

UNIVERSIDADE FEDERAL DO PARANÁ

DAIANE CRISTINA ZANATTA

ESTUDO DE PARÂMETROS DO MÉTODO *MULTIGRID GEOMÉTRICO* PARA
EQUAÇÕES 2D EM CFD USANDO MALHAS CURVILÍNEAS ESTRUTURADAS NÃO
ORTOGONAIAS

Curitiba

2019

DAIANE CRISTINA ZANATTA

ESTUDO DE PARÂMETROS DO MÉTODO *MULTIGRID GEOMÉTRICO* PARA
EQUAÇÕES 2D EM CFD USANDO MALHAS CURVILÍNEAS ESTRUTURADAS NÃO
ORTOGONAIAS

Tese apresentada ao Programa de Pós-Graduação
em Método Numérico em Engenharia, Área de Con-
centração em Mecânica Computacional, Setor de
Ciências Exatas e de Tecnologia, Universidade Fe-
deral do Paraná, como requisito parcial para a ob-
tenção do grau de Doutora em Métodos Numéricos
em Engenharia.

Orientador: Prof. Dr. Luciano Kiyoshi Araki
Coorientador: Prof. Dr. Marcio Augusto Villela
Pinto

Curitiba

2019

Catalogação na Fonte: Sistema de Bibliotecas, UFPR
Biblioteca de Ciência e Tecnologia

Z27a

Zanatta, Daiane Cristina

Estudo de parâmetros do método multigrid geométrico para equações 2D em CFD usando malhas curvilíneas estruturadas não ortogonais [recurso eletrônico] / Daiane Cristina Zanatta. – Curitiba, 2019.

Tese - Universidade Federal do Paraná, Setor de Ciências Exatas, Programa de Pós-Graduação em Método Numérico em Engenharia, 2019.

Orientador: Luciano Kiyoshi Araki.

Coorientador: Marcio Augusto Villela Pinto.

1. Fluidodinâmica computacional. 2 Equações diferenciais elípticas. 3. Algoritmos. I. Universidade Federal do Paraná. II. Araki, Luciano Kiyoshi. III. Pinto, Marcio Augusto Villela. IV. Título.

CDD: 532.05

Bibliotecária: Vanusa Maciel CRB- 9/1928



MINISTÉRIO DA EDUCAÇÃO
SETOR DE CIENCIAS EXATAS
UNIVERSIDADE FEDERAL DO PARANÁ
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO MÉTODOS NUMÉRICOS
EM ENGENHARIA - 40001016030P0

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em MÉTODOS NUMÉRICOS EM ENGENHARIA da Universidade Federal do Paraná foram convocados para realizar a arguição da tese de Doutorado de **DAIANE CRISTINA ZANATTA** intitulada: **Estudo de parâmetros do método multigrid geométrico para equações 2D em CFD usando malhas curvilíneas estruturadas não-ortogonais**, sob orientação do Prof. Dr. LUCIANO KIYOSHI ARAKI, que após terem inquirido a aluna e realizada a avaliação do trabalho, são de parecer pela sua APROVADA no rito de defesa.

A outorga do título de doutor está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

CURITIBA, 13 de Novembro de 2019.

LUCIANO KIYOSHI ARAKI
Presidente da Banca Examinadora

SIMONE DE FATIMA TOMAZZONI GONÇALVES
Avaliador Externo (UNIVERSIDADE FEDERAL DO PARANÁ)

VIVIANA COCCO MARIANI
Avaliador Externo (PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ)

SAÚLO POMPONET OLIVEIRA
Avaliador Externo (UNIVERSIDADE FEDERAL DO PARANÁ)

FABIANE DE OLIVEIRA
Avaliador Externo (UNIVERSIDADE ESTADUAL DE PONTA GROSSA)

AGRADECIMENTOS

Agradeço a Deus, por tudo.

Aos meus pais Angelina e Benildo, pelo incentivo ao estudo.

Ao meu namorado Rodrigo por estar sempre próximo.

Ao meu orientador professor Luciano Kiyoshi Araki e ao meu coorientador professor Marcio Augusto Villela Pinto por todos os ensinamentos, pela dedicação e paciência.

