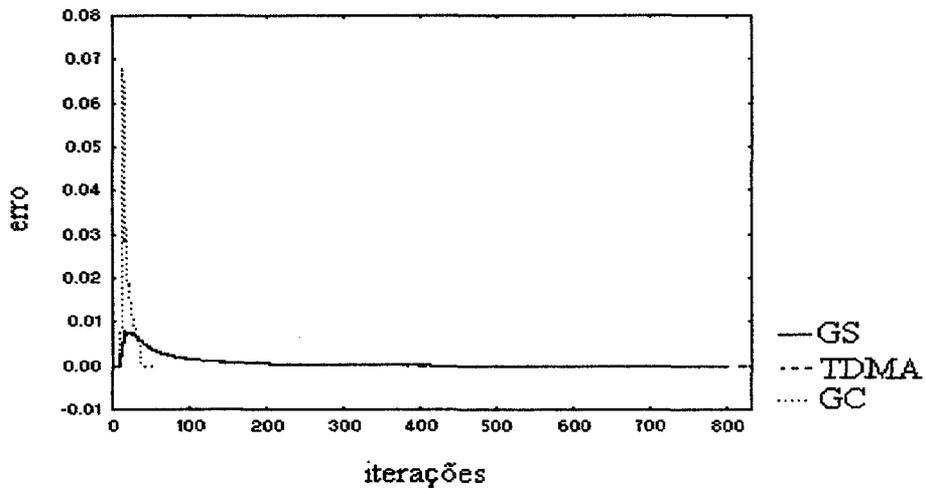


Figura 6.16 - Variação do critério de convergência no *sistema b*.

A figura 6.17 expõe o comportamento dos métodos aplicados na solução do *sistema c*, o primeiro gráfico apresenta o valor para a temperatura até a centésima iteração, em um ponto próximo a parede com temperatura prescrita de 0°C., a figura 6.18 mostra a variação no erro. O GC é aproximadamente 14 vezes mais rápido que os demais métodos.

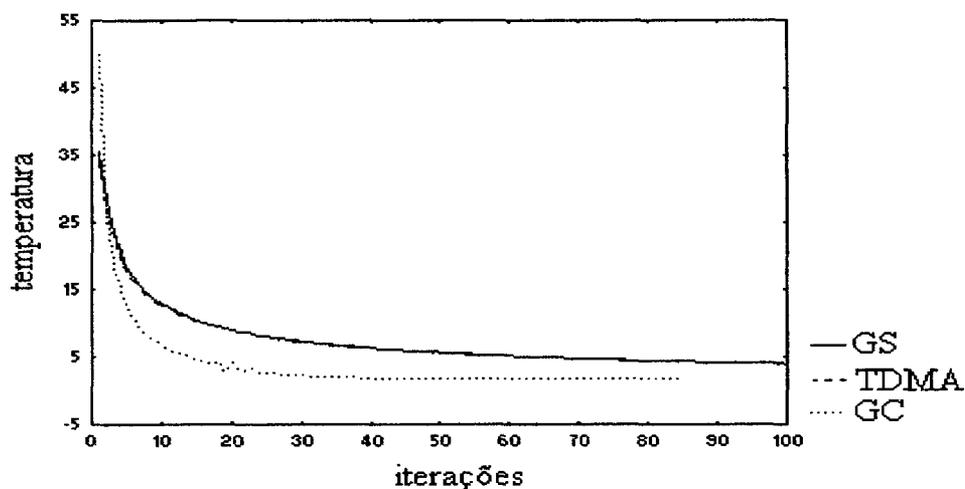


Figura 6.17 - Convergência da temperatura, até a centésima iteração, no *sistema c*.

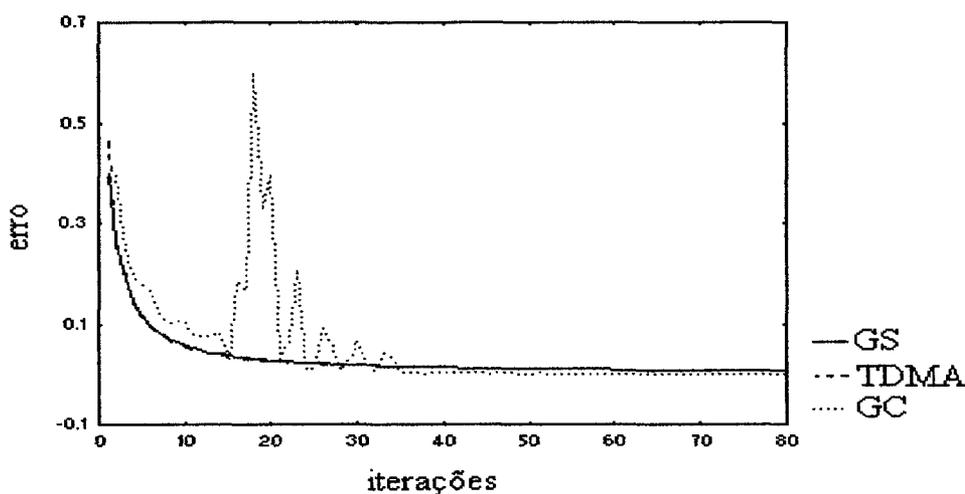


Figura 6.18 - Variação do critério de convergência no *sistema c*.

As ordenações não aumentaram a velocidade de convergência do TDMA, tornando-o inclusive mais lento que o próprio GS, porém seria necessário realizar um refino na malha para uma análise mais precisa dos resultados. Contudo, deve-se analisar se o tempo dispensado à obtenção de uma ordenação não poderia ser dedicado a melhora dos pré-condicionadores existentes na literatura, os quais podem melhorar a velocidade de convergência dos métodos da família do GC.

6.7 Conclusão

No presente capítulo apresentou-se ordenações: em linha horizontal, contínua e diagonal aplicando-se os métodos GS, TDMA e GC na solução da equação da condução de calor. Constatou-se, nos três sistemas abordados, que o método GC resolveu-os de forma mais adequada, com número de iterações e tempo de processamento menores, o TDMA teve desempenho semelhante ao GS. O GS apresentou baixo custo por iteração, mas esta vantagem é perdida pela velocidade de convergência extremamente baixa. As ordenações usadas não melhoraram a performance do TDMA, onde esperava-se um número menor de iterações na convergência.

Finalmente, não se sabe até que ponto pode ser considerado vantajoso estudar e implementar diversas formas de ordenações em malhas com geometrias e discretizações diversificadas, enquanto poder-se-ia estudar de forma mais detalhada os pré-condicionamentos, que já são reconhecidamente *aceleradores* da velocidade de convergência da família do GC.

Capítulo 7 - Resultados: Análise do GC com pré-condicionamento

7.1 Introdução

Para comparar o desempenho do GC pré-condicionado, empregado na solução da equação da condução de calor, são utilizados três conjuntos de sistemas lineares, cujas matrizes de coeficientes são simétricas positivas definidas, resultantes de malhas não-estruturadas com domínios variados.

7.2 Geometrias

As geometrias utilizadas no presente capítulo tem formatos variados. A geometria em formato L utiliza uma malha gerada a partir de pontos nodais distribuídos aleatoriamente com os volumes podendo ser distorcidos ou irregulares, seguindo as propriedades do DV. A geometria triangular utiliza uma malha com volumes de controle hexagonais e a cavidade quadrada tem seu domínio modelado por malha com volumes de controle retangulares. As geometrias computacionais apresentadas na figura 3.4 são discretizadas por malhas não-estruturadas geradas por DV apresentadas nas figuras (7.1) a (7.3).

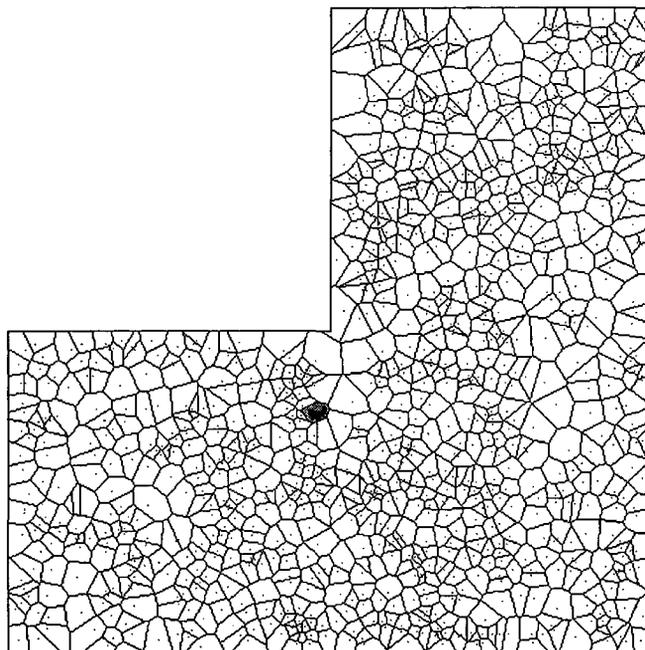


Figura 7.1 - Malha aleatória com 1003 volumes, geometria em formato L.

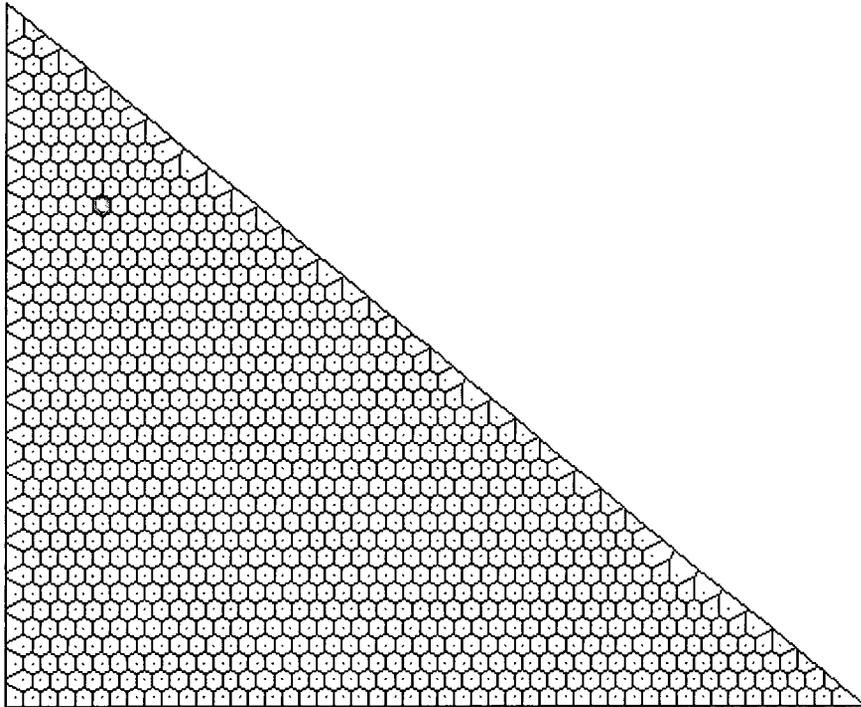


Figura 7.2 - Malha hexagonal com 976 volumes, geometria triangular.

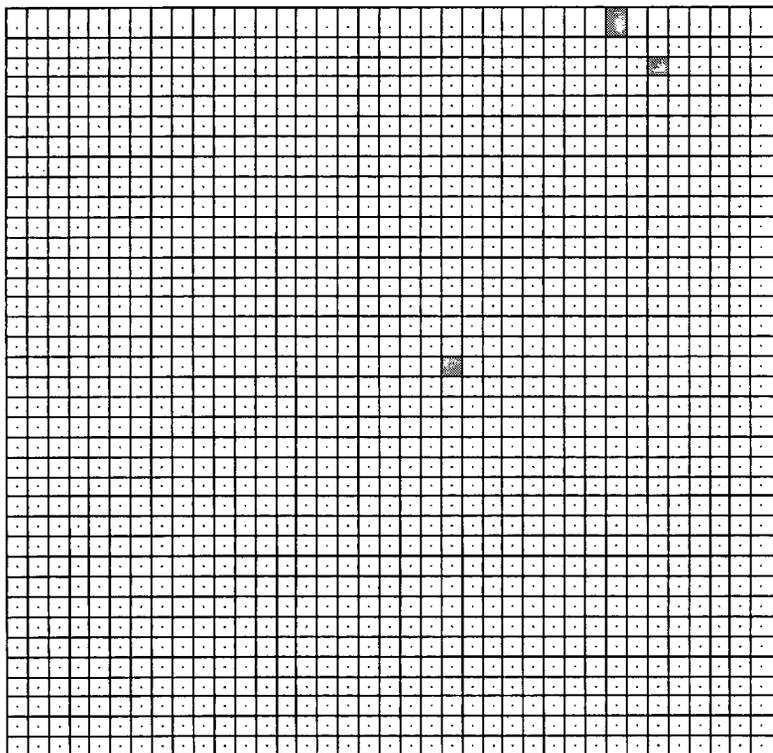


Figura 7.3 - Malha retangular com 1369 volumes, geometria quadrada.

As características dos sistemas de equações algébricas lineares, resultantes da discretização da equação da condução de calor (equação (3.10)) nas malhas não-estruturadas, são apresentadas na tabela 7.1. Os sistemas apresentados têm dimensões relativamente pequenas e o número de condição é baixo, sendo que a maior condição é encontrado no sistema 1.

Tabela 7.1 - Descrição dos sistemas resultantes da discretização da equação do calor.

Identificação da geometria	ordem	ne	λ_{max}	λ_{min}	κ_2 =número de condição
formato L (<i>sistema 1</i>)	1003	6789	6,1884	0,0046	1345,3
formato triangular (<i>sistema 2</i>)	976	6560	7,8949	1,0027	7,87
quadrada retangular (<i>sistema 3</i>)	1369	6697	8,4905	0,0071	1195,84

onde *ne* é o número de elementos não nulos

Na tabela 7.2 são apresentados os resultados da resolução dos sistemas com o método GC fazendo uso dos pré-condicionamentos diagonal (GCPD), SSOR (GCPSSOR) e Cholesky Incompleto sem “fill-ins” (GCPCI). A tolerância usada como critério de convergência é descrita na equação (7.1),

$$\text{erro}^k = \frac{\|r^k\|_2}{\|r^0\|_2} \leq \xi \tag{7.1}$$

onde $\xi = 10^{-5}$.

Na tabela 7.2 o desempenho computacional dos métodos iterativos foram avaliados em uma plataforma Sun Ultra, comparando número de iterações e tempo de processamento. Observa-se que alguns sistemas de equações algébricas lineares não são resolvidos em poucos passos iterativos.

Tabela 7.2 - Resultados do GC pré-condicionado.

Sistemas	GC		GCPD		GCPSSOR $\omega = 1,5; \psi = 1$		GCCCI	
	iter	t	iter	t	iter	t	iter	t
<i>sistema 1</i>	186	4,16	139	3,54	139	230,27	126	286,33
<i>sistema 2</i>	96	1,17	86	1,27	86	98,07	64	105,35
<i>sistema 3</i>	72	0,80	68	0,98	68	91,00	50	98,04

onde iter é o número de iterações necessários para alcançar a convergência e t é o tempo em segundos necessário para atingir a convergência;

Na tabela 7.2 verifica-se que a técnica de pré-condicionamento diagonal é eficiente, pois ela permite resolver todos os sistemas em menor número de iterações que o GC, porém com tempo de processamento superior ao obtido com o GC, nos sistemas 1 e 2. O pré-condicionamento diagonal é favorecido porque a matriz de coeficientes é diagonalmente dominante.

O SSOR, usando $\omega = 1,5$ e $\psi = 1$ encontra resultados semelhante aos obtidos com o pré-condicionamento diagonal, vindo de encontro com as afirmações de Papadrakakis (1993). Com $\psi = 0$ o desempenho é igual ao método SSOR. Será mostrado na seção 7.4 que o SSOR reduz os autovalores maiores quase para a unidade, mas comumente possui autovalores pequenos, que causam problemas de convergência. Note que nos pré-condicionadores SSOR e diagonal não são requeridas armazenagem e tempo extra, para construir o pré-condicionador, contudo o pré-condicionamento SSOR torna-se mais lento que o diagonal.

No GCPCI sem “fill-ins” o tempo de processamento é maior que o encontrado com os demais pré-condicionadores, e o número de iterações é razoável. Infelizmente se o sistema não tem uma M-matriz que satisfaça a equação (1.1), a fatoração pode falhar pelo fato da matriz ter uma diagonal imaginária. Espera-se um desempenho melhor deste pré-condicionamento, quando os sistemas lineares tiverem dimensões maiores.

7.3 Análise do número de condição

O pré-condicionamento explícito da equação $Ax = b$ é formado por:

$$(P_E^{-1} A P_D^{-T})(P_D^T x) = P_E^{-1} b \quad (7.2)$$

Portanto, é possível calcular todos os autovalores da matriz pré-condicionada, o que é interessante para conhecer a distribuição dos autovalores e a condição da matriz pré-condicionada. Assim, tem-se:

- (i) pré-condicionamento diagonal $P_E = \text{diag}(A)^{1/2}$, vide equação (5.7);
- (ii) pré-condicionamento SSOR, $P_E = (D + \omega B)D^{-1/2}$, encontrado na equação (5.22);
- (iii) pré-condicionamento por fatoração incompleta de Cholesky $P_E = L^{-1}$, vide equação (5.25).

Por razões práticas a análise gráfica será restrita apenas ao pré-condicionador diagonal aplicado a dois sistemas. A distribuição dos autovalores do *sistema 2* sem pré-condicionamento e com o pré-condicionamento diagonal é apresentada na figura 7.4. Observa-se que com o pré-condicionamento os autovalores tendem a agrupar-se em um intervalo próximo da unidade, porém alguns autovalores não conseguem se agrupar desta forma.

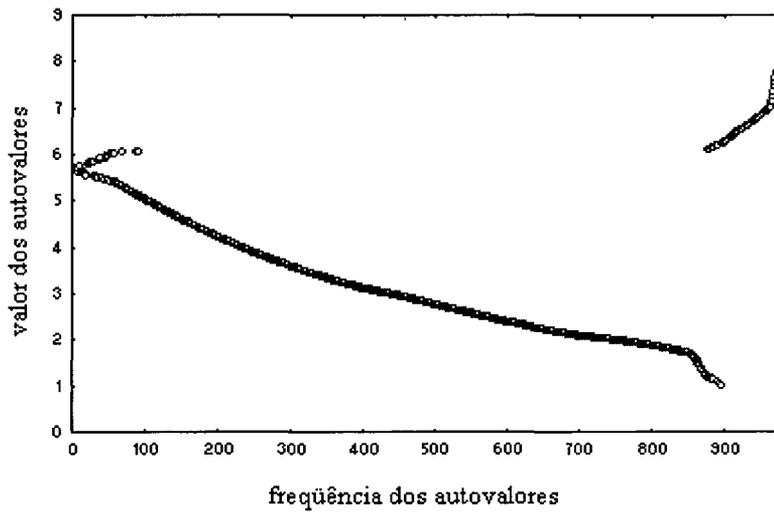


Figura 7.4.a - Autovalores do *sistema 2* sem pré-condicionamento.

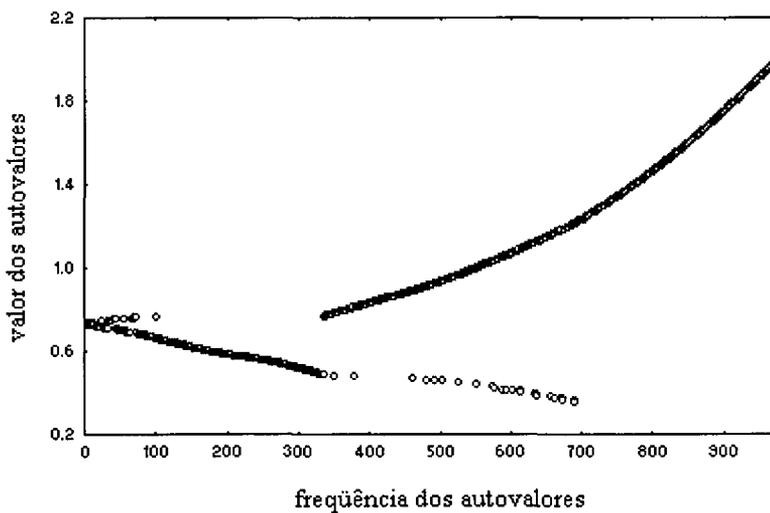


Figura 7.4.b - Autovalores do *sistema 2* com PD⁶.

A figura 7.5 apresenta a distribuição dos autovalores, sob outra visão, sem o pré-condicionamento e com o pré-condicionamento diagonal para o *sistema 2*, onde nota-se que o pré-condicionamento diagonal agrupou os autovalores próximos da unidade.

⁶ PD significa pré-condicionamento diagonal.