Agradeço aos membros da banca examinadora, professoras Dra Simone de Fátima Tomazzoni Gonçalves, Dra Viviana Cocco Mariani, Dra. Fabiane de Oliveira e professor Dr. Saulo Pomponet Oliveira pelo tempo dispensado à leitura deste trabalho e pelas sugestões recebidas.

Ao Programa de Pós-Graduação em Métodos Numéricos em Engenharia (PPGMNE) da Universidade Federal do Paraná (UFPR) pela oportunidade de cursar o doutorado.

Agradeço aos amigos e colegas do Laboratório de Experimentação Numérica pela amizade e disposição em ajudar.

A Universidade Estadual do Centro-Oeste (UNICENTRO) pela licença concedida para a dedicação aos estudos.

RESUMO

Muitos problemas de Engenharia estão relacionados a geometrias em que o uso de um sistema de coordenadas cartesianas, cilíndricas ou esféricas não se mostra prático ou adequado, sendo preferível, por exemplo, empregar malhas curvilíneas estruturadas. A influência de alguns parâmetros do método *multigrid* geométrico sobre o tempo da Unidade Central de Processamento (CPU) na resolução de problemas de Dinâmica dos Fluidos Computacional usando malhas curvilíneas estruturadas não ortogonais foi investigada. Os parâmetros investigados são o número de iterações internas do solver (ν), o número das malhas (L), o número de incógnitas (N) e os métodos Gauss-Seidel com ordenação lexicográfica (GS-Lex), Gauss-Seidel linha em η (η -linha-GS), *Modified Strongly Implicit* (MSI) e decomposição LU incompleta modificada (MILU). Para a geração de malhas curvilíneas estruturadas não ortogonais foram empregados métodos que utilizam interpolação de Lagrange e sistema de equações diferenciais elípticas. As equações diferenciais foram discretizadas usando Método dos Volumes Finitos com esquema de aproximação central de segunda ordem e correção adiada. As condições de contorno, do tipo Dirichlet, foram aplicadas mediante a técnica de volumes fictícios. Para a resolução do sistema de equações algébricas resultante da discretização, foi utilizado o método *multigrid* geométrico com esquema de aproximação completa, ciclo V e razão de engrossamento padrão. Os estudos do efeito dos parâmetros ν , L e N e dos solvers GS-Lex, η -linha-GS, MSI e MILU no tempo da CPU se deram para problemas com tamanhos até 4096×4096 . Verificou-se principalmente que: entre os suavizadores empregados, os solvers MSI e MILU produzem algoritmos mais eficientes para os problemas estudados; o número de iterações internas que obteve o melhor desempenho médio para o problema de Poisson para malhas geradas usando interpolação de Lagrange e equações elípticas, é diferente, porém é o mesmo para o problema de Burgers para ambos os geradores de malhas e o solver MSI; o número de níveis de malha que obteve o melhor desempenho médio para o problema de Poisson para malhas geradas usando interpolação de Lagrange e equações elípticas, também é diferente, porém, é igual ao número máximo menos um para o problema de Burgers para ambos os geradores de malhas e o solver MSI; o solver MSI é mais rápido que o MILU para ambos os geradores de malha; e as soluções são mais acuradas para o problema de Burgers com malhas geradas usando equações elípticas.

Palavras-chave: Dinâmica dos Fluidos Computacional. Método dos Volumes Finitos. Malhas curvilíneas não ortogonais. *Multigrid* geométrico. Equação de Poisson. Equações de Burgers.