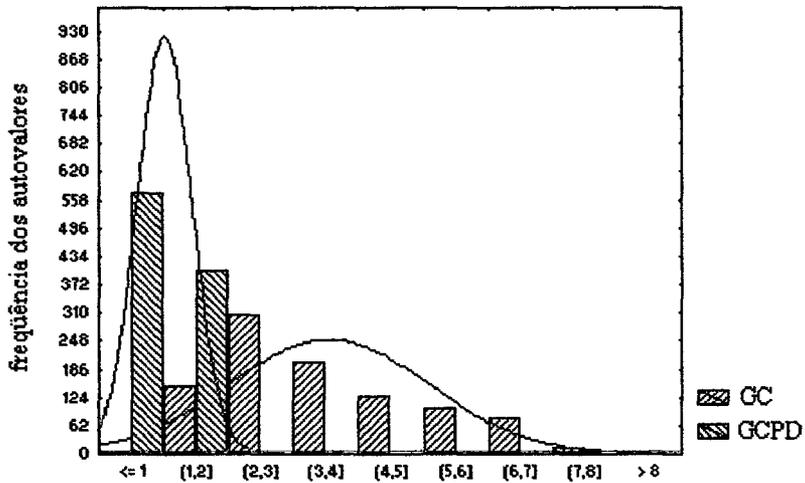


Figura 7.5 - Comparativo dos autovalores sem pré-condicionamento e com PD, *sistema 2*.

Analisando a distribuição dos autovalores para o sistema original e com pré-condicionamento diagonal no *sistema 3* (vide figuras 7.6 e 7.7), nota-se que os autovalores iniciais tendem a ficar mais concentrados, contudo restam autovalores pequenos distantes do “cluster” o que resulta em um aumento pequeno da velocidade de convergência do GC.

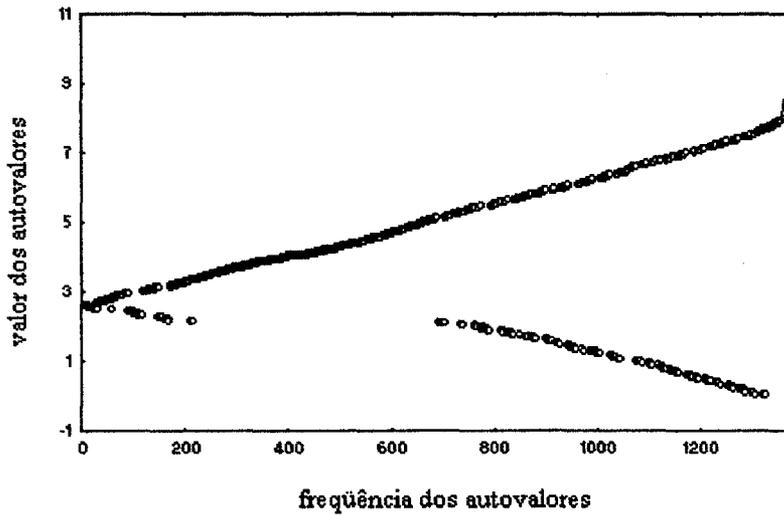


Figura 7.6.a - Autovalores do *sistema 3* sem pré-condicionamento.

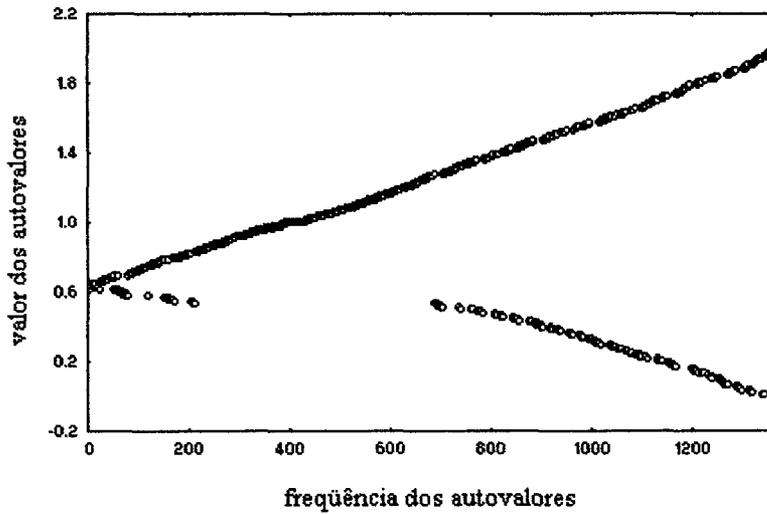


Figura 7.6.b - Autovalores do *sistema 3* com PD.

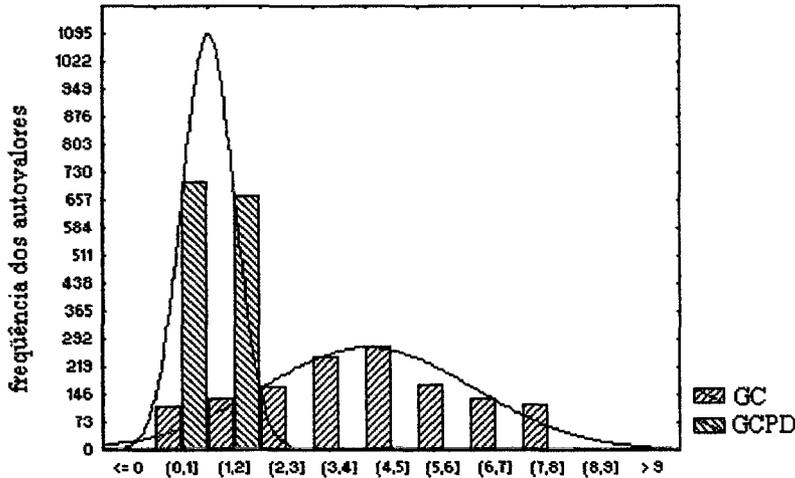


Figura 7.7 - Comparativo dos autovalores sem pré-condicionamento e com PD, *sistema 3*.

7.4 Efeitos do pré-condicionamento

A tabela 7.3 apresenta o maior autovalor (λ_{max}), o menor autovalor (λ_{min}) e o número de condição (κ_2) da matriz pré-condicionada usando os pré-condicionamentos descritos anteriormente. Para estes casos testados, todos os pré-condicionadores produziram o maior autovalor próximo da unidade. Além disso, o escalonamento diagonal e o SSOR (com $\omega = 1,5$ e $\psi = 0,8$) as vezes dão autovalores muito pequenos. A fatoração de Cholesky incompleta sem “fill-ins” (CI(0)) também produz o número de condição menor, produzindo o autovalor máximo próximo da unidade.

Tabela 7.3 - Autovalores das matrizes pré-condicionadas.

Sistemas	Diagonal			SSOR ($\omega = 1,5; \psi = 0,8$)			CI(0)		
	$\lambda_{\text{máx}}$	λ_{min}	κ_2	$\lambda_{\text{máx}}$	λ_{min}	κ_2	$\lambda_{\text{máx}}$	λ_{min}	κ_2
<i>sistema 1</i>	1,9876	0,0019	1046,1	1,9984	0,0018	1110,2	1,002	0,0011	910,9
<i>sistema 2</i>	1,9973	0,3372	5,92	1,9981	0,3273	6,1048	1,05	0,3043	3,45
<i>sistema 3</i>	1,9982	0,0018	1110,1	1,9991	0,0017	1175,9	1,1	0,0012	916,7

As figuras 7.8 a 7.10 representam a distribuição dos elementos da matriz de coeficientes nos *sistemas 1, 2 e 3* respectivamente.

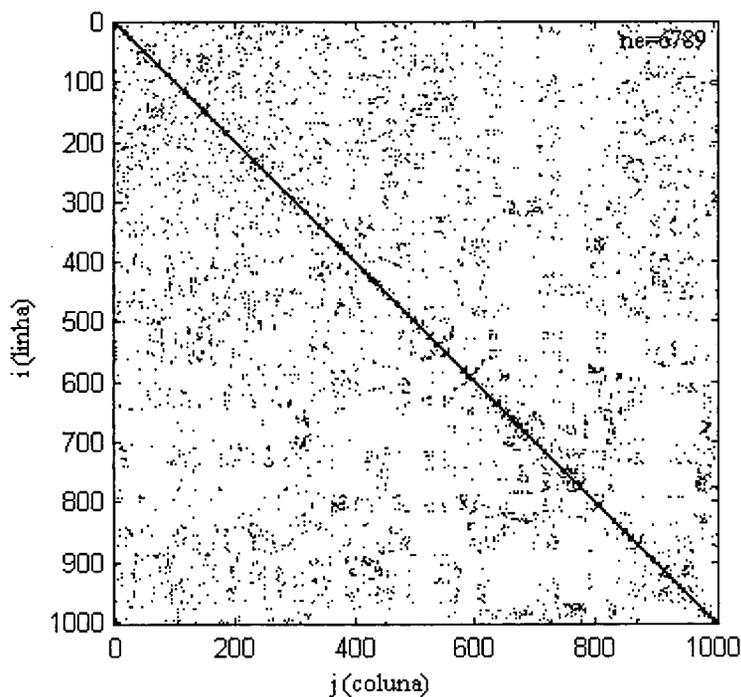


Figura 7.8 - Representação matricial (i, j) do *sistema 1*, (n = 1003).

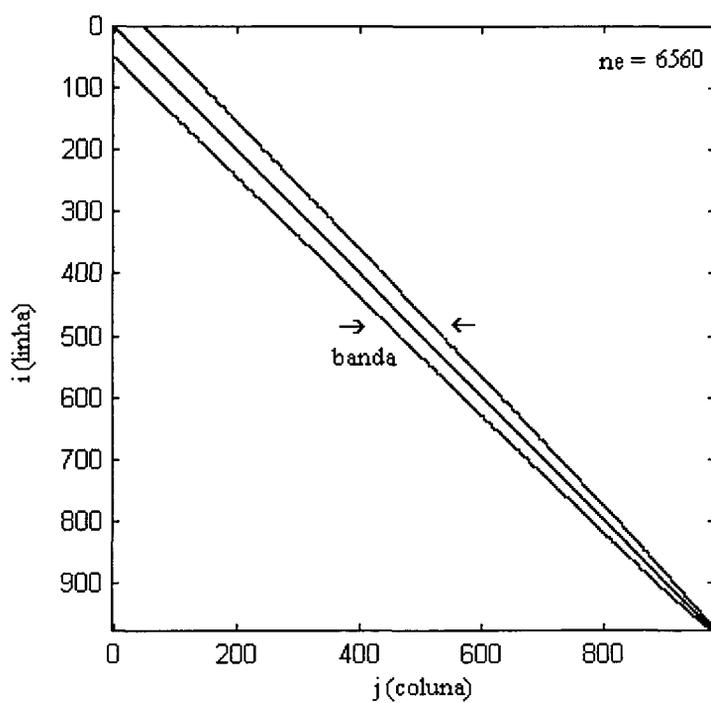


Figura 7.9 - Representação matricial (i, j) do *sistema 2*, (n = 976).

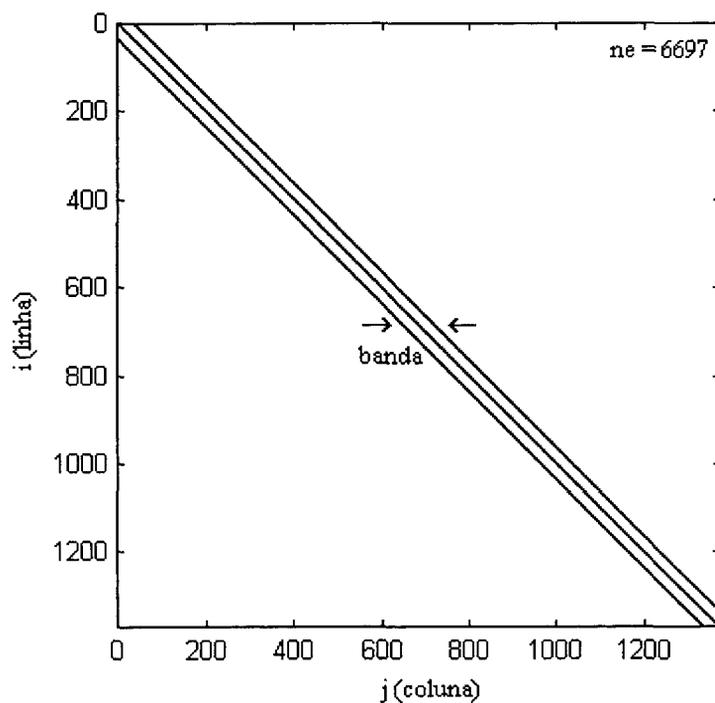


Figura 7.10 - Representação matricial (i, j) do *sistema 3*, (n = 1369).

7.5 Análise dos resultados

Na figura 7.11 apresenta-se o comportamento da temperatura no ponto nodal (41,6; 47,7), identificado na figura 7.1, na geometria em formato L com malha aleatória. De acordo com o gráfico o método GC, GCPD e GCPSSOR apresentaram comportamento inferior ao GCPCI, que exigiu menor número de iterações para convergir. O comportamento do resíduo pré-condicionado, é apresentado na figura 7.12, observando-se que este é bastante oscilatório para o GC e GCPCI, se comparado com os demais, no entanto convergem com menor número de iterações. A figura 7.13 detalha esta evolução do resíduo até a centésima iteração.

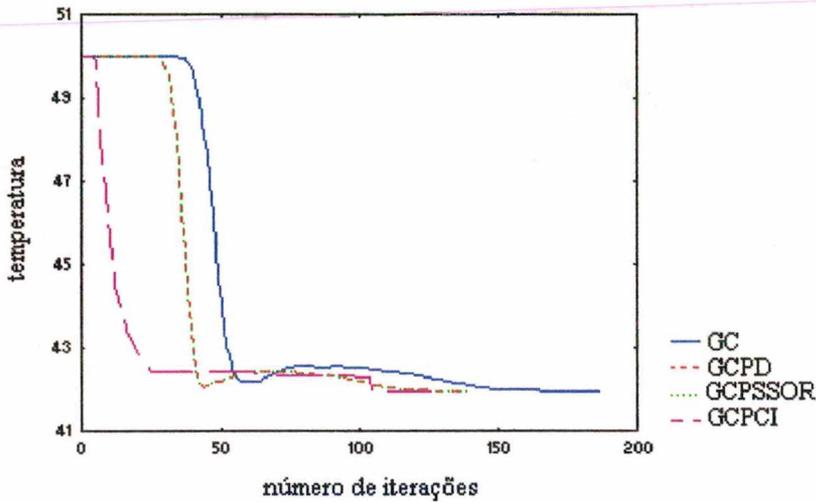


Figura 7.11 - Processo iterativo para a temperatura no *sistema 1*.

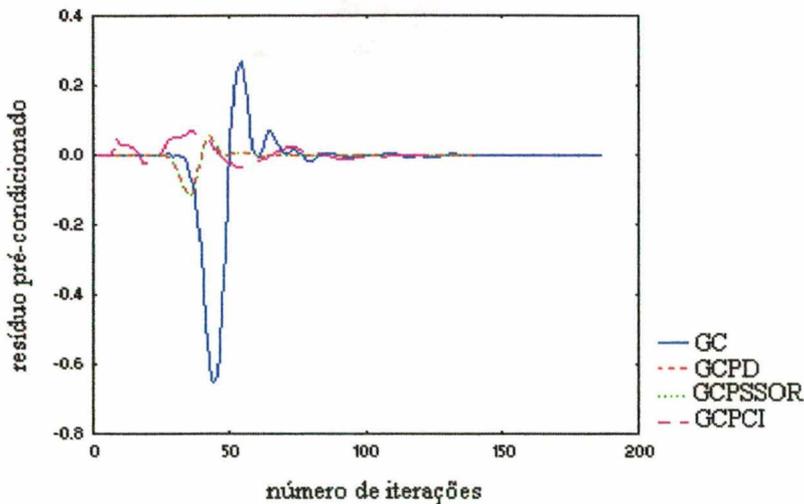


Figura 7.12 - Comportamento do resíduo pré-condicionado no *sistema 1*.

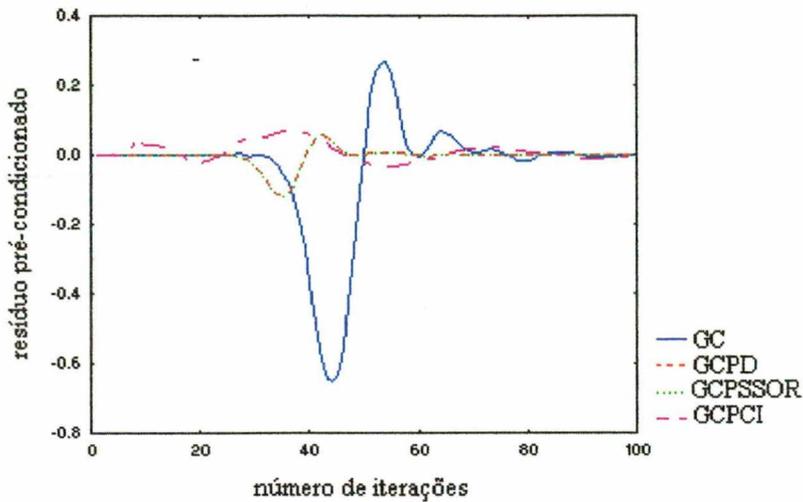


Figura 7.13 -. Comportamento do resíduo pré-condicionado, até a centésima iteração.

A figura 7.14 expõe a temperatura no ponto (2,85; 11,65) identificado na figura 7.2, na geometria em formato triangular. Verifica-se que a temperatura é semelhante a temperatura de condição prescrita 100°C , pois o ponto escolhido para a análise está próximo do contorno. De acordo com o gráfico o GCPD e GCPSSOR apresentaram comportamentos semelhantes e o GCPCI apresentam desempenho um pouco superior. A figura 7.15 apresenta o processo iterativo até a quinquagésima iteração. O comportamento do resíduo pré-condicionado durante o processo iterativo, é apresentado na figura 7.16 e é bastante oscilatório para o GCPCI, se comparado com os demais pré-condicionamentos.

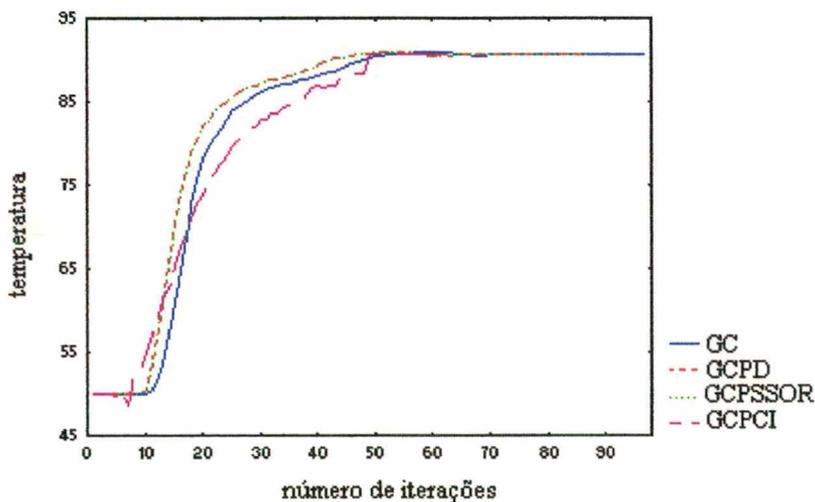


Figura 7.14 - Temperatura nas primeiras iterações, *sistema 2*.

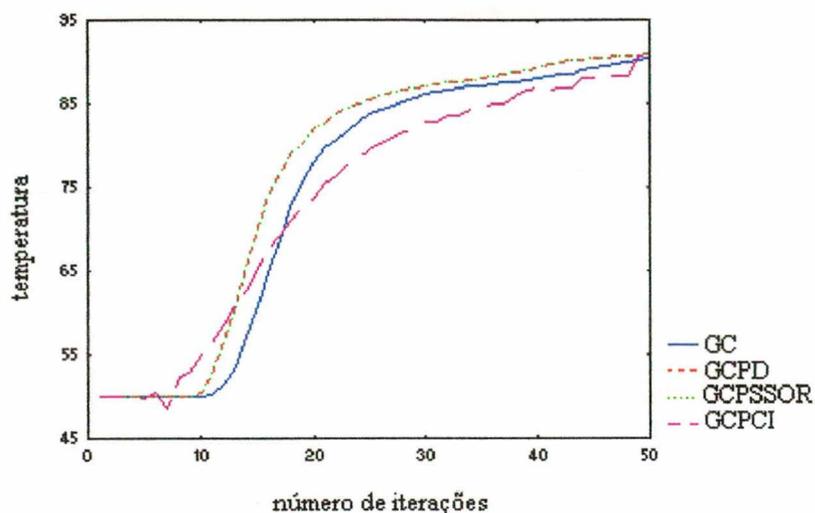


Figura 7.15 - Convergência da temperatura no *sistema 2* até a quinquagésima iteração.

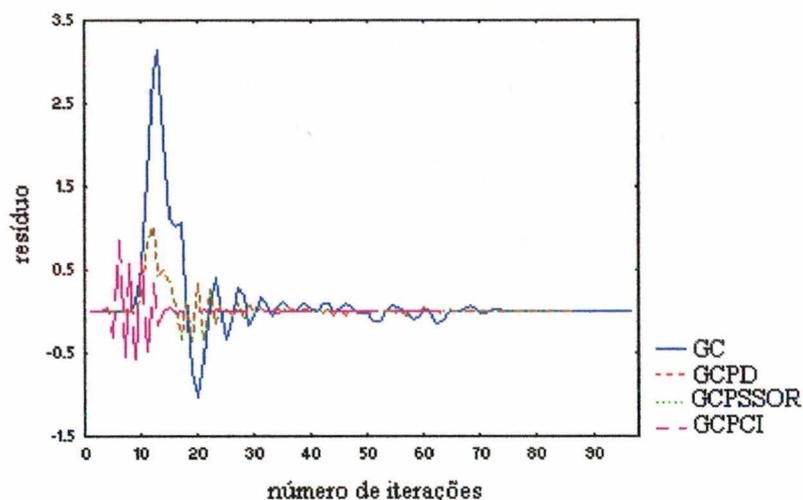
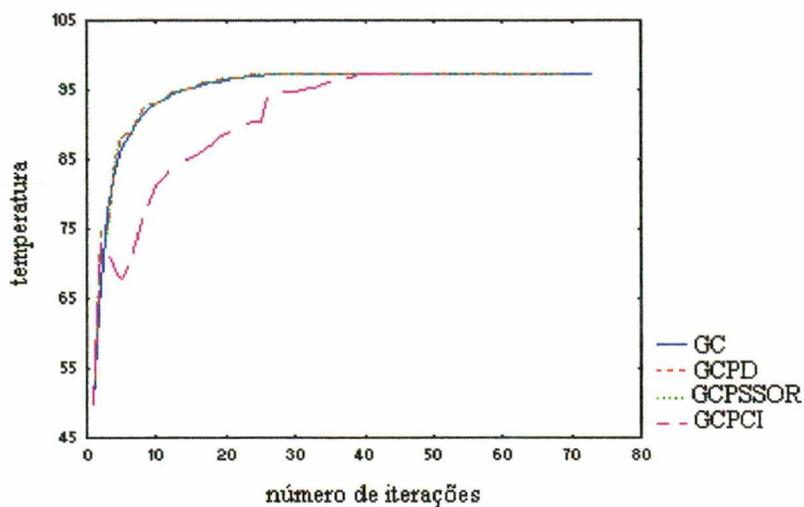
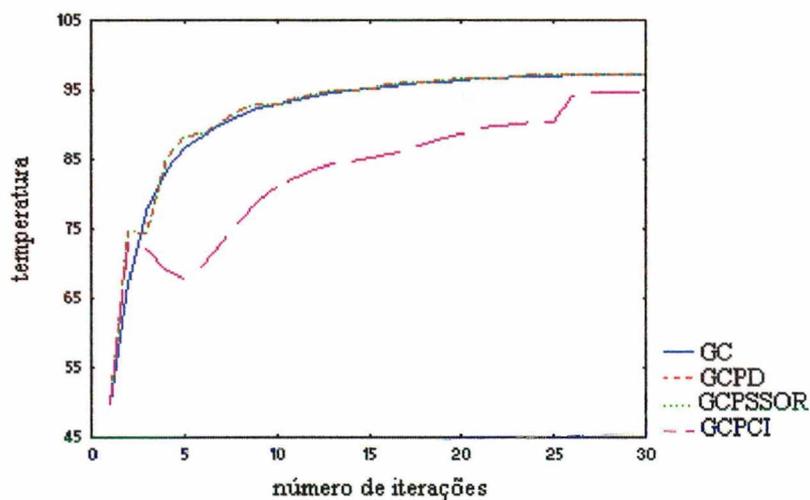


Figura 7.16 - Comportamento do resíduo pré-condicionado, *sistema 2*.

Na figura 7.17 é apresentado o processo de convergência da temperatura no ponto nodal (16,8; 17,9) identificado na figura 7.3, *sistema 3*, verifica-se que a temperatura é semelhante a temperatura prescrita 100°C, pois o ponto está próximo do contorno da geometria. De acordo com o gráfico o método GC, GCPD e GCPSSOR apresentaram comportamento pouco inferior ao GCPCI, que por sua vez exigiu menor número de iterações para convergir. A figura 7.18 apresenta o processo iterativo até a trigésima iteração.

Figura 7.17 - Processo iterativo para a temperatura no *sistema 3*.Figura 7.18 - Temperatura no *sistema 3* até a trigésima iteração.

Os resíduos pré-condicionados encontrados com os pré-condicionamentos aplicados para a resolução do *sistema 3* são apresentados na figura 7.19. O resíduo até a trigésima iteração é apresentado na figura 7.20, observando-se que este é bastante oscilatório para o GCPCI, se comparado com os demais pré-condicionamentos.

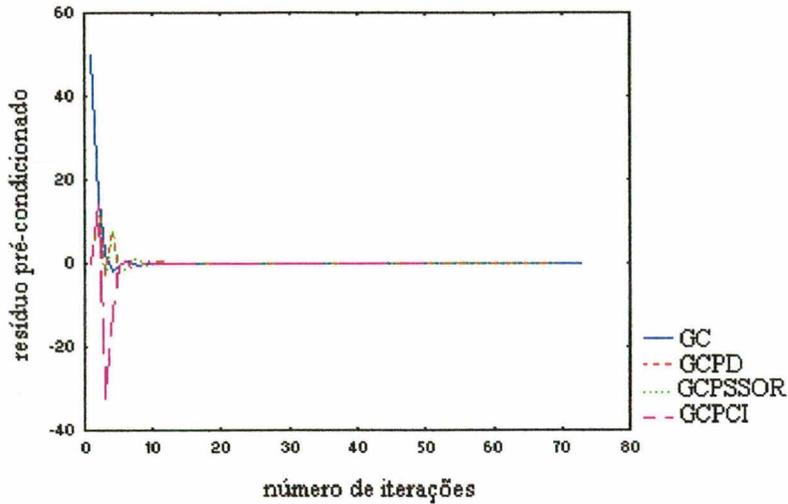


Figura 7.19 - Comportamento do resíduo pré-condicionado, *sistema 3*.

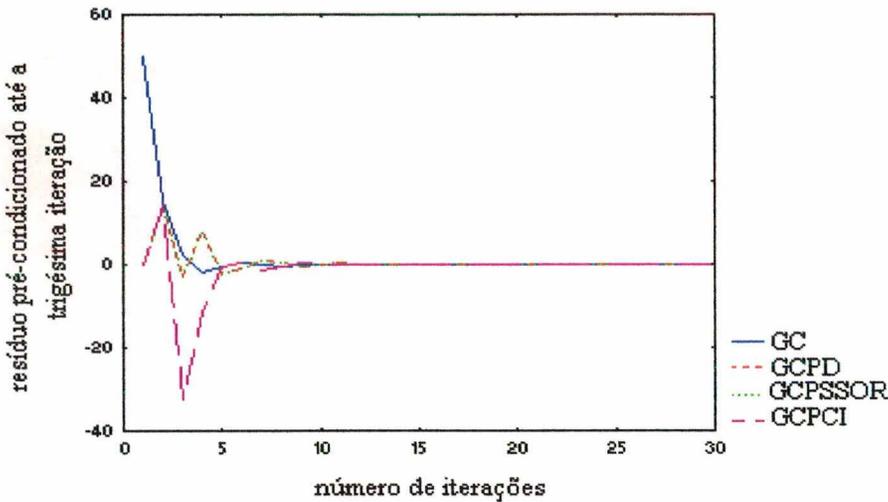


Figura 7.20 - Resíduo pré-condicionado até a trigésima iteração, *sistema 3*.

Na figura 7.21 expõe-se os resultados encontrados para a temperatura no ponto nodal (12,6; 13,4), identificado na figura 7.3, aplicando o GCPSSOR utilizando valores diferenciados para ω e ψ . Quando $\psi = 0$ o método tem comportamento semelhante ao método SSOR com convergência lenta, conforme observado por Papadrakakis (1993). A variação do fator de relaxação ω com ψ mantido como sendo 1, não ocasionou alterações no processo de convergência.

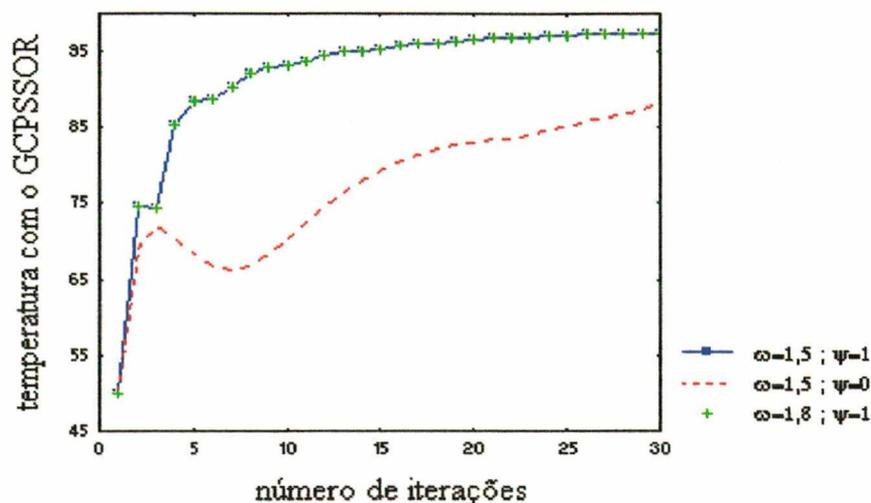


Figura 7.21 - Convergência da temperatura utilizando o GCPSSOR.

7.6 Conclusão

Neste capítulo foi empregado o método Gradiente Conjugado, juntamente com algumas técnicas de pré-condicionamentos para a resolução do sistema resultante da discretização da equação da condução de calor em malhas de Voronoi.

A técnica mais simples de pré-condicionamento é a diagonal, tal pré-condicionamento remove os efeitos negativos das grandes variações em magnitude dos elementos diagonal, que para matrizes diagonalmente dominante pode causar um número de condição alto.

O pré-condicionamento diagonal tem um desempenho bastante adequado para matrizes diagonalmente dominante, conforme observado neste capítulo, porém esta é uma aplicação específica e o pré-condicionamento diagonal não é, normalmente, eficiente. Contudo, o pré-condicionamento diagonal pode tornar-se menos efetivo quando os elementos fora da diagonal assumem importância. (Behr and Tezduyar, 1994). Este fato ocorre na prática em problemas onde o número de Reynolds cresce.

O pré-condicionamento SSOR ($\omega=1,5; \psi=1$) convergiu igual ao pré-condicionamento diagonal, sendo necessário o mesmo número de iterações, porém com o tempo de processamento maior. Mudando os parâmetros ω , ψ o método tende a convergir de forma mais lenta e com maior número de iterações.

O pré-condicionamento por fatoração de CI diminuiu o número de condição dos três sistemas convergindo com menor número de iterações porém com o tempo de processamento maior. Logo, no presente caso o pré-condicionamento por fatoração de CI não é viável devido

seu custo computacional e porque não trouxe novas contribuições. Espera-se que este pré-condicionamento tenha um desempenho superior quando aplicado em malhas refinadas, e utilizando níveis de enchimento entre 1 e 3, conforme Marcondes (Marcondes, 1996), o que não será abordado no presente trabalho.

Capítulo 8 - Resultados numéricos: equações de Navier-Stokes

8.1 Introdução

Os resultados numéricos obtidos no presente capítulo têm como finalidade verificar a eficiência dos métodos da família do GC, descritos no capítulo 4, para a resolução das equações discretizadas de Navier-Stokes. Tais resultados são validados utilizando os valores encontrados no trabalho de Ghia *et al* (1982). Foram aplicados diferentes esquemas de avaliação de gradiente de pressão para fins de comparação de desempenho e avaliação do comportamento de cada gradiente.

8.2 Análise dos métodos da família do GC

O teste utilizado para validação dos métodos numéricos, é o escoamento bidimensional em uma cavidade de seção quadrada de largura infinita, sujeita a uma parede superior deslizante, semelhante a uma esteira. A especificação da cavidade quadrada, juntamente com as condições de contorno é apresentada na figura 8.1.

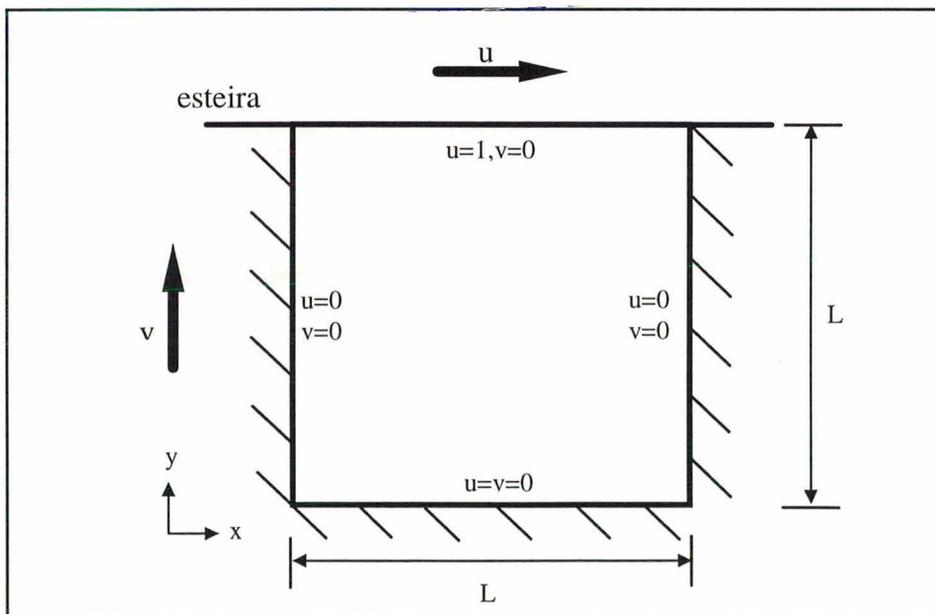


Figura 8.1 - Cavidade quadrada com parede superior deslizante e condições de contorno.

Na obtenção das soluções foram simulados escoamentos onde o número de Reynolds assumiu valores 100, 400 e 1000. As malhas são formadas por 50x49 pontos com disposição

hexagonal conforme figura 6.7, 37×37 pontos com disposição retangular apresentada na figura 7.3 e a malha aleatória formada por 530 volumes de controle.

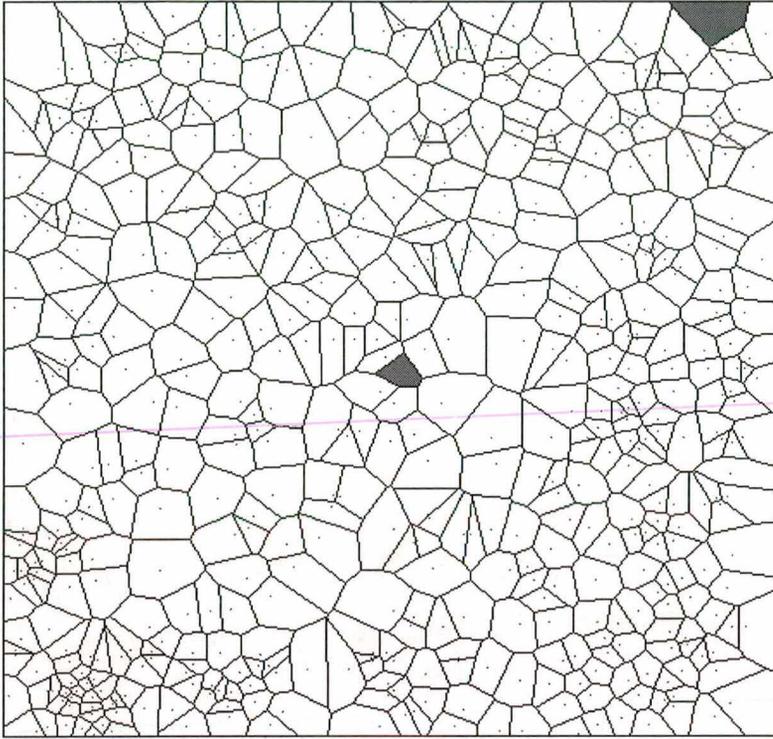


Figura 8.2 - Malha aleatória com 530 volumes em geometria quadrada.

A velocidade de convergência, dos métodos utilizados, está relacionada ao fator de relaxação ω empregado e, ao número de vezes em que cada uma das variáveis são resolvidas durante cada iteração. O fator de sub-relaxação ótimo é usualmente escolhido em função do tamanho da malha. De acordo com Tanyi e Thatcher, para malhas de dimensões 36×36 o valor ótimo encontrado para o fator de sub-relaxação é aproximadamente 0,8 para ω_u e ω_v para as equações (3.52) e (3.53). Em malhas com maior número de pontos recomenda-se 0,9 para ω_u e ω_v . O fator de sub-relaxação utilizado neste trabalho é 0,5 e os sistemas para o cálculo das velocidades u (equação (3.52)), v (equação (3.53)) e para a correção da pressão (equação (3.54)) são resolvidas de maneira desacoplada para cada uma das variáveis.

A tabela 8.1 apresenta o desempenho dos diferentes métodos da família do GC utilizados para resolver os sistemas linearizados resultantes da discretização das equações de Navier-Stokes em malhas de Voronoi. Os resultados são coletados com Reynolds 100, utilizando para o cálculo dos gradientes de pressão o MRQ, MPMGNP e MPMGNP (Taniguchi *et al.*, 1991; Taniguchi, and Kobayashi, 1991; Cardoso, 1997). Os sistemas são resolvidos por um laço iterativo interno, formado por uma iteração para cada variável.

Tabela 8.1 - Processo iterativo nas equações de Navier-Stokes com Reynolds = 100.

Métodos		CGS		Bi-CGSTAB		TFQMR		GC	
malhas		iter	t	iter	t	iter	t	iter	t
gradientes									
MRQ	<i>r</i>	1624	4,25	1166	3,15	2918	9,09	762	1,51
	<i>h</i>	2734	17,35	2197	13,34	4300	31,57	739	4,19
	<i>a</i>	4481	6,15	1560	2,11	6724	28,40	661	0,50
MPGNP	<i>r</i>	1624	4,31	1166	3,15	2918	9,30	918	2,06
	<i>h</i>	3104	20,35	2335	15,31	4845	34,37	1060	6,21
	<i>a</i>	nc	-	nc	-	nc	-	nc	-
MPMGNP	<i>r</i>	1624	5,37	1166	4,08	2918	11,33	932	2,39
	<i>h</i>	3204	25,54	2444	19,54	4620	40,23	789	5,53
	<i>a</i>	nc	-	2505	4,27	9971	37,43	1188	1,57

onde *r* representa a malha retangular, *h* a malha hexagonal, a *a* malha aleatória, nc significa que o método numérico não convergiu, t é o tempo medido em minutos e iter expressa o número de iterações até alcançar a convergência definida pela equação que segue,

$$\text{erro}^k = |F_{ij}^k| < \xi \quad (8.1)$$

sendo $\xi = 1e^{-4}$, e F_{ij} é o resíduo da conservação da massa definido no capítulo 3.

Cabe lembrar que os resultados deste capítulo foram obtidos utilizando o *sp2*, super-computador disponível no Núcleo de Processamento de Dados da Universidade Federal de Santa Catarina. O *sp2* trabalha com processamento paralelo e serial, sendo que em paralelo o tempo de processamento obtido é menor. Contudo os programas implementados neste trabalho utilizaram apenas o processamento serial.

Comparando, na tabela 8.1, os resultados dentre os quatro métodos da família do GC empregados, o GC é o método mais eficiente porque usa menor tempo de processamento e número de iterações para convergir. Contudo os sistemas resolvidos são assimétricos (exceto o sistema da correção de pressão) e o GC para sistemas cuja matriz é simétrica positiva e definida é inaplicável, porém no presente trabalho o GC convergiu. O Bi-CGSTAB é mais adequado que o CGS e o TFQMR, sendo este último o mais lento para alcançar a convergência.

Entre as diferentes avaliações utilizadas para calcular os gradientes de pressão, o MRQ apresentou os resultados de forma mais rápida, a MPGNP não convergiu para a malha aleatória com nenhum dos métodos utilizados. A MPMGNP apresentou convergência, geralmente, mais lenta que as outras duas formas de avaliação dos gradientes de pressão, contudo para a malha

aleatória convergiu em três métodos iterativos, e na malha hexagonal os métodos TFQMR e GC tiveram desempenho superior a MPGNP.

Dando continuidade a investigação dos métodos da família do GC, são mostradas nas figuras 8.3 a 8.8 perfis de velocidades u e v para Reynolds 100 ao longo de uma linha média vertical e horizontal, respectivamente, para as velocidades na direção x e y , na malha modelada por volumes retangulares. As linhas horizontal e vertical estão localizadas no centro geométrico da cavidade quadrada.