ABSTRACT

Many Engineering problems are related to geometries in which the use of a cartesian, cylindrical or spherical coordinate system is not practical or suitable, and it is preferable, for example, to employ structured curvilinear grids. The influence of some parameters of the geometric multigrid method on the time of the Central Processing Unit (CPU) in the solving Computational Fluid Dynamics problems with the use of non-orthogonal structured curvilinear grids was investigated. The parameters number of inner iterations of the solver (ν), number of grids (L) and number of unknowns (N), as well as the solvers, lexicographical Gauss-Seidel (Lex-GS), η -line Gauss-Seidel (η -line-GS), Modified Strongly Implicit (MSI) and modified incomplete LU decomposition (MILU) were assessed. Methods which employs Lagrange interpolation and elliptic equations system were used to generate the non-orthogonal structured curvilinear grids. The differential equations were discretized by Finite Volume Method with second-order approximation Central Differencing Scheme and deferred correction. Dirichlet boundary conditions were employed according to the ghost cell approach. Geometric multigrid method with Full Approximation Scheme, V-cycle and standard coarsening ratio was used to solve the system of algebraic equations that resulted from the discretization of equations. Problems up to size 4096×4096 volumes were employed in the study of the influence of the aforementioned parameters ν , L e N as well as the Lex-GS, η -line-GS, MSI and MILU solvers on the CPU time. The main results that should be emphasized are: from the solvers employed, MSI and MILU are the most efficient for the problems assessed; the number of inner iterations of the solver that obtained the best average performance for the Poisson problem for grids generated using Lagrange interpolation and elliptic equations, is different, however it has same for the Burgers problem for both grids generators and the MSI solver; the number of grids levels that obtained the best average performance for the Poisson problem for grids generated using Lagrange interpolation and elliptic equations is also different, though it is equal to the maximum number minus one for the Burgers problem for both grids generators and the MSI solver; MSI solver is faster than MILU for both grids generators; and the solutions are more accurate for the Burgers problem with grids generated using elliptic equations.

Keywords: Computational Fluid Dynamics. Finite volume method. Non-orthogonal curvilinear grids. Geometric multigrid. Poisson equation. Burgers equations.

LISTA DE ILUSTRAÇÕES

FIGURA 1 – DISCRETIZAÇÃO UTILIZANDO MALHA (a) ESTRUTURADA CARTESIANA; (b) ESTRUTURADA GENERALIZADA; (c) NÃO ESTRUTURADA	20
FIGURA 2 – DISPOSIÇÃO DE UM VOLUME DE CONTROLE P E SEUS VIZINHOS N (NORTE), S (SUL), E (LESTE) E W (OESTE), EM UMA MALHA BIDIMENSIONAL UNIFORME	36
FIGURA 3 – CONDIÇÕES DE CONTORNOS COM VOLUMES FICTÍCIOS . .	39
FIGURA 4 – ORDENAÇÃO LEXICOGRÁFICA EM UMA MALHA BIDIMENSIONAL	41
FIGURA 5 – NOVE VOLUMES EM UMA MALHA BIDIMENSIONAL UNIFORME	41
FIGURA 6 – ORDENAÇÃO DOS PONTOS DA MALHA POR LINHA NA DIREÇÃO Y	42
FIGURA 7 – ESTRUTURA DA MATRIZ \mathbf{A}	42
FIGURA 8 – ESTRUTURA DAS MATRIZES \mathbf{L} E \mathbf{U} COM A DECOMPOSIÇÃO LU CONVENCIONAL	43
FIGURA 9 – ILU(0) PARA UMA MATRIZ COM 9 DIAGONAIS	44
FIGURA 10 – MALHA BIDIMENSIONAL DO PROCEDIMENTO ILU(0) PARA O ESQUEMA COM 9 VOLUMES	44
FIGURA 11 – MATRIZ \mathbf{A} REESCRITA	44
FIGURA 12 – MATRIZES \mathbf{L} , \mathbf{D} E \mathbf{U}	45
FIGURA 13 – MATRIZES \mathbf{L} E \mathbf{U} PARA O MÉTODO MSI	48
FIGURA 14 – MATRIZ \mathbf{M} PARA O PROCEDIMENTO MSI	49
FIGURA 15 – MALHA BIDIMENSIONAL PARA O ESQUEMA COM 9 VOLUMES PARA O PROCEDIMENTO MSI	49
FIGURA 16 – DOMÍNIO (a) FÍSICO E (b) TRANSFORMADO	51
FIGURA 17 – COMPRIMENTOS NO DOMÍNIO FÍSICO	54
FIGURA 18 – ÁREAS DO DOMÍNIO FÍSICO (a) E PLANO TRANSFORMADO (b)	55
FIGURA 19 – MALHA PARA O TRIÂNGULO (a) E PLANO TRANSFORMADO (b)	55
FIGURA 20 – MALHA PARA UMA GEOMETRIA MULTIPLAMENTE CONEXA (a) E PLANO TRANSFORMADO (b)	56
FIGURA 21 – TRATAMENTO DOS CORTES DA FIGURA 20a	57
FIGURA 22 – INTERPOLAÇÃO UNIDIMENSIONAL	58
FIGURA 23 – MALHA GERADA UTILIZANDO INTERPOLAÇÃO DE LAGRANGE	59
FIGURA 24 – MALHA GERADA UTILIZANDO A SOLUÇÃO DO SISTEMA DE EQUAÇÕES ELÍPTICAS	60