As figuras 8.3 e 8.4 apresentam os valores para as velocidades u e v utilizando o MRQ para o cálculo dos gradientes de pressão, nas figuras 8.5 e 8.6 fez-se uso da MPGNP na avaliação dos gradientes e nas figuras 8.7 e 8.8 a MPMGNP foi aplicada. Ambas velocidades u e v , são comparadas com os resultados obtidos por Ghia *et al.* (Ghia *et al.*, 1982). Nota-se, em uma primeira análise, que os valores obtidos na resolução das equações de Navier-Stokes utilizando os métodos propostos foi semelhante ao encontrado por Ghia *et al.*.

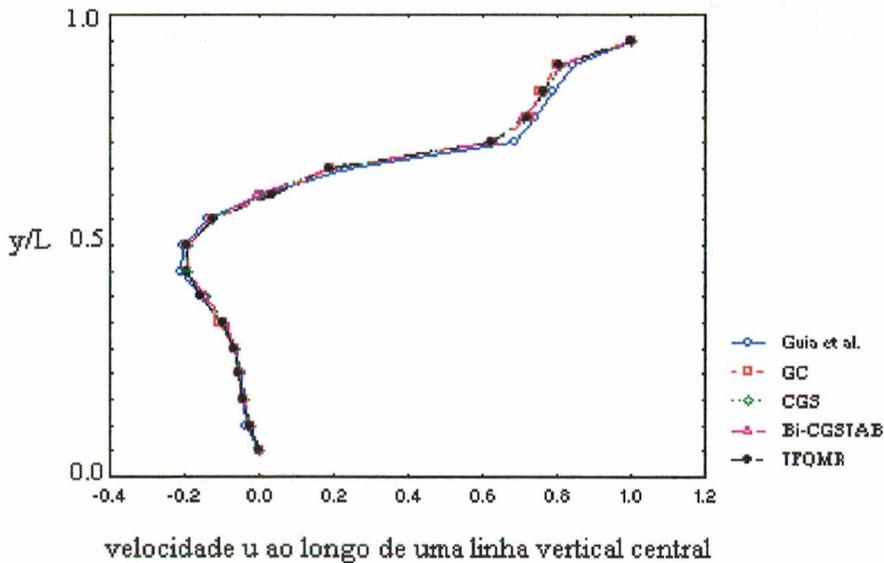


Figura 8.3 - Velocidade u com MRQ, malha retangular, (1369 pontos nodais).

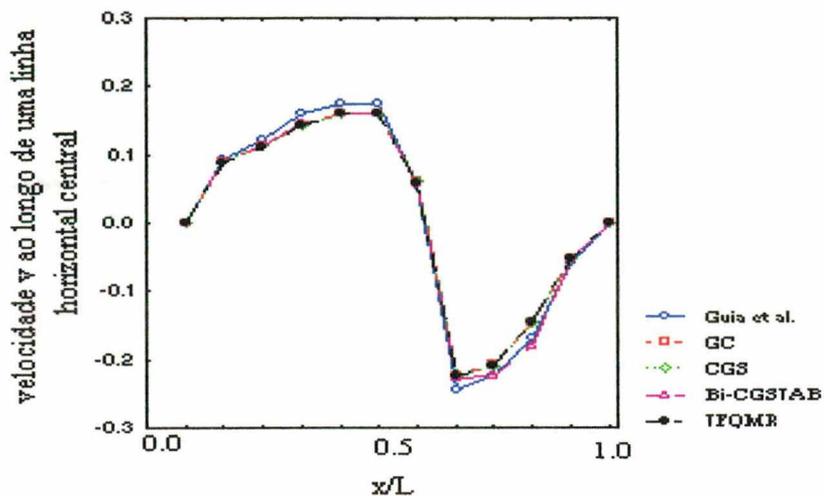


Figura 8.4 - Velocidade v com MRQ, malha retangular, (1369 pontos nodais).

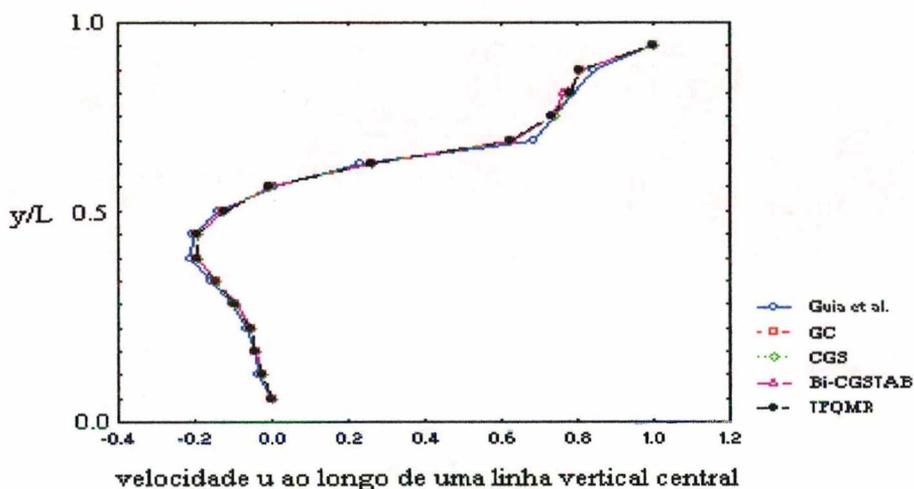


Figura 8.5 - Velocidade u com MPGNP, malha retangular, (1369 pontos nodais).

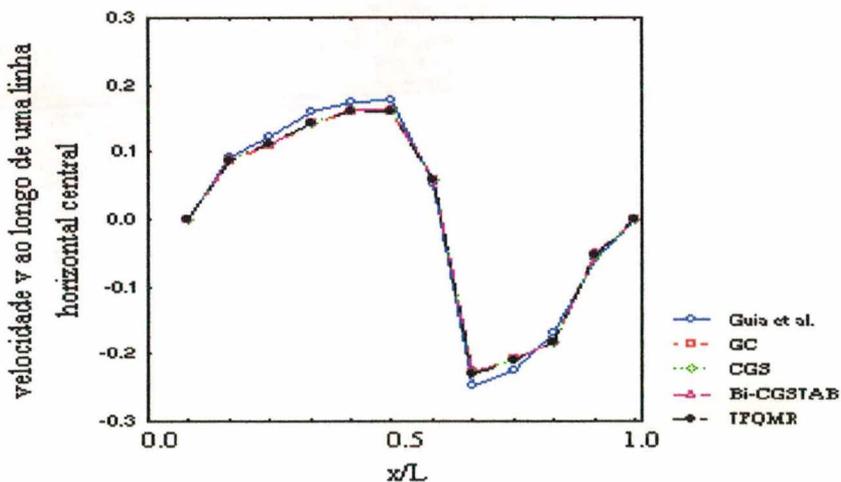


Figura 8.6 - Velocidade v com MPGNP, malha retangular, (1369 pontos nodais).

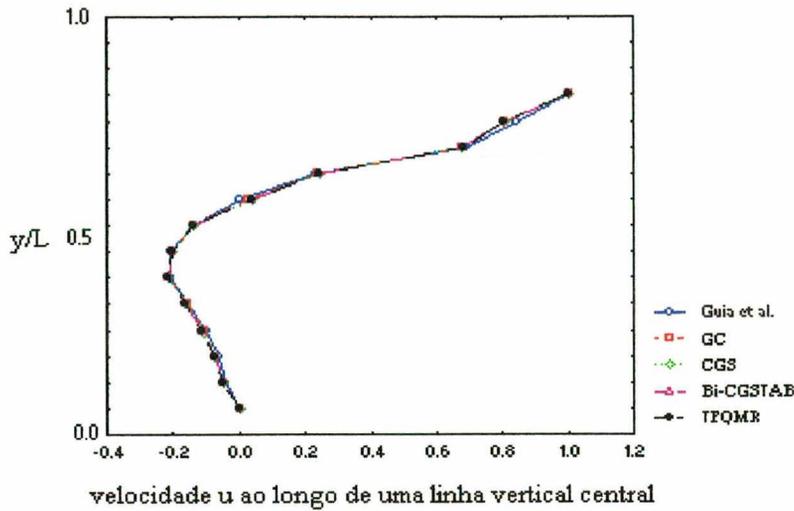


Figura 8.7 - Velocidade u com MPMGPN, malha retangular, (1369 pontos nodais).

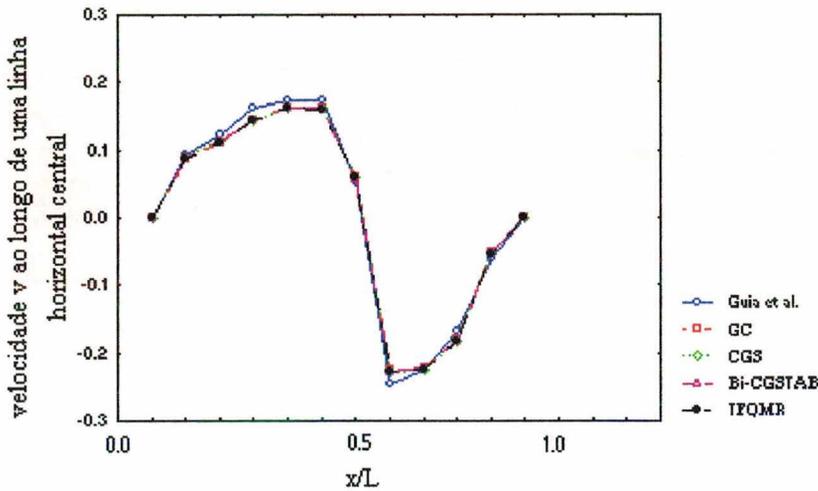


Figura 8.8 - Velocidade v com MPMGPN, malha retangular, (1369 pontos nodais).

Os resultados apresentados até o momento mostram-nos que a MPMGPN, captou com relativa precisão, o perfil das velocidades obtidas por Ghia *et al.* (Ghia *et al.*, 1982), embora em alguns pontos o perfil de velocidade v calculado com MPMGPN manteve-se ligeiramente distante dos resultados de Ghia et al. Desta forma a MPMGPN torna-se atrativa para futuras avaliações.

Nas figuras que seguem os resultados são obtidos utilizando Reynolds=100 em malha hexagonal. Nas figuras 8.9 e 8.10 apresentam-se os valores para as velocidades u e v utilizando o MRQ para o cálculo dos gradientes de pressão, nas figuras 8.11 e 8.12 faz-se uso da MPMGPN na avaliação dos gradientes de pressão e na figuras 8.13 e 8.14 a MPMGPN é aplicado. Observando as figuras, todos métodos iterativos acompanharam os resultados obtidos por Ghia *et al.*, com um comportamento semelhante entre eles.

Nota-se que os valores encontrados com a MPGNP e a MPMGNP, na resolução da equação da velocidade na direção x (velocidade u), foram semelhantes entre si e próximos aos de Ghia *et al.*. Na resolução da equação da velocidade na direção y (velocidade v) a MPGNP e a MPMGNP obtiveram resultados menos adequados, isto é, produziram valores ligeiramente distantes dos encontrados com os outros gradientes.

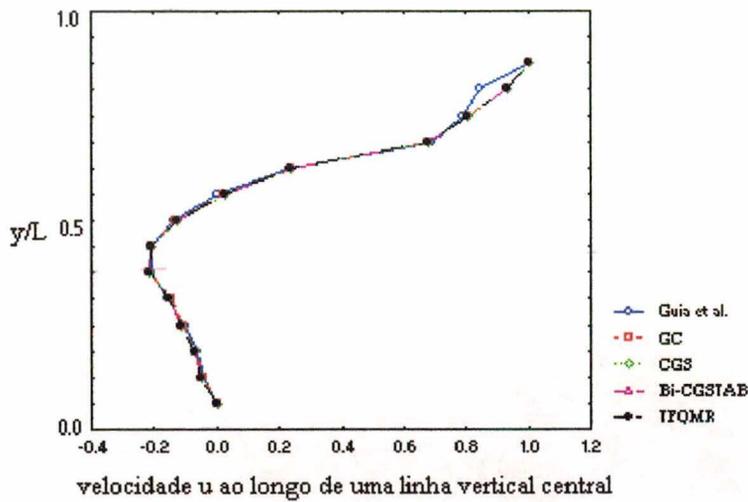


Figura 8.9 - Velocidade u com MRQ, malha hexagonal, (2475 pontos nodais)

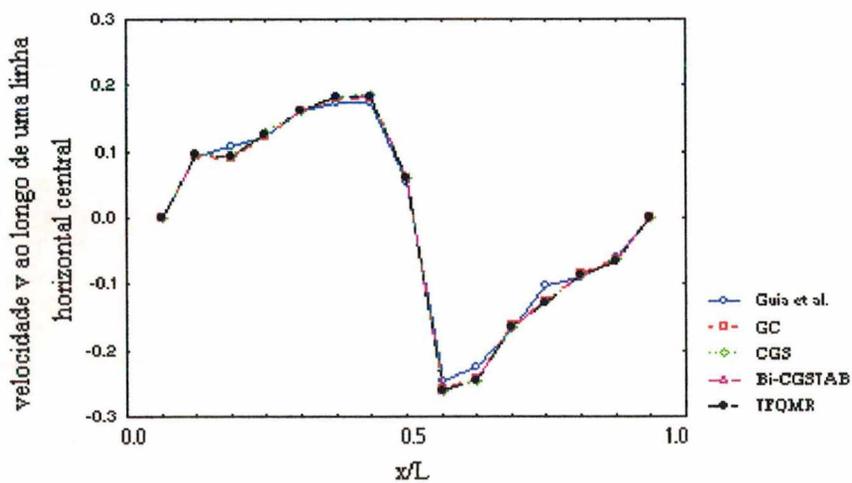


Figura 8.10 - Velocidade v com MRQ, malha hexagonal, (2475 pontos nodais).

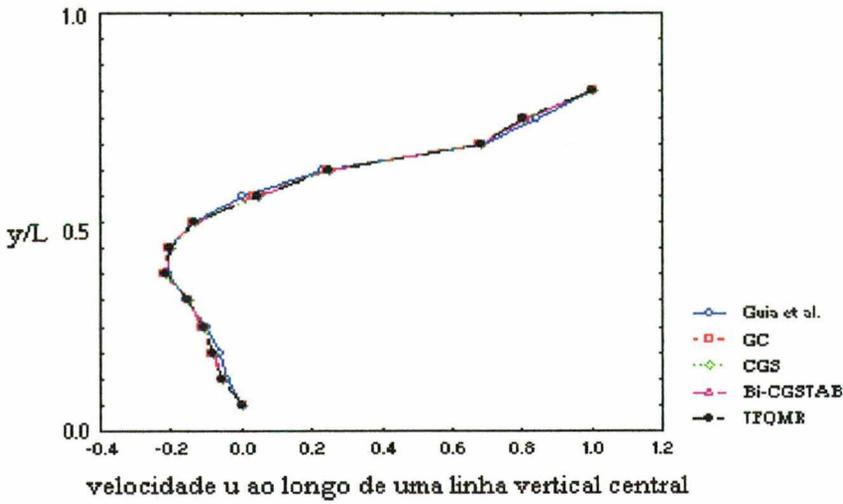


Figura 8.11 - Velocidade u com MPGNP, malha hexagonal, (2475 pontos nodais).

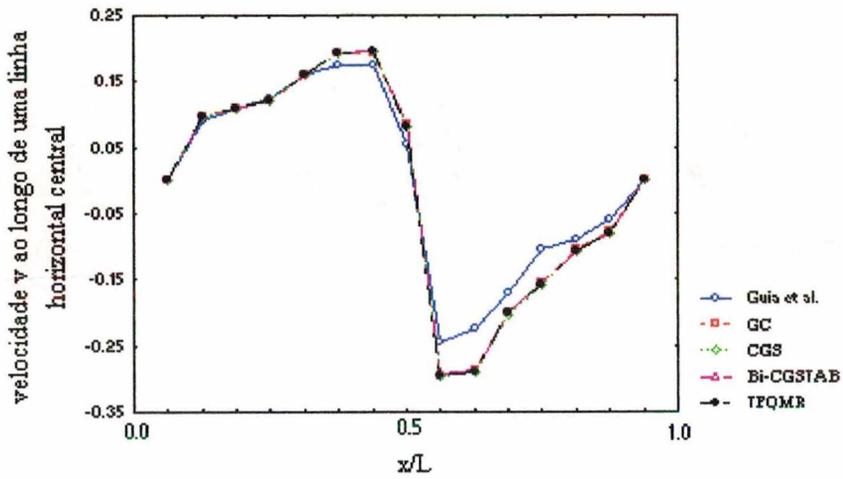


Figura 8.12 - Velocidade v com MPGNP, malha hexagonal, (2475 pontos nodais).

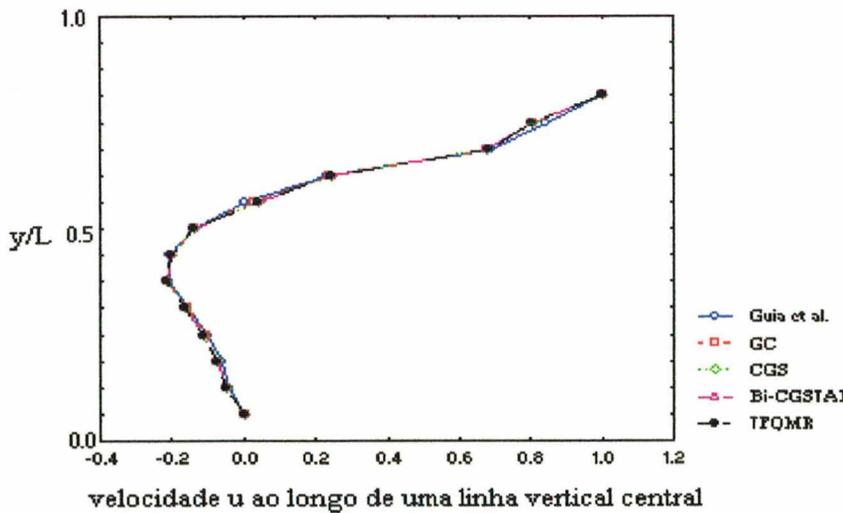


Figura 8.13 - Velocidade u com MPMGNP, malha hexagonal, (2475 pontos nodais).

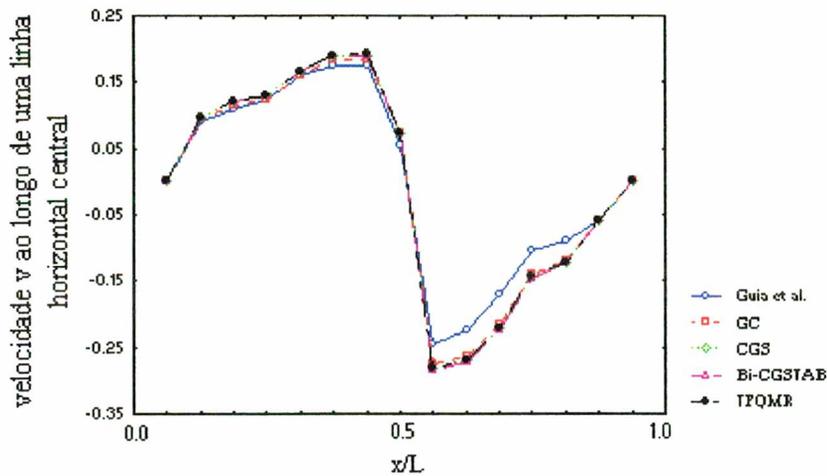


Figura 8.14 - Velocidade v com MPMGNP, malha hexagonal, (2475 pontos nodais).

Os resultados obtidos com o método GC, utilizando malha aleatória, são apresentados nas figuras 8.15 e 8.16. Nestas figuras observa-se que a malha aleatória não obteve resultados próximos dos apresentados por Ghia *et al.* (1982), deve-se considerar entretanto, que esta malha tem dimensões menores que as outras adotadas (retangular e hexagonal). Porém, mesmo tendo dimensões menores os métodos iterativos, quando utilizados com a malha aleatória, tiveram desempenho inferior ao obtido com as outras malhas, quanto ao número de iterações e tempo de CPU, o que pode ser observado na tabela 8.1. A avaliação dos gradientes de pressão pela MPMGNP não convergiu na malha aleatória, fato pelo qual só o MRQ e a MPMGNP têm os resultados apresentados graficamente.

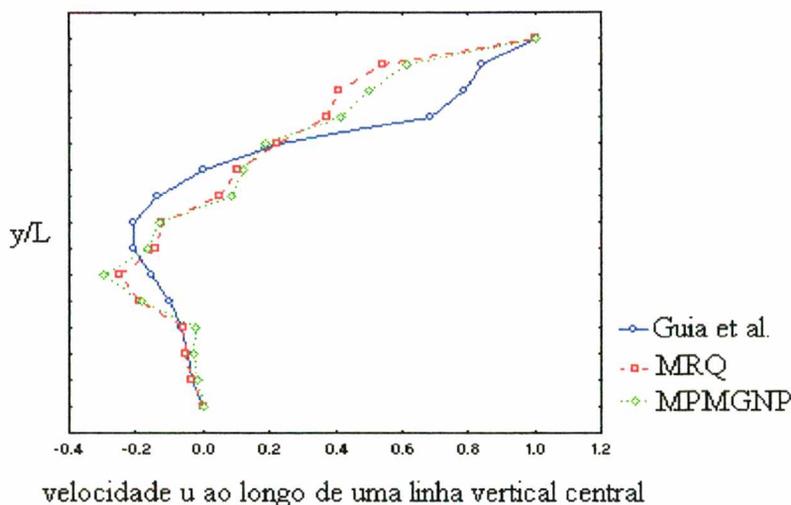


Figura 8.15 - Velocidade u com MRQ e MPMGNP na malha aleatória, (530 pontos nodais).

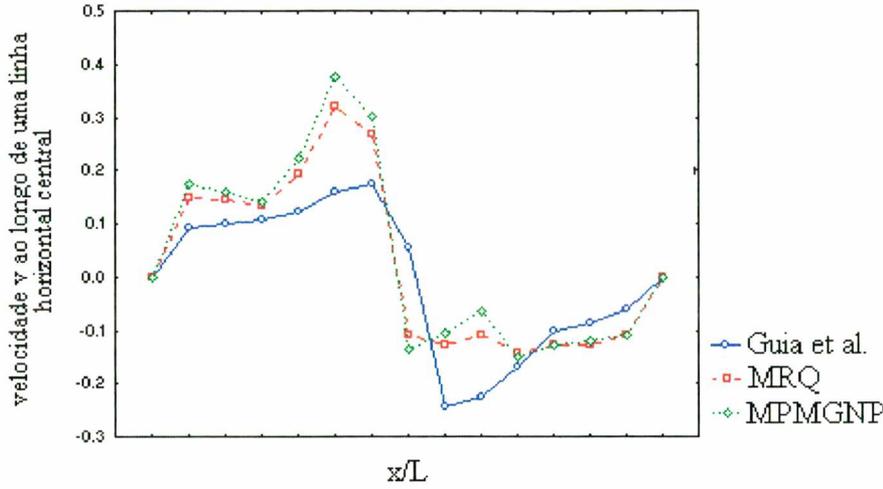


Figura 8.16 - Velocidade v com MRQ e MPMGNP na malha aleatória, (530 pontos nodais).

A figura 8.17 ilustra o comportamento das diferentes avaliações dos gradientes de pressão utilizados no cálculo da velocidade na direção x na malha modelada por volumes retangulares. A figura 8.18 mostra o comportamento da velocidade na direção y calculada com a aplicação dos diferentes gradientes de pressão, na mesma malha. Nestas figuras observa-se que ambas avaliações dos gradientes de pressão obtiveram comportamento semelhantes, porém na figura 8.18 o campo de velocidades na direção y , calculado com a MPMGNP se sobrepôs aos resultados obtidos por Ghia *et al.* (1982), em alguns pontos. Os valores apresentados nas figuras 8.17 e 8.18 foram obtidos com o método GC.

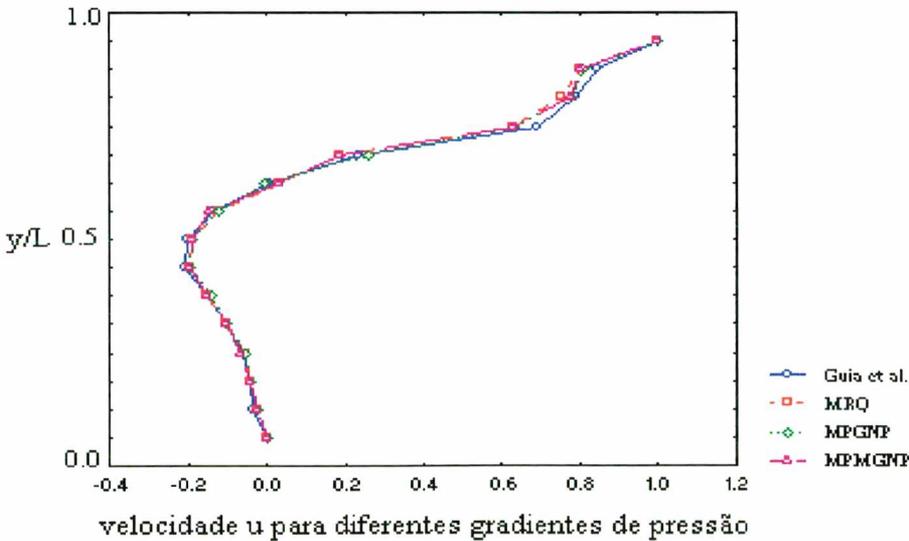


Figura 8.17 - Avaliação dos gradientes de pressão para a velocidade u com Reynolds = 100, malha retangular.

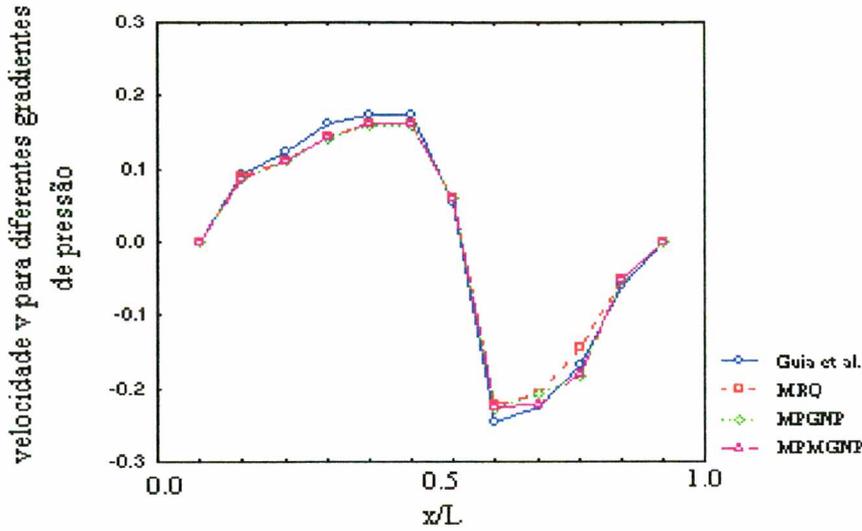


Figura 8.18 - Avaliação dos gradientes de pressão para a velocidade v com Reynolds = 100, malha retangular.

O comportamento das diferentes avaliações dos gradientes de pressão utilizados no cálculo da velocidade na direção x na malha modelada por volumes hexagonais é mostrado na figura 8.19. A figura 8.20 ilustra o comportamento da velocidade na direção y . Nas figuras 8.19 e 8.20 os valores foram obtidos utilizando o método CGS na resolução dos sistemas.

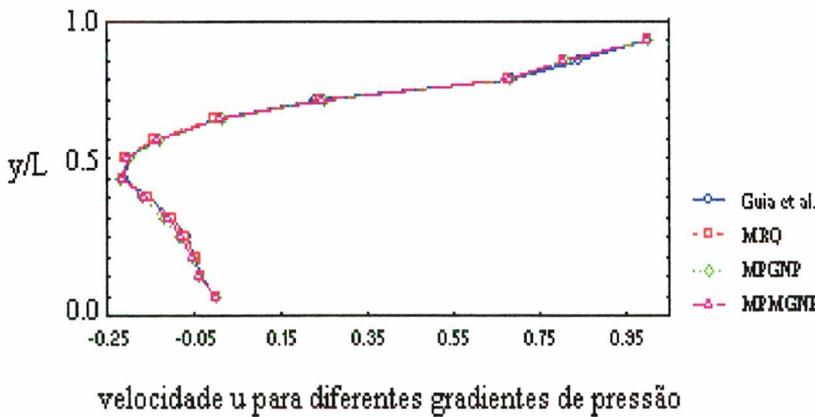


Figura 8.19 - Avaliação dos gradientes de pressão para a velocidade u com Reynolds = 100, malha hexagonal.

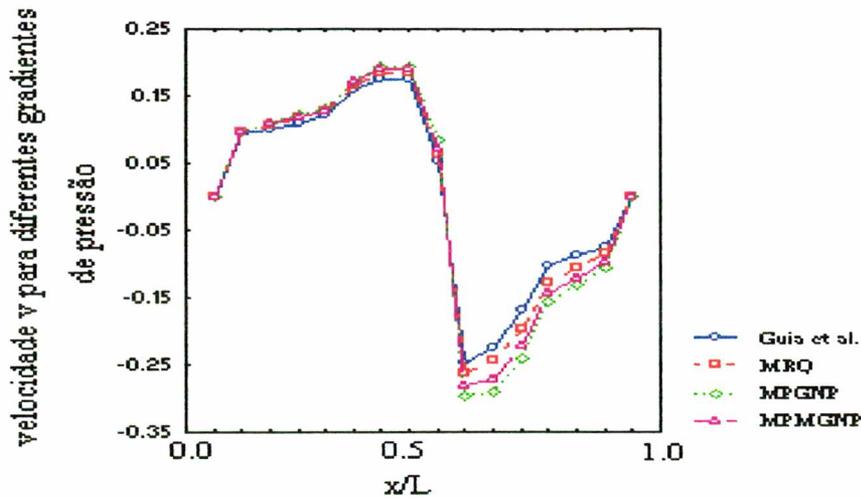


Figura 8.20 - Avaliação dos gradientes de pressão para a velocidade v com Reynolds = 100, malha hexagonal.

Na figura 8.19 nota-se que os valores encontrados com as diferentes propostas de cálculo para os gradientes de pressão se sobrepõem aos valores de Ghia *et al.*. No caso do cálculo da velocidade na direção y com o MRQ obteve-se os resultados mais adequados que os demais, e a MPGNP obteve os piores resultados. Vale ressaltar que nenhum deles obteve valores aproximadamente iguais aos de Ghia *et al.* em todos os pontos.

A tabela 8.2 expõe os resultados obtidos utilizando Reynolds = 400 para os diversos métodos iterativos propostos. Nos sistemas lineares, usa-se um laço iterativo para as variáveis u e v com 2 iterações internas para a correção da pressão com 3 iterações internas.

Tabela 8.2- Processo iterativo nas equações de Navier-Stokes com Reynolds = 400.

Métodos		CGS		Bi-CGSTAB		TFQMR		GC		
		iter	t	iter	t	iter	t	iter	t	
malhas	gradiantes									
	MRQ	r	nc	-	nc	-	nc	-	397	1,13
		h	nc	-	1158	12,42	nc	-	1593	10,50
a		nc	-	nc	-	nc	-	1012	1,31	
MPGNP	r	nc	-	nc	-	nc	-	395	1,13	
	h	nc	-	871	7,35	nc	-	3526	24,41	
	a	nc	-	nc	-	nc	-	nc	-	
MPMGNP	r	nc	-	nc	-	nc	-	408	1,31	
	h	nc	-	885	9,00	nc	-	2301	19,44	
	a	nc	-	nc	-	nc	-	nc	-	

Na tabela 8.2 o GC é o método mais eficiente para a malha retangular sendo que os demais métodos não convergiram. Na malha hexagonal o método mais eficiente é o Bi-CGSTAB

utilizando menor tempo de processamento e número de iterações para convergir. Os métodos que não convergiram talvez possam vir a convergir com algum pré-condicionamento, o que pode vir a ser feito em trabalhos futuros.

O comportamento dos diferentes gradientes de pressão utilizados no cálculo da velocidade u com Reynolds = 400, na malha modelada por volumes retangulares é apresentado na figura 8.21. A figura 8.22 ilustra o comportamento da velocidade v calculada com a aplicação dos diferentes gradientes de pressão. Nas figuras 8.23 e 8.24 apresenta-se o comportamento dos gradientes para a malha hexagonal.

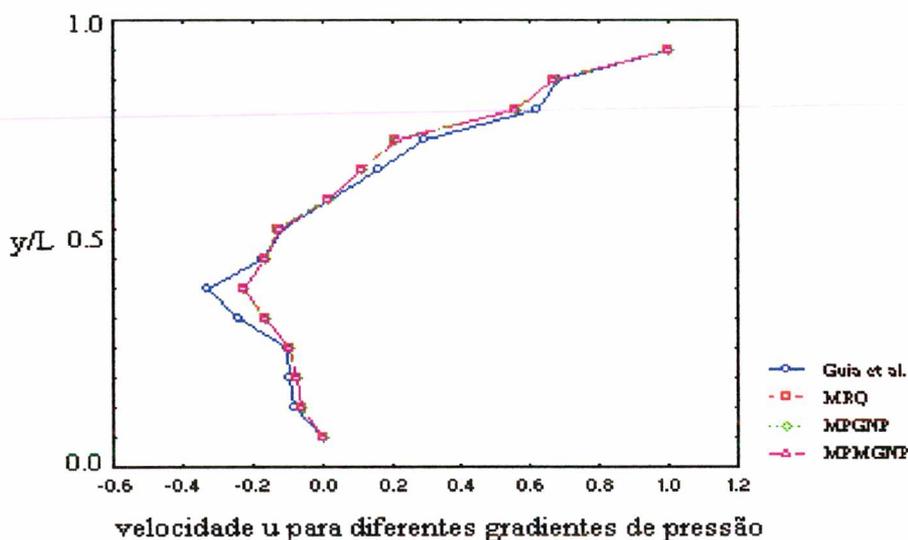


Figura 8.21 - Avaliação dos gradientes de pressão para a velocidade u com Reynolds = 400, malha retangular.

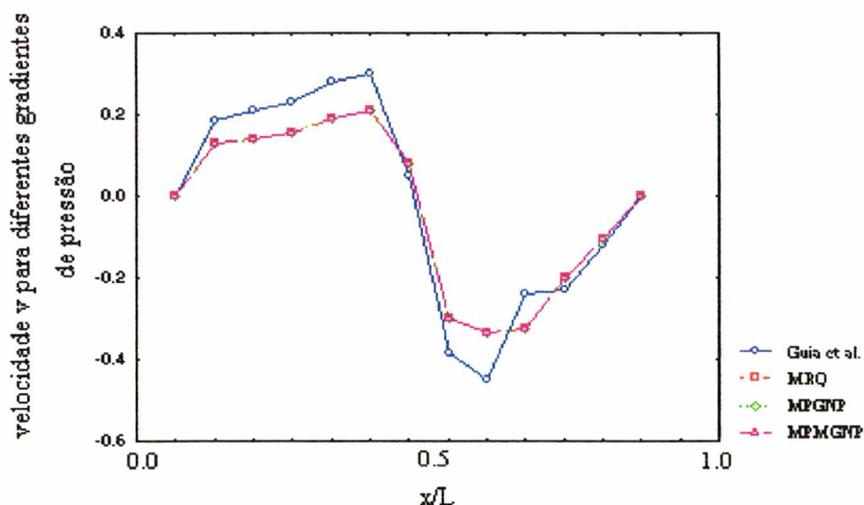


Figura 8.22 - Avaliação dos gradientes de pressão para a velocidade v com Reynolds = 400, malha retangular.

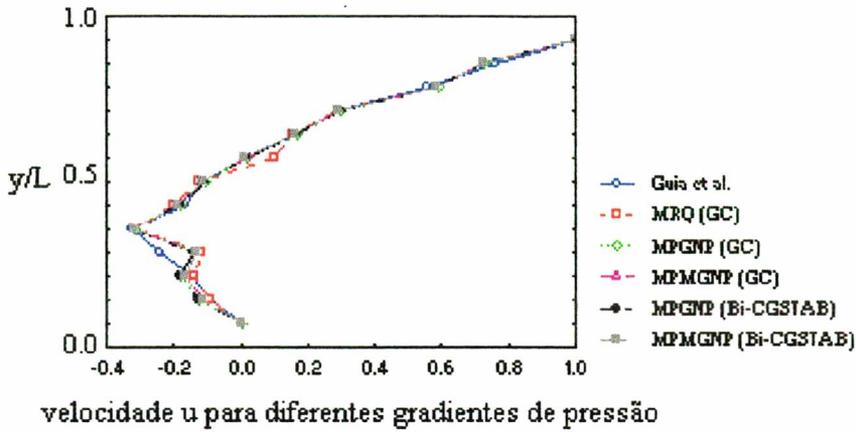


Figura 8.23 - Avaliação dos gradientes de pressão para a velocidade u com Reynolds = 400, malha hexagonal.

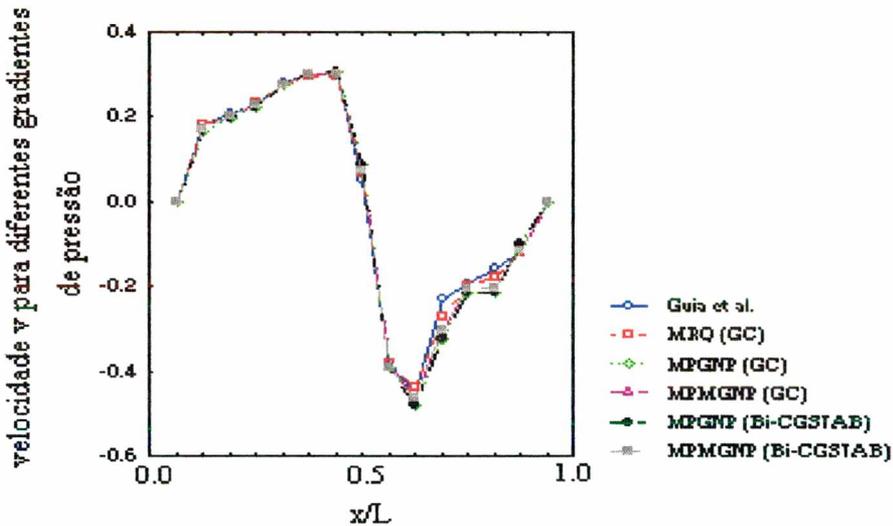


Figura 8.24 - Avaliação dos gradientes de pressão para a velocidade v com Reynolds = 400, malha hexagonal.

Nas figuras 8.21 e 8.22 os resultados são obtidos com o GC e nota-se que os valores encontrados tanto para a velocidade u como para a velocidade v , estão ligeiramente distantes dos valores encontrados por Ghia *et al.*, sendo que o MRQ, a MPGNP e a MPMGNP obtiveram os mesmos resultados entre si, isto é suas curvas se sobrepõem. Atribui-se estas diferenças às malhas não suficientemente refinadas.

Nas figuras 8.23 e 8.24 os valores obtidos, utilizando a malha hexagonal, estão mais próximos dos valores de Ghia *et al.*, apenas a velocidade u tem pequenas distorções. Para a velocidade u , a MPMGNP avalia resultados que se aproximam mais dos valores de Ghia *et al.* enquanto que com a MPGNP (GC) obteve-se os valores mais distantes de Ghia *et al.*, o MRQ

(GC) é o método mais adequado para a velocidade v enquanto a MPGNP (Bi-CGSTAB) é o menos adequado.

A tabela 8.3 apresenta os valores para Reynolds = 1000. Nos sistemas lineares usa-se um laço iterativo interno para o cálculo das variáveis u e v , com 2 iterações e para a correção da pressão com 3 iterações

Tabela 8.3 - Processo iterativo nas equações de Navier-Stokes com Reynolds = 1000.

Métodos		CGS		Bi-CGSTAB		TFQMR		GC	
malhas		iter	t	iter	t	iter	t	iter	t
gradientes									
MRQ	r	nc	-	nc	-	nc	-	676	1,22
	h	nc	-	nc	-	nc	-	nc	-
	a	nc	-	nc	-	nc	-	nc	-
MPGNP	r	nc	-	nc	-	nc	-	631	1,43
	h	nc	-	nc	-	nc	-	nc	-
	a	nc	-	nc	-	nc	-	nc	-
MPMGNP	r	nc	-	nc	-	nc	-	nc	-
	h	nc	-	nc	-	nc	-	nc	--
	a	nc	-	nc	-	nc	-	nc	-

As diferentes avaliações utilizadas para o cálculo dos gradientes de pressão na direção x e na direção y com Reynolds = 1000 em malha retangular são apresentados, respectivamente, na figura 8.25.