FIGURA 25 – PONTOS VIZINHOS DO PONTO P NO PLANO TRANSFORMADO	65
FIGURA 26 – SEQUÊNCIA DE TRÊS MALHAS COM RAZÃO DE ENGROSSAMENTO PADRÃO	68
FIGURA 27 – MODOS DE FOURIER	70
FIGURA 28 – MODO DO ERRO NA MALHA Ω^h PROJETADO NA MALHA GROSSA Ω^{2h}	71
FIGURA 29 – SUAVIZAÇÃO DO ERRO COM O MÉTODO GAUSS-SEIDEL NA MALHA 33×33	72
FIGURA 30 – SUAVIZAÇÃO DO ERRO COM O MÉTODO GAUSS-SEIDEL NA MALHA 17×17	73
FIGURA 31 – DISPOSIÇÃO DO VOLUME DA MALHA FINA E DOS VOLUMES DA MALHA IMEDIATAMENTE MAIS GROSSA	74
FIGURA 32 – ESQUEMA DE MALHAS PARA OPERADOR DE PROLONGAÇÃO DA MALHA GROSSA (PONTOS A, B, C E D) PARA A MALHA IMEDIATAMENTE MAIS FINA (PONTOS $1, 2, 3$ E 4)	75
FIGURA 33 – CICLOS V, W E F	76
FIGURA 34 – DOMÍNIO DE CÁLCULO	78
FIGURA 35 – VOLUME DE CONTROLE E SEUS VIZINHOS USADOS PARA A DISCRETIZAÇÃO DA EQUAÇÃO TRANSFORMADA	87
FIGURA 36 – CONTORNO LESTE COM OS VOLUMES FICTÍCIOS	89
FIGURA 37 – VOLUME P COM SEUS RESPECTIVOS VÉRTICES A, B, C E D	90
FIGURA 38 – ESQUEMA DE MALHAS NO DOMÍNIO COMPUTACIONAL PARA RESTRIÇÃO DAS MÉTRICAS DA TRANSFORMAÇÃO DE COORDENADAS DA MALHA FINA (PONTOS $1, 2, 3, \dots, 36$) PARA A MALHA IMEDIATAMENTE MAIS GROSSA (PONTOS A, B, C, D, F, G, H, I E O)	95
FIGURA 39 – COMPORTAMENTO DA NORMA l_1 DO RESÍDUO NA ITERAÇÃO ATUAL ADIMENSIONALIZADA PELO RESÍDUO NA ESTIMATIVA INICIAL EM FUNÇÃO DO NÚMERO DE CICLOS V PARA UMA MALHA $N = 4096 \times 4096$ GERADA POR INTERPOLAÇÕES DE LAGRANGE	98
FIGURA 40 – FATOR DE CONVERGÊNCIA MÉDIO EMPÍRICO EM MALHAS GERADAS PELA INTERPOLAÇÃO DE LAGRANGE	100
FIGURA 41 – FATOR DE CONVERGÊNCIA MÉDIO EMPÍRICO EM MALHAS GERADAS POR EQUAÇÕES ELÍPTICAS	100
FIGURA 42 – FATOR DE CONVERGÊNCIA MÉDIO EMPÍRICO VERSUS N	101