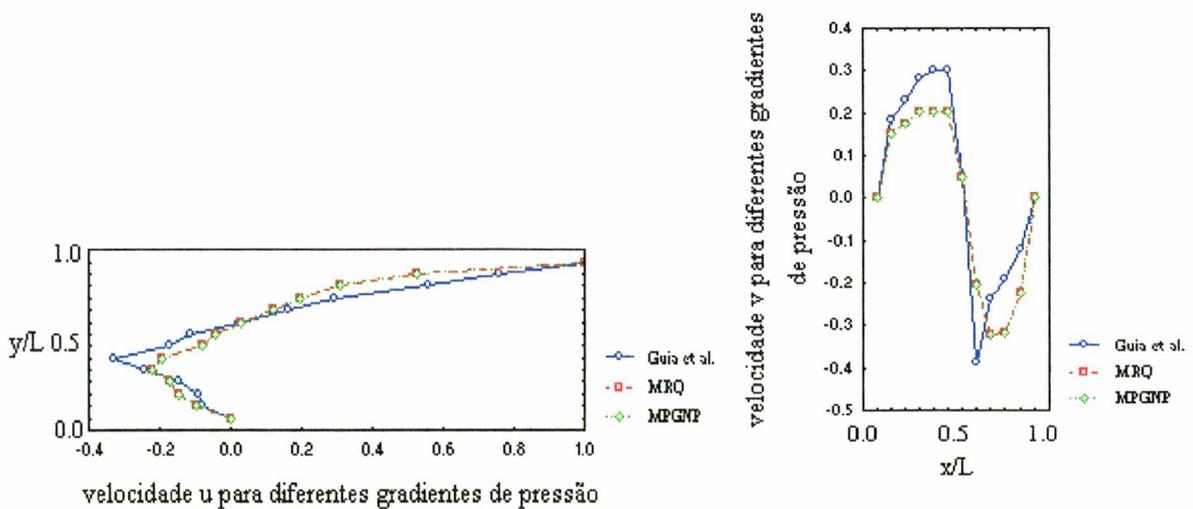


Figura 8.25 - Avaliação dos gradientes de pressão para a velocidade u e v com Reynolds = 1000.

Observa-se nas figuras anteriores que os resultados obtidos com Reynolds = 1000 estão ligeiramente afastados da curva das distribuições do campo de velocidades u e v apresentados por Ghia *et al.*, 1982.

8.3 Análise da correção da pressão

Para os procedimentos de cálculo da correção de pressão equação (5.51), se faz necessário um número maior de iterações internas na resolução do sistema o que, geralmente, resulta em uma solução mais rápida e estável dos métodos utilizados (Patankar, 1980; Tanyi and Thatcher, 1996). Visando explorar esta análise, as tabelas 8.4 a 8.10 apresentam várias combinações variando os passos iterativos, para os sistemas de solução das equações u , v e p' com Reynolds=100.

Os sistemas de equações para as variáveis u , v e para a correção da pressão são originalmente não-lineares, porém estes sistemas são linearizados, e deste modo cada incógnita é resolvida através da seqüência iterativa apresentada no fluxograma da figura 3.4.

A tabela 8.4 apresenta o número de iterações encontrado com os métodos iterativos GC, CGS, Bi-CGSTAB e TFQMR para o cálculo dos gradientes de pressão avaliados pela proposta de Taniguchi (Taniguchi *et al.*, 1991; Taniguchi and Kobayashi, 1991) na cavidade quadrada modelada por volumes retangulares conforme figura (7.3), com MRQ

Tabela 8.4 - Performance (iteraões) do GC, CGS, Bi-CGSTAB e TFQMR com o MRQ, malha retangular.

<i>Mínimo resíduo quadrático</i>						
u	v	p'	GC	CGS	Bi-CGSTAB	TFQMR
1	1	1	762	1624	1166	2918
1	1	2	784	1010	1216	2611
1	1	3	343	997	1213	2321
2	2	3	352	1014	1211	2091
2	2	6	505	833	1211	nc
4	4	6	548	892	1210	nc

A tabela 8.5 expõe o número de iterações necessário para a convergência de acordo com o critério pré-estabelecido, utilizando os métodos GC, CGS, Bi-CGSTAB e TFQMR com o cálculo

dos gradientes de pressão obtidos pela MPGNP (Cardoso, 1997) e pela MPMGNP na cavidade quadrada modelada por volumes retangulares.

Tabela 8.5 - Performance (iterações) do GC, CGS, Bi-CGSTAB e TFQMR com a MPGNP e MPMGNP, malha retangular.

<i>Média ponderada entre os gradientes normais projetados</i>						
u	v	p'	GC	CGS	Bi-CGSTAB	TFQMR
1	1	1	918	1624	1166	2918
1	1	2	807	1010	1216	2632
1	1	3	333	997	1213	2321
2	2	3	340	1014	1211	2091
2	2	6	505	833	1211	nc
4	4	6	547	892	1210	nc
<i>Média ponderada modificada entre os gradientes normais projetados</i>						
u	v	p'	GC	CGS	Bi-CGSTAB	TFQMR
1	1	1	932	1624	1166	2918
1	1	2	806	1010	1216	2632
1	1	3	350	997	1213	2321
2	2	3	505	1014	1211	2091
2	2	6	271	833	1211	nc
4	4	6	522	892	1210	nc

Nas tabelas 8.4 e 8.5 observa-se que o número de iterações obtidas para os diferentes cálculos dos gradientes de pressão são os mesmos para o CGS e Bi-CGSTAB, e na tabela 8.5 a MPGNP e a MPMGNP necessitam do mesmo número de iterações. Esta conclusão também se verifica para o TFQMR, só o GC obtém resultados diferenciados, onde a MPGNP possui menor número de iterações que a MPMGNP até $u=v=2$ e $p=3$, para laços iterativos maiores a MPGNP possui menor número de iterações para convergir.

Na figura 8.26 verifica-se o comportamento iterativo do GC, aplicado na solução do sistema linear resultante da discretização das equações de Navier-Stokes em malha retangular, com laço iterativo formado por uma iteração para as velocidades (u , v) e para a correção da pressão, utilizando-se diferentes gradientes de pressão. Já na figura 8.27 apresenta-se o processo iterativo até a iteração 300.

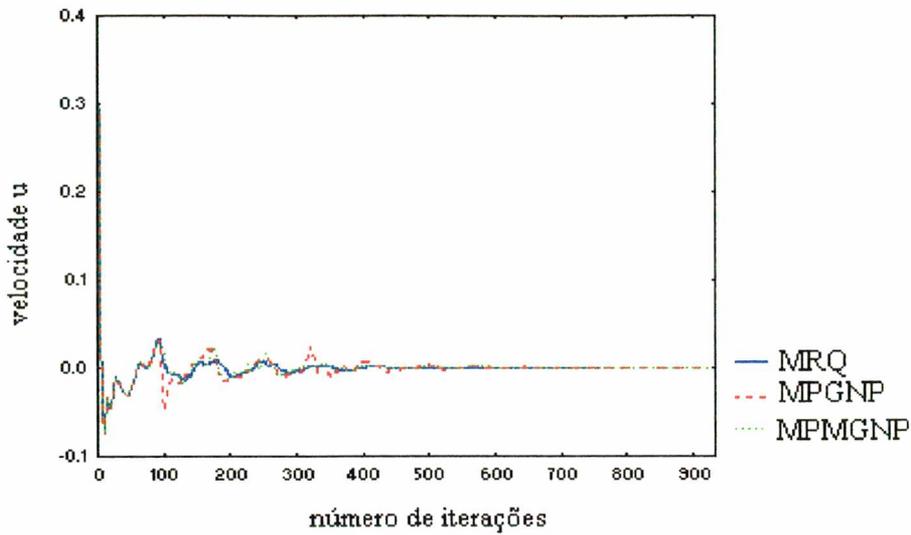


Figura 8.26 - Velocidade u obtida com o GC, utilizando MRQ, MPGNP e MPMGNP.

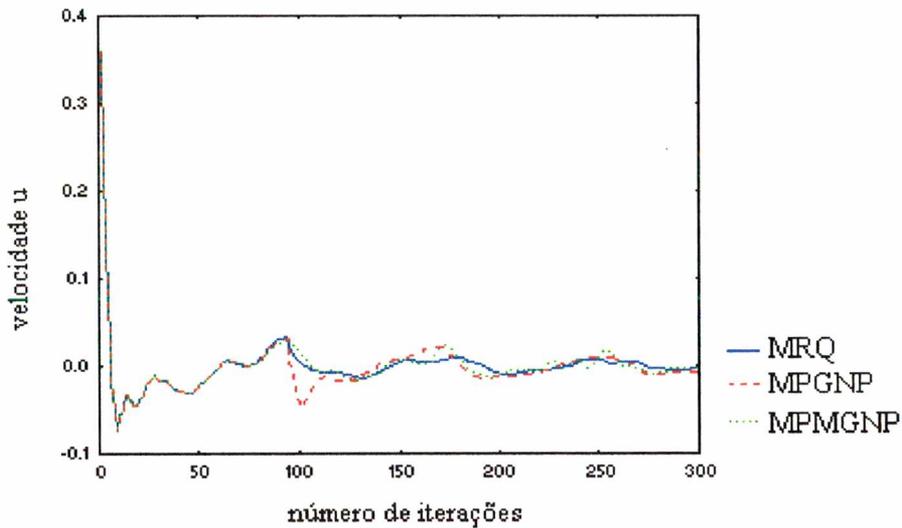


Figura 8.27 - Velocidade u até a iteração 300, com o MRQ, MPGNP e MPMGNP.

Observa-se nas figuras 8.26 e 8.27 que os gradientes de pressão MRQ e MPMGNP tiveram comportamento semelhantes, já o MPGNP é mais oscilatório que os demais, o que também pode ser observado nas figuras 8.28 e 8.29, as quais apresentam o comportamento do resíduo da equação da conservação da massa, ou erro durante o processo iterativo, para avaliação da convergência.

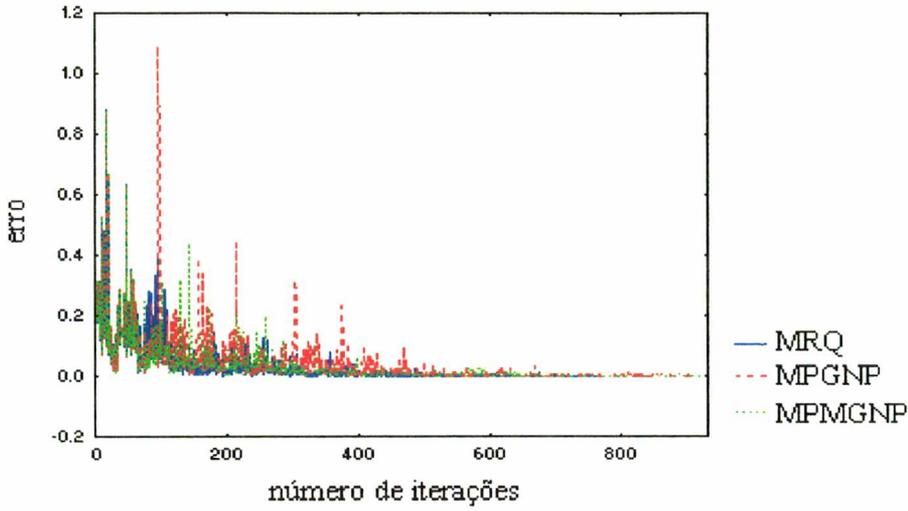


Figura 8.28 - Avaliação da convergência aplicando GC para o cálculo da velocidade u.

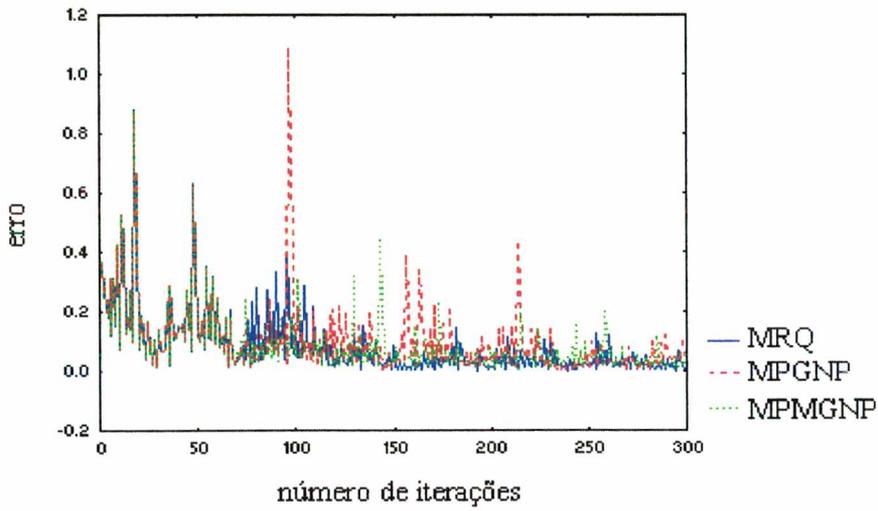


Figura 8.29 - Avaliação da convergência aplicando GC na velocidade u até a iteração 300.

A tabela 8.6 apresenta o processo iterativo dos métodos GC, CGS, Bi-CGSTAB e TFQMR utilizando o MRQ para calcular os gradientes de pressão, obtidos na cavidade quadrada modelada por volumes hexagonais.

Tabela 8.6 - Performance (iterações) do GC, CGS, Bi-CGSTAB e TFQMR com o MRQ, malha hexagonal.

<i>Mínimo resíduo quadrático</i>						
u	v	p'	GC	CGS	Bi-CGSTAB	TFQMR
1	1	1	804	2734	2197	4300
1	1	2	1879	2256	2093	3645
1	1	3	392	1971	2093	5948
2	2	3	614	1928	2197	6478
2	2	6	1117	1025	2089	nc
4	4	6	1355	1039	2089	nc

A tabela 8.7 expõe o processo iterativo dos métodos GC, CGS, Bi-CGSTAB e TFQMR utilizando a MPGNP para calcular os gradientes de pressão (Cardoso; 1997) na malha hexagonal.

Tabela 8.7 - Performance (iterações) do GC, CGS, Bi-CGSTAB e TFQMR com a MPGNP, malha hexagonal.

<i>Média ponderada entre os gradientes normais projetados</i>						
u	v	p'	GC	CGS	Bi-CGSTAB	TFQMR
1	1	1	nc	3104	2335	4845
1	1	2	1628	2523	2335	3474
1	1	3	487	2395	2335	6342
2	2	3	515	2260	2314	nc
2	2	6	1493	1143	2314	nc
4	4	6	1443	1148	2293	nc

A tabela 8.8 mostra os resultados obtidos com os métodos iterativos GC, CGS, Bi-CGSTAB e TFQMR utilizando a MPMGNP para o cálculo dos gradientes de pressão na malha hexagonal.

Tabela 8.8 - Performance (iterações) do GC, CGS, Bi-CGSTAB e TFQMR com a MPMGNP, malha hexagonal.

<i>Média ponderada modificada entre os gradientes normais projetados</i>						
u	v	p'	GC	CGS	Bi-CGSTAB	TFQMR
1	1	1	771	3204	2444	4620
1	1	2	732	2833	2444	4300
1	1	3	385	2535	2444	7437
2	2	3	402	2628	2406	nc
2	2	6	1601	1345	2406	nc
4	4	6	1422	1347	2384	nc

A tabela 8.9 apresenta o número de iterações encontrado com os métodos iterativos GC, CGS, Bi-CGSTAB e TFQMR utilizando o MRQ na cavidade quadrada modelada por volumes aleatórios, já a tabela 8.10 apresenta os resultados obtidos com a MPMGNP para os mesmos métodos iterativos citados anteriormente. A MPMGNP não tem os resultados apresentados pois não convergiu com nenhum dos métodos utilizados.

Tabela 8.9 - Performance (iterações) do GC, CGS, Bi-CGSTAB e TFQMR com o MRQ, malha aleatória.

<i>Mínimo resíduo quadrático</i>						
u	v	p'	GC	CGS	Bi-CGSTAB	TFQMR
1	1	1	661	4481	1560	6724
1	1	2	1354	2371	1560	5045
1	1	3	1495	2371	1560	25187
2	2	3	1497	nc	1059	nc
2	2	6	1685	nc	1059	nc
4	4	6	1674	nc	1174	nc

Tabela 8.10 - Performance (iterações) do GC, CGS, Bi-CGSTAB e TFQMR com a MPMGPN, malha aleatória.

<i>Média ponderada modificada entre os gradientes normais projetados</i>						
u	v	p'	GC	CGS	Bi-CGSTAB	TFQMR
1	1	1	1188	nc	2505	9971
1	1	2	2863	4813	2505	9947
1	1	3	3369	6142	2505	11343
2	2	3	3538	nc	2232	nc
2	2	6	2877	nc	2232	nc
4	4	6	3514	nc	2265	nc

8.4 Análise das diferentes formas de avaliação dos gradientes de pressão

Na tabela 8.11 apresenta-se os valores dos gradientes de pressão utilizando as diferentes formas de avaliação do gradiente para a geometria apresentada na figura (7.3). Escolhe-se dois pontos nodais um localizado na região central da cavidade quadrada, ponto (11,5; 10,4) e outro no ponto (15,8; 19,5) localizado próximo ao contorno da parede deslizante.

Tabela 8.11 - Avaliação dos gradientes de pressão na malha retangular (1369 pontos nodais).

Gradientes de Pressão	MRQ	MPGPN	MPMGPN
$\frac{\partial p}{\partial x}_{725}$	0,002906	0,002476	0,002914
$\frac{\partial p}{\partial y}_{725}$	-0,016825	-0,015636	-0,016243
$\frac{\partial p}{\partial x}_{1362}$	0,080579	0,080116	0,080379
$\frac{\partial p}{\partial y}_{1362}$	0,003113	0,003161	0,003123

Na tabela anterior observa-se que os valores encontrados com o MPMGPN são semelhantes aos valores obtidos com o MRQ, enquanto o MPGPN apresenta valores sensivelmente diferentes.

Na tabela 8.12 apresentam-se os valores dos gradientes de pressão utilizando as diferentes formas de avaliação do gradiente para a geometria apresentada na figura (6.7) discretizada com malha hexagonal, tomando-se dois pontos nodais, um localizado na região central da cavidade quadrada, ponto (10,6; 10,6) e outro no ponto (13,6; 19,8) localizado próximo ao contorno da parede deslizante.

Tabela 8.12 - Avaliação dos gradientes de pressão na malha hexagonal (2475 pontos nodais).

Gradientes de Pressão	MRQ	MPGNP	MPMGNP
$\frac{\partial p}{\partial x}_{1314}$	0,003364	-0,015103	0,002787
$\frac{\partial p}{\partial y}_{1314}$	-0,012030	-0,01300	-0,013937
$\frac{\partial p}{\partial x}_{2450}$	-0,010256	-0,009673	-0,011828
$\frac{\partial p}{\partial y}_{2450}$	0,006295	0,006532	0,006510

Na tabela 8.12, com malha hexagonal, os valores apresentam características distintas para cada um dos métodos de avaliação de gradientes, para o cálculo do gradiente de pressão na direção x o MRQ e a MPMGNP obtém resultados semelhantes, enquanto no cálculo do gradiente de pressão na direção y os três métodos apresentam resultados semelhantes. Nesta tabela, observa-se, também, que o gradiente de pressão na direção x calculado com a MPGNP, no ponto localizado no centro da cavidade quadrada, obteve resultado diferente do encontrado com as outras formas de avaliação do gradiente.

Na tabela 8.13 apresenta-se os valores dos gradientes de pressão utilizando as diferentes formas de avaliação do gradiente, para a geometria apresentada na figura (8.2) tomando-se dois pontos nodais um localizado próximo ao centro da cavidade quadrada (10,00; 10,00) e outro no ponto (18,5; 19,3) localizado no contorno da parede deslizante. Nota-se, nesta tabela, que os valores obtidos com cada forma de avaliação são semelhantes.

Tabela 8.13 - Avaliação dos gradientes de pressão na malha aleatória (530 pontos nodais).

Gradientes de Pressão	MRQ	MPGNP	MPMGNP
$\frac{\partial p}{\partial x}_{511}$	0,008548	-	0,011881
$\frac{\partial p}{\partial y}_{511}$	-0,002521	-	-0,003132
$\frac{\partial p}{\partial x}_{238}$	0,037540	-	0,039696
$\frac{\partial p}{\partial y}_{238}$	-0,015307	-	-0,019583

8.5 Conclusão

Este capítulo apresentou o problema da cavidade quadrada com profundidade infinita, sujeita a uma parede deslizante, modelando o escoamento do fluido regido pelas equações de Navier-Stokes. As malhas avaliadas foram: hexagonal (2475 pontos nodais), retangular (1369 pontos nodais) e aleatória (530 pontos nodais). As malhas formadas por volumes hexagonais e

retangulares mostraram-se mais adequadas que as aleatórias, pois aproximaram melhor os resultados encontrados por Ghia *et al.* (1982), além do que os métodos iterativos GC, CGS, Bi-CGSTAB e TFQMR, em geral, demoraram mais para convergir na malha aleatória.

Os métodos iterativos utilizados para solução das equações das velocidades u e v , e correção da pressão (presente no fluxograma da figura 3.4) foram o GC, CGS, Bi-CGSTAB e TFQMR. O GC apresentou desempenho superior, atingindo a convergência, geralmente, para todos os valores de Reynolds adotados. O Bi-CGSTAB convergiu para Reynolds = 100, mais rápido que os métodos CGS e TFQMR, sendo este último inferior aos demais, no presente trabalho.

Nos resultados coletados para Reynolds = 400, o GC foi o método mais estável, convergindo com quase todas as avaliações dos gradientes de pressão, e o Bi-CGSTAB em malha hexagonal, também convergiu. Com Reynolds 1000 os métodos, em geral, não convergiram, necessitando de uma análise posterior mais cuidadosa.