FIGURA 43 – EFEITO DO NÚMERO DE ITERAÇÕES INTERNAS SOBRE O TEMPO DE CPU (SEGUNDOS) PARA MALHAS GERADAS USANDO INTERPOLAÇÃO DE LAGRANGE	102
FIGURA 44 – EFEITO DO NÚMERO DE ITERAÇÕES INTERNAS SOBRE O TEMPO DE CPU (SEGUNDOS) PARA MALHAS GERADAS USANDO EQUAÇÕES ELÍPTICAS	103
FIGURA 45 – EFEITO DO NÚMERO DE NÍVEIS DE MALHA SOBRE O TEMPO DE CPU PARA MALHAS GERADAS USANDO INTERPOLAÇÃO DE LAGRANGE	105
FIGURA 46 – EFEITO DO NÚMERO DE NÍVEIS DE MALHAS SOBRE O TEMPO DE CPU PARA MALHAS GERADAS USANDO EQUAÇÕES ELÍPTICAS	106
FIGURA 47 – EFEITO DO NÚMERO DE INCÓGNITAS SOBRE O TEMPO DE CPU	108
FIGURA 48 – EFEITO DO NÚMERO DE ITERAÇÕES INTERNAS SOBRE O TEMPO DE CPU PARA O PROBLEMA DE BURGERS UTILIZANDO O <i>SOLVER</i> MILU PARA MALHAS GERADAS USANDO INTERPOLAÇÃO DE LAGRANGE	111
FIGURA 49 – EFEITO DO NÚMERO DE ITERAÇÕES INTERNAS SOBRE O TEMPO DE CPU PARA O PROBLEMA DE BURGERS UTILIZANDO O <i>SOLVER</i> MILU PARA MALHAS GERADAS USANDO EQUAÇÕES ELÍPTICAS	111
FIGURA 50 – EFEITO DO NÚMERO DE NÍVEIS DE MALHA SOBRE O TEMPO DE CPU PARA O PROBLEMA DE BURGERS UTILIZANDO O <i>SOLVER</i> MILU E MALHAS GERADAS USANDO INTERPOLAÇÃO DE LAGRANGE	112
FIGURA 51 – EFEITO DO NÚMERO DE NÍVEIS DE MALHA SOBRE O TEMPO DE CPU PARA O PROBLEMA DE BURGERS UTILIZANDO O <i>SOLVER</i> MILU E MALHAS GERADAS USANDO EQUAÇÕES ELÍPTICAS	112
FIGURA 52 – COMPARATIVO DO TEMPO DE CPU (SEGUNDOS) VERSUS N PARA O PROBLEMA DE BURGERS COM OS <i>SOLVERS</i> MSI E MILU PARA MALHAS GERADAS USANDO INTERPOLAÇÃO DE LAGRANGE	113
FIGURA 53 – COMPARATIVO DO TEMPO DE CPU (SEGUNDOS) VERSUS N PARA O PROBLEMA DE BURGERS COM OS <i>SOLVERS</i> MSI E MILU PARA MALHAS GERADAS USANDO EQUAÇÕES ELÍPTICAS	114

FIGURA A.1–ORDEM EFETIVA DA NORMA INFINITO DO ERRO NA SOLUÇÃO T E DA TEMPERATURA MÉDIA EM FUNÇÃO DO ESPAÇAMENTO DA MALHA, CALCULADA PARA A EQUAÇÃO DE POISSON PARA MALHAS GERADAS USANDO INTERPOLAÇÃO DE LAGRANGE	129
FIGURA A.2–ORDEM APARENTE DA NORMA INFINITO DO ERRO NA SOLUÇÃO T E DA TEMPERATURA MÉDIA EM FUNÇÃO DO ESPAÇAMENTO DA MALHA, CALCULADA PARA A EQUAÇÃO DE POISSON PARA MALHAS GERADAS USANDO INTERPOLAÇÃO DE LAGRANGE	129
FIGURA A.3–ORDEM EFETIVA DA NORMA INFINITO DO ERRO NA SOLUÇÃO T E DA TEMPERATURA MÉDIA EM FUNÇÃO DO ESPAÇAMENTO DA MALHA, CALCULADA PARA A EQUAÇÃO DE POISSON PARA MALHAS GERADAS USANDO EQUAÇÕES ELÍPTICAS	130
FIGURA A.4–ORDEM APARENTE DA NORMA INFINITO DO ERRO NA SOLUÇÃO T E DA TEMPERATURA MÉDIA EM FUNÇÃO DO ESPAÇAMENTO DA MALHA, CALCULADA PARA A EQUAÇÃO DE POISSON PARA MALHAS GERADAS USANDO EQUAÇÕES ELÍPTICAS	130
FIGURA A.5–ORDEM EFETIVA DA NORMA INFINITO DO ERRO NA SOLUÇÃO E DA FORÇA DO FLUIDO NA FRONTEIRA EM FUNÇÃO DO ESPAÇAMENTO DA MALHA, CALCULADA PARA AS EQUAÇÕES DE BURGERS PARA MALHAS GERADAS USANDO INTERPOLAÇÃO DE LAGRANGE	131
FIGURA A.6–ORDEM APARENTE DA NORMA INFINITO DO ERRO NA SOLUÇÃO E DA FORÇA DO FLUIDO NA FRONTEIRA EM FUNÇÃO DO ESPAÇAMENTO DA MALHA, CALCULADA PARA AS EQUAÇÕES DE BURGERS PARA MALHAS GERADAS USANDO INTERPOLAÇÃO DE LAGRANGE	131
FIGURA A.7–ORDEM EFETIVA DA NORMA INFINITO DO ERRO NA SOLUÇÃO E DA FORÇA DO FLUIDO NA FRONTEIRA EM FUNÇÃO DO ESPAÇAMENTO DA MALHA, CALCULADA PARA AS EQUAÇÕES DE BURGERS PARA MALHAS GERADAS USANDO EQUAÇÕES ELÍPTICAS	131