No cálculo dos gradientes de pressão nas direções x e y , três avaliações foram utilizadas: o MRQ, a MPGNP e a MPMGNP. Com Reynolds = 100, o MRQ foi eficaz para todas as malhas, a MPGNP com performance superior a MPMGNP, quanto ao tempo de processamento e número de iterações, sendo a última superior na malha hexagonal com os métodos GC e TFQMR.

Cabe ressaltar que a MPMGNP não convergiu na malha aleatória, com nenhum método iterativo adotado. Deve-se fazer alguns testes com outras malhas aleatórias para se ter um resultado mais conclusivo.

Alguns resultados foram coletados, combinando variações no número de iterações internas, para resolução dos sistemas das equações u , v e correção da pressão. Verificou-se, que após determinado número de iterações os resíduos das equações voltam a aumentar, aumentando o número de iterações para convergir, isto ocorre devido ao acoplamento e não-linearidade das equações governantes (Tanyi and Thatcher, 1996). Contudo, é difícil prever o comportamento da convergência e determinar o número de iterações internas exato ou padrão, com o objetivo de encontrar a convergência ótima.

Capítulo 9 - Conclusões e perspectivas para futuros trabalhos

No presente trabalho implementou-se um código computacional empregando malhas não-estruturadas geradas por diagramas de Voronoi, para a discretização da equação da condução de calor e equações de Navier-Stokes. O diagrama de Voronoi mostrou-se adequado para geração das malhas, captando qualquer contorno geométrico e permitindo o refino em regiões específicas do domínio de cálculo. Adicionalmente, as propriedades do diagrama de Voronoi facilitam a discretização de qualquer tipo de equação, onde haja necessidade de cálculo de um fluxo atravessando uma superfície.

As malhas de Voronoi aliam a versatilidade das malhas não-estruturadas com a simplicidade das malhas ortogonais, característica esta, que por sua vez, simplifica as aproximações numéricas existentes, facilitando a realização de balanços de massa, energia, cálculo de fluxos no volume de controle, entre outros.

No sistema de equações lineares, resultante da discretização da equação da condução de calor em malhas de Voronoi, comparou-se a eficiência de três tipos de ordenações utilizando os métodos GC, GS e TDMA, e três tipos de pré-condicionadores: diagonal, SSOR e por fatoração de Cholesky incompleta, utilizando o GC.

Nos sistemas de equações resultantes da discretização das equações de Navier-Stokes em malhas de Voronoi, comparou-se a performance de quatro métodos iterativos não-estacionários da família do Gradiente Conjugado: GC, CGS, Bi-CGSTAB e TFQMR. Os métodos foram avaliados utilizando Reynolds com valores 100, 400 e 1000 e adotando-se diferentes formas de avaliações dos gradientes de pressão nas direções x e y .

1. Análise da eficiência de três tipos de ordenações das incógnitas aplicadas com o GC: ordenação natural (horizontal e diagonal) e a ordenação geométrica.

A maior parte das técnicas de ordenação estão baseadas na estrutura da matriz, e não usam os valores numéricos de seus coeficientes. Existem trabalhos na literatura que empregam variadas formas de ordenação, algumas estão baseadas no grafo da matriz, reduzindo o sistema ou os “fill-ins”, assim como existem consideráveis trabalhos sobre o efeito da ordenação em problemas envolvendo malhas não-estruturadas (Maliska Jr. e Bezerra, 1994; Marcondes, 1996, Cardoso *et al.*, 1996).

No presente trabalho foram utilizadas duas malhas para comparação do efeito das ordenações empregadas. Na malha hexagonal (2475 pontos nodais) aplicou-se dois tipos de ordenações das variáveis: a ordenação geométrica em linha contínua formando um zigue-zague, proposta por Cardoso *et al.* (Cardoso *et al.*, 1996) e a ordenação natural. A ordenação natural é oriunda do gerador da malha não-estruturada, onde a numeração dos volumes de controle é feita através de linhas traçadas no sentido horizontal, usando a cota igual a y mínimo. Na malha retangular, a numeração dos volumes de controle é efetuada em linhas diagonais, a partir da origem da cavidade quadrada.

O método GC resolveu os três sistemas com melhor performance que o TDMA e GS. O GS apresentou baixo custo por iteração, o que é perdido pela velocidade de convergência extremamente lenta. As ordenações não obtiveram os resultados esperados, talvez seja justificável pelas malhas utilizadas serem pouco refinadas. Este é um tópico em aberto na literatura e para conclusões mais precisas pode-se empregar malhas aleatórios e geometrias diversificadas, com maior número de pontos nodais. Com o objetivo de melhorar a convergência do GC utilizou-se alguns pré-condicionadores, cujos resultados são apresentados a seguir.

2. Comportamento dos aceleradores de convergência, empregando três pré-condicionadores encontrados na literatura: diagonal, SSOR e por fatoração de CI.

Na análise da eficiência dos pré-condicionamentos foram usadas três malhas com geometrias variadas: formato L (malha aleatória, formada por 1003 pontos nodais), formato triangular (malha hexagonal, formada por 976 pontos nodais) e a cavidade quadrada (malha retangular, formada por 1369 pontos nodais).

A técnica mais simples de pré-condicionamento é a diagonal, removendo os efeitos das grandes variações, em magnitude, dos elementos diagonal, que em matrizes diagonalmente dominante pode causar um número de condição alto. O pré-condicionamento diagonal teve um desempenho excelente na resolução dos sistemas deste trabalho, isto deve-se ao fato que a matriz de coeficientes é diagonalmente dominante, e o número de condição também é relativamente pequeno.

Observou-se que o pré-condicionamento SSOR, usando $\omega = 1,5$ e $\psi = 1$ encontrou resultados semelhante aos obtidos com o pré-condicionamento diagonal, com $\psi = 0$ o desempenho foi igual ao método SSOR. O pré-condicionamento SSOR reduziu os autovalores maiores quase para a unidade, mas comumente permaneceram alguns autovalores pequenos,

causando problemas de convergência. Nos pré-condicionadores SSOR e diagonal não são requeridas armazenagem e tempo extra, para construção dos pré-condicionadores, contudo, geralmente, o pré-condicionamento SSOR torna-se mais lento que o diagonal.

No GCPCI sem “fill-ins” o tempo de processamento foi maior que o encontrado com os demais pré-condicionadores, e o número de iterações razoável, porém a fatoração pode falhar se a matriz tiver uma diagonal imaginária, assim algumas modificações podem ser feitas na matriz de coeficientes, evitando falhas. Um desempenho superior é esperado no pré-condicionamento por fatoração de Cholesky incompleta, quando aplicado em malhas mais refinadas, e quando utilizados níveis de enchimento entre 1 e 3 (Marcondes, 1996).

Em todas as figuras obtidas, utilizando o GC com os pré-condicionadores diagonal e SSOR os resultados foram semelhantes, a fatoração de CI teve desempenho um pouco superior. O resíduo pré-condicionado foi bastante oscilatório para este último, se comparado com os demais pré-condicionadores.

3. Comparação da eficiência dos métodos GC, CGS, Bi-CGSTAB e TFQMR, aplicados no problema clássico na área de CFD: o escoamento em uma cavidade quadrada com profundidade infinita, sujeita a uma parede superior deslizante.

Os resultados apresentados na resolução dos sistemas lineares em termos de tempo de CPU e número de iterações no “solver” mostraram que o GC é ligeiramente superior aos demais métodos, indo de acordo com as observações de Tanyi e Thatcher (Tanyi and Thatcher, 1996). Para Reynolds = 100 o método Bi-CGSTAB mostrou-se superior ao CGS e TFQMR em todas as malhas adotadas, utilizando 1 iteração interna na resolução das equações para a velocidade u , v e correção da pressão. Com o aumento de iterações, no laço interno dos sistemas, o método CGS convergiu com menor número de iterações que o Bi-CGSTAB e TFQMR. O TFQMR teve desempenho inferior aos demais métodos, porém alguns trabalhos mostram que ele é eficiente em computadores com processamento paralelo (Basermann and Bücker, 1994) e se usado em conjunção com pré-condicionadores (Knoll and McHugh, 1994).

Nos resultados com Reynolds = 400, o GC foi o método mais estável, convergindo com quase todas as avaliações dos gradientes de pressão, e o Bi-CGSTAB em malha hexagonal também convergiu. Com Reynolds = 1000 os métodos, em geral, não convergiram.

Nos procedimentos de cálculo da correção de pressão, se faz necessário um número maior de iterações internas na resolução dos sistemas de equações, buscando-se uma solução mais

rápida e estável dos métodos utilizados. Explorando esta análise, foram apresentadas várias combinações, variando os passos iterativos internos nos sistemas das equações u , v e correção da pressão, com Reynolds = 100.

Com as variações, observou-se que o GC é o método mais eficaz nas malhas modeladas por volumes retangulares e aleatórios, do que os demais métodos. Na malha modelada por volumes hexagonais, o GC possui performance superior até determinado número de iterações internas (geralmente $u = v = 2$ e $p' = 3$), após este número de iterações internas o CGS é superior. Entretanto é difícil determinar o número de iterações internas exato, para encontrar a convergência ótima, este valor está relacionado ao tamanho da malha, ao domínio geométrico e a malha computacional utilizada.

4. Comparação da eficiência das avaliações utilizadas para o cálculo dos gradientes de pressão nas direções x e y : MRQ, MPGNP e MPMGNP.

Nos resultados obtidos com Reynolds = 100, o MRQ proposto por Taniguchi *et al.* (1991) foi eficaz para todas as malhas, a MPGNP superior a MPMGNP, quanto ao tempo de processamento e número de iterações, sendo a última superior na malha hexagonal com os métodos GC e TFQMR. A MPGNP não convergiu com a malha aleatória, com nenhum dos métodos adotados, deve-se fazer outros testes para uma análise mais precisa.

O cálculo dos gradientes de pressão pelo MRQ, considera todas as influências das pressões dos pontos nodais vizinhos em relação ao ponto nodal central buscando as diferenças mínimas, fato pelo qual encontra os melhores resultados.

Perspectivas Futuras

Com o programa computacional desenvolvido surgem várias perspectivas de trabalhos e projetos a serem desenvolvidos, entre eles:

- Resolver os sistemas de equações não-lineares por métodos que usam homotopias e continuação, assim como utilizar a expansão em Série de Taylor (Newton) para a linearização das equações algébricas geradas pela discretização numérica;
- Resolver simultaneamente os três sistemas de equações algébricas gerados para as velocidades u , v e pressão;

- Estudar novos métodos para o cálculo dos gradientes de pressão e analisar as avaliações, utilizadas no presente trabalho com malhas mais refinadas, objetivando conclusões mais precisas;
- Estudar novas ordenações nas malhas geométricas a fim de minimizar a esparsidade da matriz de coeficientes, tornando próxima de uma matriz banda, tentando diminuir a condição espectral;
- Investigar os métodos da família do GC aplicando o pré-condicionador por fatoração de Cholesky incompleta, com vários níveis de enchimento, na solução dos sistemas lineares;
- Utilizar os pré-condicionadores por série truncada de Neumann e Maclaurin em computadores paralelos de alto desempenho, para a solução dos sistemas resultantes da discretização das equações de Navier-Stokes em malhas de Voronoi. Estes pré-condicionadores por séries devem ser eficientes, pois com matrizes diagonalmente dominante tem-se séries convergentes;
- Paralelizar os códigos computacionais objetivando uma melhora na performance, porém as vezes é muito complicado de se usar.

Referências Bibliográficas

1. Ajiz, M. A. and Jennings, A., 1984, "A Robust Incomplete Choleski-Conjugate Gradient Algorithm", *International J. for Num. Meth. in Eng.*, Vol. 20, pp. 949-966.
2. Angeleri, F., Sonnad, V. and Bathe, K., 1989, "Studies of Finite Element Procedures - An Evaluation of Preconditioned Iterative Solvers", *Computers & Structures*, Vol. 32, n° 3/4, p. 671-677.
3. Aurenhammer, F., 1991, "Voronoi Diagrams - A Survey of a Fundamental Geometric Data Structure", *ACM Computing Surveys*, Vol. 23, n° 3.
4. Axelsson, O., 1985, "Incomplete Block Matrix Factorization preconditioning Methods. The Ultimate Answer?", *Journal of Computational and Applied Mathematics*, 12&13, p. 3-18.
5. Azevedo, J. L. L., 1996, "Um Estudo de Simulação Dinâmica em Sistemas Elétricos Usando Computadores Paralelos de Alta Performance". Dissertação de Mestrado, Universidade Federal de Santa Catarina, Florianópolis.
6. Barrett, R., Berry, M., Chan, T. F., Demmel, J., Donato, J., M., Dongarra, J., Eijkhout, V., Pozo, R., Romine, C., and Vorst, H. V., 1994, "Templates for Solution of Linear Systems: Building Blocks for Iterative Methods", <http://www.netlib.org/templates/Templates.html>.
7. Behr, M. and Tezduyar, T. E., 1994, "Finite Element Solution Strategies for Large-Scale Flow Simulations", *Computer Methods in Applied Mechanics and Engineering*, 112, pp. 3-24.
8. Borges, C. L. T., Coutinho, A. L. G. A., Falcão, D. M., 1996, "Solução do Fluxo de Potência em Ambiente Vetorial Usando o Método do Gradiente Bi-Conjugado Estabilizado", *XI Congresso Brasileiro de Automática*, São Paulo, p. 161-166.
9. Brussino, G. and Sonnad, V., 1989, "A Comparison of Direct and Preconditioned Iterative Techniques for Sparse, Unsymmetric Systems of Linear Equations", *International Journal for Numerical Methods in Engineering*, Vol. 28, p. 801-815.
10. Bücker, M. and Basermann, A., 1994, "A Comparison of QMR, CGS and TFQMR on a Distributed Memory Machine", Interner Bericht, KFA-ZAM-IB-9412.
11. Campos, F. F. and Rollett, J. S., 1995, "Analysis of Preconditioners for Conjugate Gradients Through Distribution of Eigenvalues", *International J. Computer Math.*, Vol. 58, p. 135-158.
12. Campos, F. F., 1995, "Analysis of Conjugate Gradients - Type Methods for Solving Linear Equations", D Phil. Thesis, Oxford University Computing Laboratory, Oxford, UK.
13. Canny, J. and Donald, B., 1988, "Simplified Voronoi Diagrams", *Discrete Comput. Geom.*, Vol. 3, p. 219-236.

14. Cardoso, F. C., 1997, "Algoritmo para Simulação Numérica das Equações do Movimento pelo Método dos Volumes Finitos Usando Diagramas de Voronoi", Dissertação de Mestrado, Universidade Federal de Santa Catarina, Florianópolis.
15. Cardoso, F. C., Mariani, V. C. e Peters, S., 1996, "Linha de Ordenação Contínua em Malha Não-Estruturada Gerada por Diagrama de Voronoi", *XIX Congresso Nacional de Matemática Aplicada e Computacional*, p. 344-346, Goiânia.
16. Christofides, N., 1975, "Graph Theory, an Algorithmic Approach", Academic Press, London.
17. Coutinho, A. L. G. A., Alves, J. L. D., Landau, L., Ebecken, N. F. F. and Troina, L. M., 1991, "Comparison of Lanczos and Conjugate Gradients for the Element-by-Element Solution of Finite Element Equations on the IBM 3090 Vector Computer", *Computers & Structures*, Vol. 39, n° 1/2, p. 47-55.
18. D'Azevedo, E. F., Forsyth, P. A. and Tang, W., 1992, "Ordering Methods for Preconditioned Conjugate Gradient Methods Applied to Unstructured Grid Problems", *SIAM J. Matrix Anal. Appl.*, Vol. 13, n° 3, p. 944-961.
19. De Bortoli, A. L., 1996, "Aplicações de Métodos Numéricos na Solução de Problemas de Escoamento", EIMAC'96, Coletânea das Notas de Aula dos Mini-Cursos, W. L. R. (ed.), Porto Alegre.
20. Deng, G. B., Piquet, J., Queutey, P and Vinsonneau, M., 1994, "A New Fully Coupled Solution of the Navier-Stokes Equations". *International Journal Numerical Methods in Fluids*, Vol. 19, pp. 605-639.
21. Dracopoulos, M. C. and Crisfield, M. A., 1995, "Coarse/Fine Mesh Preconditioners for the Iterative Solution of Finite Element Problems", *International Journal for Numerical Methods in Engineering*, Vol. 38, pp.3297-3313.
22. Fletcher, R., 1976, "Conjugate Gradient Methods for Indefinite Systems", *Numerical Analysis Dundee 1975*, pp. 73-89, ed. G. Watson, Springer-Verlag, New York.
23. Fortune, S., 1987, "A Sweep-line Algorithm for Voronoi Diagrams", *Algorithmica*, n° 2, p. 153-174.
24. Freund, R. W., 1993, "A Transpose-Free Quasi-Minimal residual Algorithm for Non-Hermitian Linear Systems", *SIAM J. Sci. Comput*, Vol. 14, n° 2, pp. 470-482.
25. Fujino, S. and Zhang, S., 1992, "Analysis on Convergence Behavior of the CGS and BI-CGSTAB Methods", *Computer Arithmetic and Enclosure Methods*, L. Atanassova and J. Herzberger (Editors), Elsevier Science Publishers B. V.
26. Ghia, U., Guia, K. N. and Shin, C. T., "High-Re Solutions for Incompressible Flow Using the Navier-Stokes Equations and Multigrid Method", *Journal of Computational Physics*, Vol. 48, pp. 387-411, 1982.

27. Golub, G. H. and O'Leary, D. P., 1989, "Some History of the Conjugate Gradient and Lanczos Algorithms: 1948-1976", *SIAM Review*, Vol. 31, n° 1, p. 50-102.
28. Golub, G. H., and Van Loan, C. F., 1996, "Matrix Computations" . The Johns Hopkins University Press, 3^a. edition.
29. Hageman, L. A. and Young, D. M., 1981, "Applied Iterative Methods", Academic N. Y..
30. Hribersek, M. and Skerget, L., 1996, "Iterative Methods in Solving Navier-Stokes Equations by the Boundary Element Method", *International Journal for Numerical Methods in Engineering*, Vol. 39, pp. 115-139.
31. Johnson, A. G., Micchelli, C. A. and Paul, G., 1983, "Polynomial Preconditioners for Conjugate Gradient Calculations", *SIAM J. Numer. Anal.*, Vol. 20, n° 2, pp. 362-367.
32. Jones, M. T. and Plassmann, P. E., 1995; "An Improved Incomplete Cholesky Factorization", *ACM Transactions on Mathematical Software*, Vol. 21, n° 1, março 1995, p. 5-17.
33. Kershaw, D. S., 1978, "The Incomplete Cholesky-Conjugate Gradient Method for the Iterative Solution of Systems of Linear Equations", *Journal of Comp. Physics*, Vol. 26, pp. 43-65.
34. Lin, H., Yang, D. Y. and Chieng, C., 1995, "Variants of Biconjugate Gradient Method for Compressible Navier-Stokes Solver", *AIAA Journal*, Vol. 33, n° 7, p. 1177-1184..
35. Liu, W. and Sherman, A. H., 1976, "Comparative Analysis of the Cuthill-McKee and the Reverse Cuthill-McKee Ordering Algorithms for Sparse Matrices", *SIAM J. Numer. Anal.*, Vol. 13, n° 2.
36. Maliska Jr., C. R. e Bezerra, L. H., 1994, "Resolução Numérica de Problemas de Transferência de Calor em Diagramas de Voronoi", *XVII Congresso Nacional de Matemática Aplicada e Computacional*, pp. 449-450, Vitória.
37. Maliska Jr., C. R., 1993, "Um robusto gerador de Diagramas de Voronoi para Discretização de Domínios Irregulares", *XIV Congresso Ibero Latino-Americano sobre Métodos Computacionais para a Engenharia*, p. 753-762, São Paulo.
38. Maliska, C. R., 1995 "Transferência de Calor e Mecânica dos Fluidos Computacional". Livros Técnicos e Científicos Editora, Rio de Janeiro.
39. Marcondes, F., 1996, "Solução Numérica Usando Métodos Adaptativos-Implícitos e Malhas de Voronoi de Problemas de Reservatório de Petróleo", Tese de Doutorado, Universidade Federal de Santa Catarina, Florianópolis, Brasil.
40. Marcondes, F., Zambaldi, M. C. and Maliska, C. R., 1995, "Comparação de Métodos Iterativos Não Estacionários em Malhas Não Estruturadas na Solução de Problemas de Petróleo", COBEM-CIDIM.