FIGURA A.8–ORDEM APARENTE DA NORMA INFINITO DO ERRO NA SOLUÇÃO E DA FORÇA DO FLUIDO NA FRONTEIRA EM FUNÇÃO DO ESPAÇAMENTO DA MALHA, CALCULADA PARA AS EQUAÇÕES DE BURGERS PARA MALHAS GERADAS USANDO EQUAÇÕES ELÍPTICAS	132
FIGURA A.9–DOMÍNIO NO FORMATO DE L E SEU RESPECTIVO DOMÍNIO COMPUTACIONAL	133
FIGURA A.10–ERRO NUMÉRICO EM FUNÇÃO DO ESPAÇAMENTO DA MALHA PARA A EQUAÇÃO DE POISSON PARA MALHAS GERADAS USANDO INTERPOLAÇÃO DE LAGRANGE	134
FIGURA A.11–ERRO NUMÉRICO EM FUNÇÃO DO ESPAÇAMENTO DA MALHA PARA A EQUAÇÃO DE POISSON PARA MALHAS GERADAS USANDO EQUAÇÕES ELÍPTICAS	134
FIGURA A.12–ERRO NUMÉRICO EM FUNÇÃO DO ESPAÇAMENTO DA MALHA PARA AS EQUAÇÕES DE BURGERS PARA MALHAS GERADAS USANDO INTERPOLAÇÃO DE LAGRANGE	134
FIGURA A.13–ERRO NUMÉRICO EM FUNÇÃO DO ESPAÇAMENTO DA MALHA PARA AS EQUAÇÕES DE BURGERS PARA MALHAS GERADAS USANDO EQUAÇÕES ELÍPTICAS	135