41. Mariani, V. C. e Peters, S., 1996, "Métodos para Solução de Sistemas Lineares Oriundos da Discretização em Malhas Não Estruturadas de Voronoi", *XIX Congresso Nacional de Matemática Aplicada e Computacional*, p. 355-356, Goiânia.
42. Mariani, V. C. e Peters, S., 1997, "Analysis of performance of iterative methods for solution of the heat conduction problem in Voronoi unstructured mesh", *III International Congress on Information Engineering*, pp.476-485, Buenos Aires.
43. McHugh, P. R. and Knoll, D. A., 1994, "Fully Coupled Finite Volume Solutions of the Incompressible Navier-Stokes and Energy Equations Using an Inexact Newton Method", *International J. for Numerical Methods in Fluids*, Vol. 19, p. 439-455.
44. O'Leary, D. P., 1995, "Conjugate Gradients and Related KMP Algorithms: The Beginnings", Technical report, CS-TR-3556, University of Maryland.
45. Paolini, G. V. and Brozolo, G. R., 1989, "Data Structures to Vectorize CG Algorithms for General Sparsity Patterns", *BIT*, Vol. 27.
46. Papadrakakis, M., 1993, "Solving Large-Scale Problems in Mechanics". John Wiley & Sons.
47. Papadrakakis, M. and Dracopoulos, M. C., 1991, "A Global Preconditioner for the Element-by-Element Solution Methods", *Computer Methods in Applied Mechanics and Engineering*, Vol. 88, pp. 275-286.
48. Patankar, S. V., 1980, "Numerical Heat Transfer and Fluid Flow", McGraw-Hill
49. Peters, S., 1996, "Resolução de Equações Diferenciais Parciais com Diagramas de Voronoi", EIMAC'96, Coletânea das Notas de Aula dos Mini-Cursos, W. L. R. (ed), Porto Alegre.
50. Peters, S., 1994, "Bifurcação e Oscilações Auto-Induzidas em escoamentos de Fluidos em Difusores Radiais", Tese de Doutorado, Universidade Federal de Santa Catarina, Florianópolis, Brasil.
51. Preparata, F. P. and Shamos, M., 1985, "Computational Geometry an Introduction", Spring Verlag.
52. Reid, J. K., 1971, "On the Method of Conjugate Gradients for the Solution of Large Sparse Systems of Linear Equations" *In Large Sparse Sets of Linear Equations*, p. 231-254, London, Proceedings of the Oxford conference of the Institute of Mathematics and Its Applications held in April, Academic Press.
53. Rich, E., 1988, "Inteligência Artificial", Editora McGraw-Hill, São Paulo.
54. Ronzani, E. R. e Nieckele, A. O., 1995, "Método de Solução Numérica de escoamentos Incompressíveis em Geometrias Complexas", Congresso Ibero Latino Americano sobre Métodos Computacionais para Engenharia, p. 41-50, Paraná.

55. Roos, T., 1991, "Dynamic Voronoi Diagrams", Würzburg.
56. Roque, W. L., 1996, "Introdução a Técnicas de Planejamento de Trajetórias de Robôs Móveis". *IV Escola Regional de Informática - SBC*.
57. Saad, Y. and Schultz, M. H., 1986, "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems", *SIAM J. Sci. Stat. Comput.*, Vol. 7, n° 34, p. 856-869.
58. Saint-Georges, P., Warzee, G., Beauwens, R., Notay, Y., 1996, "High-Performance PCG Solvers for FEM Structural Analysis", *International Journal for Numerical Methods in Engineering*, Vol. 39, pp. 1313-1340.
59. Santos, L. A. dos, 1996, "O Desacoplamento Par-Ímpar do Campo de Pressão e Algoritmos para Simulação de Escoamentos Incompressíveis por Volumes Finitos", Dissertação de Mestrado, Universidade Federal de Santa Catarina, Florianópolis.
60. Silva, A. F. C. da, 1991, "Um Procedimento em Volumes Finitos para a Solução de Escoamentos de Qualquer Velocidade", Tese de Doutorado, Universidade Federal de Santa Catarina, Florianópolis.
61. Simon, H. D., 1988, "Incomplete LU Preconditioners for Conjugate-Gradient-Type Iterative Methods", *SPE Reservoir Engineering*.
62. Sleijpen, G. L. G. and Fokkema, D. R., 1993, "Bi-CGSTAB(L) for Linear Equations Involving Unsymmetric Matrices with Complex Spectrum", *Electronic Transactions on Numerical Analysis*, Vol. 1, p. 11-32.
63. Sonneveld, P., 1989, "CGS, A Fast Lanczos-Type Solver for Nonsymmetric Linear Systems", *SIAM J. Sci. Stat. Comput.*, Vol. 10, n° 1, p. 36-52.
64. Sugihara, K. and Iri, M., 1992, "Construction of the Diagram for "One Million" Generators in Single-Precision Arithmetic", *Proceedings of the IEEE*, Vol. 80, n° 9.
65. Taniguchi, N. and Kobayashi, T., 1991, "Finite Volume Method on the Unstructured Grid System", *Computers & Fluids*, Vol. 19, n° 34, p. 287-295.
66. Taniguchi, N., Arakawa, C. and Kobayashi, T., 1991, "Construction of a Flow-Simulating Method With Finite Volume Based on a Voronoi Diagram", *JSME International Journal, Serie II*, Vol. 34, p. 18-23.
67. Tanyi, B. A. and Thatcher, R. W., 1996, "Iterative Solution of the Incompressible Navier-Stokes Equations on the Meiko Computing Surface", *International Journal for Numerical Methods in Fluids*, Vol 22, p. 225-240.
68. Van der Vorst, H. A., 1992, "Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems", *SIAM J. Sci. Stat. Comput.*, Vol. 13, n° 2, p. 631-644.

69. Venkatakrishnan, V., 1990, "Preconditioned Conjugate Gradient Methods for the Compressible Navier-Stokes Equations", *AIAA Journal*, Vol. 29, n° 7, p. 1092-1100.
70. Weatherill, N. P., 1992, "Delaunay Triangulation in Computational Fluid Dynamics", *Computers Math. Applic.*, Vol. 24, n° 5/6, p. 129-150.
71. Yang, D. Y., Heng, L., and Chieng, C., 1994, "Variants of Biconjugate Gradient Method for Compressible Navier-Stokes Solver", *AIAA Journal*. Vol. 33, n° 7, pp. 1177-1184.

Apêndice A - Glossário

As definições básicas usadas nos capítulos são apresentadas através de um breve resumo. Uma pesquisa mais completa pode ser realizada em (Patankar, 1980; Varga, 1962; Barrett *et al*, 1994; Maliska, 1995). Recomenda-se aos iniciantes nesta área fazer uma breve leitura deste glossário.

1 Mecânica dos Fluidos Computacional e Transferência de Calor

1.1 Volume de controle é uma região finita do domínio computacional associada a um ponto nodal.

1.2 Ponto nodal é um ponto do domínio computacional associado a uma variável dependente ϕ , que pretende-se calcular através da discretização numérica.

1.3 Discretização numérica é a divisão do domínio de interesse em volumes de controles, sendo o primeiro passo para obtenção das equações aproximadas.

1.4 O domínio geométrico de cálculo é dividido em um número finito de volumes de controle, não se sobrepondo, tal que existe um volume de controle para cada ponto nodal.

1.5 Na malha estruturada a discretização coincidente com a fronteira é obtida através de um sistema de coordenadas e cada volume de controle interno tem sempre o mesmo número de vizinhos, a *malha não-estruturada* tem facilidade para adaptatividade e o número de vizinhos varia de volume para volume.

1.6 Na malha co-localizada a incógnita é localizada (armazenada) no centro do volume de controle, enquanto na *malha desencontrada* a pressão é armazenada no centro e as velocidades u e v são armazenadas nas faces dos volumes de controle.

1.7 As equações diferenciais parciais têm por incógnitas funções de muitas variáveis, sobre as quais se efetuam as operações funcionais usuais e as operações de derivação parcial

1.8 A localização relativa das variáveis na malha computacional (co-localizada e desencontrada) é conhecido como *arranjo de variáveis*, seu papel principal é a posição relativa entre as componentes do vetor velocidade e a pressão.

1.9 Do ponto de vista numérico, o escoamento onde ρ (massa específica) tem variação considerável com p (pressão), é denominado *compressível*.

1.10 Difusão pura processo de transferência de calor associado apenas a interação molecular na ausência de convecção e radiação.

1.11 Degenerescência ocorre na geração de um volume de controle, onde a reta perpendicular que irá formar uma face do novo volume de controle passa muito próximo de um destes pontos nodais, desta forma a reta que liga o ponto nodal central ao seu vizinho passa por dentro de outro volume de controle.

1.12 Teorema da divergência - Supõe-se que o domínio limitado G , no espaço tridimensional,

tem um contorno irregular ∂G e que v é uma área vetorial plana definida em um conjunto aberto contendo G e ∂G . Então:

$$\iiint_G \nabla \bar{v} dV = \iint_{\partial G} v \cdot n dS$$

onde o contorno ∂G é orientado pelo n considerado como a normal exterior.

2 Álgebra Matricial

2.1 A é *matriz quadrada* se o número de linhas é igual ao número de colunas. Uma matriz quadrada $L_{n \times n}$ é dita *triangular inferior* se $l_{ij} = 0$, para $\forall i < j$, e a matriz quadrada $U_{n \times n}$ é dita *matriz triangular superior* se $u_{ij} = 0$, para $\forall i > j$.

2.2 A matriz A $n \times n$ é dita *diagonal estritamente dominante* quando,

$$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}|, \text{ para } \forall i = 1, 2, \dots, n.$$

2.3 A matriz A^{-1} é *inversa* de A , se e só se $A^{-1} \cdot A = A \cdot A^{-1} = I$, ou de maneira equivalente $\det(A) \neq 0$. A é denominada de singular ou não-inversível se não possui inversa.

2.4 A *matriz identidade* de ordem n , $I = (\delta_{ij})$ é uma matriz diagonal com elementos,

$$\delta_{ij} = \begin{cases} 1, & \text{se } i = j \\ 0, & \text{se } i \neq j \end{cases}, \text{ onde } \delta_{ij} \text{ é o delta de Kronecker.}$$

2.5 *Sistemas mal condicionados*, geralmente, são decorrentes de problema do modelo sendo sensíveis a pequenas perturbações em seus parâmetros.

2.6 Uma matriz quadrada A é *simétrica* se $A = A^T$. A é *positiva definida* se for simétrica e se $x^T A x > 0$ para $\forall x \neq 0$.

2.7 Uma matriz $n \times n$ é chamada *matriz banda* se $\exists p, q$ inteir, $p > 1$ e $q < n$ tal que,

$$a_{ij} = 0, \forall i + p \leq j \text{ ou } j + q \leq i$$

A constante $w = p + q - 1$ é denominada de bandagem de A . Se $w = 3$, A é dita tridiagonal, $w = 5$ A é dita pentadiagonal, etc.

2.8 Um autovalor de uma matriz A $n \times n$ é um número λ real ou complexo que para algum vetor $v \neq 0$, satisfaz a equação matricial: $(A - \lambda I)v = 0$. Qualquer vetor v não-nulo que satisfaz a mesma equação é chamado um *autovetor* de A associado ao autovalor λ .

2.9 Se λ_{\max} e λ_{\min} são o maior e menor autovalores, respectivamente, da matriz simétrica positiva definida C , então o número de condição de C é $\kappa_2 = \lambda_{\max}(C) / \lambda_{\min}(C)$.

2.10 *Espectro* é o conjunto de todos os autovalores da matriz A .

2.11 O *raio espectral* de uma matriz A é definido por,

$\rho(A) = \max |\lambda|$, onde λ representa os autovalores da matriz A.

Nota: Para $\lambda = \alpha + \beta i$ tem-se $|\lambda| = (\alpha^2 + \beta^2)^{1/2}$

2.12 Uma classe de normas, geralmente usada são as p -normas definidas por:

$$\|v\|_p = (|v_1|^p + |v_2|^p + \dots + |v_n|^p)^{1/p} \quad p \geq 1$$

dessas as normas 1, 2 e ∞ são as mais importantes:

$$\|v\|_1 = |v_1| + |v_2| + \dots + |v_n|$$

$$\|v\|_2 \equiv \sqrt{(v, v)} = \left[\sum_{i=1}^n |v_i|^2 \right]^{1/2}$$

$$\|v\|_\infty \equiv \max_{i=1,2,\dots,n} |v_i|$$

2.13 Algumas propriedades das normas da matriz, em especial as p -normas (aqui para $p=1, 2, \infty$) são apresentadas a seguir.

$$\|A\|_1 \equiv \max_{j=1,2,\dots,n} \left\{ \sum_{i=1}^m |a_{ij}| \right\}$$

$$\|A\|_2 \equiv \sqrt{\rho(AA^T)}$$

$$\|A\|_\infty \equiv \max_{i=1,2,\dots,n} \left\{ \sum_{j=1}^n |a_{ij}| \right\}$$

ρ é o raio espectral da matriz, usa-se a transposta da matriz A se os elementos de A são reais.

2.14 Quando os elementos da matriz A são números complexos, a *matriz conjugada* de A, denotada por \overline{A} , é aquela onde cada elemento de A é substituído por seu conjugado.

2.15 A matriz A é *Hermitiana* se $A = \overline{A^T}$, onde \overline{A} é a matriz conjugada de A. Então uma matriz A é *Hermitiana* se $a_{ij} = \overline{a_{ji}}$ para todos os valores de i e j. Evidentemente, os elementos diagonais de uma matriz *Hermitiana* são números reais. Uma matriz quadrada tal que $-A = \overline{A^T}$ é *anti-hermitiana*, então $a_{ij} = -\overline{a_{ji}}$. Logo, os elementos diagonais de uma matriz *anti-hermitiana* são ou zero ou imaginários puros.

3 Geometria

3.1 *Problema do par mais próximo* consiste em determinar quais são os dois pontos que estão à menor distância, dentre todos os pares de pontos, em uma coleção finita de pontos dados, é um problema fundamental em geometria computacional tanto pela simplicidade de seu enunciado quanto pela importância de suas aplicações.

3.2 Duas retas r e s são *ortogonais* se, $m_r \cdot m_s = -1$, onde m_r é o coeficiente angular da reta r calculado como,

$$m_r = \text{tg}(\alpha_r) = -\frac{a_r}{b_r} \quad \text{e analogamente tem-se}$$

$$m_s = \operatorname{tg}(\alpha_s) = -\frac{a_s}{b_s}$$

onde $a_k x + b_k y = \text{constante}$, para $k = r$ e s (equação linear da reta). Pode-se dizer que as retas formam um ângulo de 90° entre si, ou seja, são perpendiculares.

3.3 Quaisquer que sejam os pontos $A(x_A, y_A)$ e $B(x_B, y_B)$ diz-se que $M(x_M, y_M)$ é o *ponto médio* de \overline{AB} se,

$$x_M = \frac{x_A + x_B}{2} \quad y_M = \frac{y_A + y_B}{2}$$

Apêndice B - Equações e Propriedades do Gradiente Conjugado

As equivalências entre as fórmulas para o cálculo de α , r_i e β , apresentadas no algoritmo do Gradiente Conjugado são demonstradas, fazendo uso das demais equações presentes no mesmo, na literatura reporte-se ao Reid (1971).

Aqui o produto interno entre vetores é, usualmente, o produto escalar ordinário dado por,

$$x^T y = (x, y) \quad (\text{B.1})$$

i) para provar a equivalência entre as fórmulas das equações (6.25) e (6.31), basta mostrar que a relação B.2 é satisfeita, 4.9 4.15

$$(r^k, r^k) = (r^k, p^k) \quad (\text{B.2})$$

• para $k = 0$ é claramente verdade, pois $p^0 = r^0$, pela equação (6.24).

• para $k > 0$ usando a equação (6.29) vem, 4.8

$r^{k+1} = p^{k+1} - \beta^k p^k$, multiplicando ambos os membros da equação por r^{k+1} vem, 4.13

$$(r^{k+1}, r^{k+1}) = (p^{k+1}, r^{k+1}) + \beta^k (p^k, r^{k+1}) \quad (\text{B.3})$$

pela equação (6.31) tem-se, 4.25

$\alpha^{k+1} (p^{k+1}, Ap^{k+1}) = (p^{k+1}, r^{k+1})$, substituindo na equação anterior

$(r^{k+1}, r^{k+1}) = \alpha^{k+1} (p^{k+1}, Ap^{k+1}) - \beta^k (p^k, r^{k+1})$, a equação (6.25) é verdadeira se, 4.9

$(p^k, r^{k+1}) = 0$ logo pela equação (B.3) tem-se: $(r^{k+1}, r^{k+1}) = (p^{k+1}, r^{k+1})$

ii) já a equivalência das duas fórmulas usadas para calcular o resíduo são demonstradas a seguir,

$(\Rightarrow) r^{k+1} = b - Ax^{k+1}$ fazendo uso da equação (6.26) segue a demonstração, 4.10

$r^{k+1} = b - A(x^k + \alpha^k p^k) \Rightarrow r^{k+1} = (b - Ax^k) - \alpha^k Ap^k \Rightarrow r^{k+1} = r^k - \alpha^k Ap^k$

$(\Leftarrow) r^{k+1} = r^k - \alpha^k Ap^k$ fazendo uso da equação (6.26) e (6.30) segue, 4.10 4.14

$r^{k+1} = b - Ax^k - \alpha^k Ap^k \Rightarrow r^{k+1} = b - A(x^{k+1} - \alpha^k p^k) - \alpha^k Ap^k \Rightarrow r^{k+1} = b - Ax^{k+1}$

iii) A terceira equivalência será provada com o uso das condições de ortogonalidade, que são,

$$(r^k, p^j) = 0 \text{ para } k > j \tag{B.4}$$

$$(p^k, Ap^j) = 0 \text{ para } k \neq j \tag{B.5}$$

$$(r^k, r^j) = 0 \text{ para } k \neq j \tag{B.6}$$

onde k e j são os passos no método iterativo e não a ordem dos vetores. As relações de ortogonalidade B.4 e B.5 são provadas pela indução, ou seja, elas serão usadas como hipótese para se provar que é verdade para $k+1$, ou seja, o $k+1$ -ésimo passo iterativo.

↳ Para a prova de B.4, toma-se a equação (6.27) e multiplica-se ambos os membros por p^j
 $(r^{k+1}, p^j) = (r^k, p^j) - \alpha^k (Ap^k, p^j)$ mas por hipótese $(r^k, p^j) = 0$ e $(Ap^k, p^j) = 0$, logo

$$(r^{k+1}, p^j) = 0 \text{ para } k > j. \tag{B.7}$$

↳ Para a prova de B.5, toma-se a equação (6.29) e multiplica-se ambos os membros por Ap^j
 $(p^{k+1}, Ap^j) = (r^{k+1}, Ap^j) + \beta^k (p^k, Ap^j)$ isola-se Ap^j na equação (6.27) encontrando,
 $Ap^j = (r^j - r^{j+1})(\alpha^j)^{-1}$ voltando na equação anterior vem,

$$(p^{k+1}, Ap^j) = (\alpha^j)^{-1} (r^{k+1}, r^j - r^{j+1}) - \beta^k (p^k, Ap^j) \tag{B.8}$$

para encontrar r^j e r^{j+1} faz-se uso da equação (6.29) onde,

$$r^j = p^j + \beta^{j-1} p^{j-1} \text{ e}$$

$$r^{j+1} = p^{j+1} + \beta^j p^j \text{ substituindo na equação (B.8)}$$

$$(p^{k+1}, Ap^j) = (\alpha^j)^{-1} (r^{k+1}, p^j + \beta^{j-1} p^{j-1} - p^{j+1} - \beta^j p^j) - \beta^k (p^k, Ap^j),$$

logo pelas equações B.7 e B.5 prova-se a veracidade da equação B.7

Será verdade para $j = 0$, se for usado $r^0 = p^0$ e $\beta^{-1} p^{-1} = 0$.

↳ Para a prova de B.6, assume-se sem perda de generalidade que $i > j$

- Se $j > 0$ faz-se uso do cálculo de r^j ,

$(r^k, r^j) = (r^k, p^j + \beta^{j-1}p^{j-1}) = (r^k, p^j) + \beta^{j-1}(r^k, p^{j-1})$, logo pela equação (B.4) esta expressão é nula.

- Se $j = 0$ faz-se uso de $r^0 = p^0$

$(r^i, r^0) = (r^i, p^0)$ e pela equação B.3 vem, $(r^i, r^0) = 0$.

Agora pode-se justificar a fórmula alternativa para β^k , partindo-se da equação (6.32) e encontrando a equação (6.28), o que mostra suas equivalências. Usando a equação (6.27),

$$\beta^k = \frac{(-r^{k+1}, Ap^k)}{(p^k, Ap^k)} = \frac{-(r^{k+1}, r^{k+1} - r^k)(-\alpha^k)^{-1}}{(p^k, r^{k+1} - r^k)(-\alpha^k)^{-1}}$$

a seguir faz-se necessário o uso da condição de ortogonalidade B.6 e B.4, por fim usa-se a equação B.5, o que gera

$$\beta^k = \frac{-(r^{k+1}, r^{k+1})}{(p^k, r^{k+1}) - (p^k, r^k)} + \frac{(r^{k+1}, r^k)}{(p^k, r^{k+1}) - (p^k, r^k)} = \frac{(r^{k+1}, r^{k+1})}{(r^k, r^k)}$$