LISTA DE TABELAS

TABELA 1 – NÚMERO ÓTIMO DE ITERAÇÕES INTERNAS DO <i>SOLVER</i> PARA CADA PROBLEMA	103
TABELA 2 – DIFERENÇAS ENTRE OS TEMPOS DE CPU PARA O NÚMERO DE ITERAÇÕES INTERNAS DO <i>SOLVER</i> PARA PROBLEMAS COM MALHAS GERADAS USANDO INTERPOLAÇÃO DE LA- GRANGE	103
TABELA 3 – DIFERENÇAS ENTRE OS TEMPOS DE CPU PARA O NÚMERO DE ITERAÇÕES INTERNAS DO <i>SOLVER</i> PARA PROBLEMAS COM MALHAS GERADAS USANDO EQUAÇÕES ELÍPTICAS .	104
TABELA 4 – DIFERENÇAS ENTRE OS TEMPOS DE CPU PARA <i>L</i> PARA PROBLEMAS COM MALHAS GERADAS USANDO INTERPO- LAÇÃO DE LAGRANGE	106
TABELA 5 – DIFERENÇAS ENTRE OS TEMPOS DE CPU PARA <i>L</i> PARA PROBLEMAS COM MALHAS GERADAS USANDO EQUAÇÕES ELÍPTICAS	107
TABELA 6 – TEMPO DE CPU (SEGUNDOS) DOS MÉTODOS MG E SG PARA OS PROBLEMAS DE POISSON E DE BURGERS COM AMBOS OS GERADORES DE MALHAS	109
TABELA 7 – VALORES DE <i>p</i> E <i>c</i> PARA OS PROBLEMAS PARA MALHAS GE- RADAS USANDO INTERPOLAÇÃO DE LAGRANGE E EQUA- ÇÕES ELÍPTICAS	110
TABELA 8 – DIFERENÇAS ENTRE OS TEMPOS DE CPU PARA O NÚMERO DE ITERAÇÕES INTERNAS PARA O PROBLEMA DE BUR- GERS UTILIZANDO O <i>SOLVER MILU</i> E MALHAS GERADAS USANDO EQUAÇÕES ELÍPTICAS	111
TABELA 9 – DIFERENÇAS ENTRE OS TEMPOS DE CPU PARA <i>L</i> PARA O PROBLEMA DE BURGERS, UTILIZANDO O <i>SOLVER MILU</i> E MALHAS GERADAS USANDO INTERPOLAÇÃO DE LAGRANGE	113
TABELA 10 – DIFERENÇAS ENTRE OS TEMPOS DE CPU PARA <i>L</i> PARA O PROBLEMA DE BURGERS, UTILIZANDO O <i>SOLVER MILU</i> E MALHAS GERADAS USANDO EQUAÇÕES ELÍPTICAS	113
TABELA 11 – TEMPO DE CPU (SEGUNDOS) PARA O PROBLEMA DE BUR- GERS RESOLVIDO COM OS SOLVERS MSI e MILU E MALHAS GERADAS USANDO INTERPOLAÇÃO DE LAGRANGE	115
TABELA 12 – TEMPO DE CPU (SEGUNDOS) PARA O PROBLEMA DE BUR- GERS RESOLVIDO COM OS SOLVERS MSI e MILU E MALHAS GERADAS USANDO EQUAÇÕES ELÍPTICAS	115

TABELA 13 – NORMA INFINITO DO ERRO PARA O PROBLEMA DE BURGERS RESOLVIDO COM O MSI E AMBOS OS GERADORES DE MALHAS	115
--	-----

LISTA DE ABREVIATURAS E SIGLAS

ADI	– <i>Alternating Direction Implicit</i>
AMG	– <i>Algebraic Multigrid</i>
CDS	– <i>Central Differencing Scheme</i>
CFD	– <i>Computational Fluid Dynamics</i>
CPU	– <i>Central Processing Unit</i>
CS	– <i>Correction Scheme</i>
FAS	– <i>Full Approximation Scheme</i>
FMG	– <i>Full Multigrid</i>
GS-Lex	– Gauss-Seidel com a ordenação lexicográfica
GS-RB	– Gauss-Seidel <i>red-black</i>
GMG	– <i>Geometrical Multigrid</i>
ILU	– Decomposição LU incompleta
LFA	– Análise local de Fourier
LU	– <i>Lower and Upper</i>
MDF	– Método das Diferenças Finitas
MEF	– Método dos Elementos Finitos
MG	– <i>Multigrid</i>
MILU	– Decomposição LU incompleta modificada
MGRIT	– <i>Multigrid</i> com redução no tempo
MVF	– Método dos Volumes Finitos
MSI	– <i>Modified Strongly Implicit</i>
QML	– Quantidade de Movimento Linear
SCA	– <i>Symmetric Coupled Alternating Lines</i>
SCGS	– <i>Symmetric Coupled Gauss-Seidel</i>

SIP	– <i>Strongly Implicit Procedure</i>
SOR	– <i>Successive Over-Relaxation</i>
SUDS	– <i>Skew Upstream Differencing Scheme</i>
SWUDS	– <i>Skew Weighted Upstream Differencing Scheme</i>
TDMA	– <i>Tridiagonal Matrix Algorithm</i>
t_{CPU}	– Tempo de CPU
UDS	– <i>Upwind Differencing Scheme</i>
VC	– Volume de controle
WUDS	– <i>Weighted Upstream Differencing Scheme</i>
η -linha-GS	– Gauss-Seidel linha na direção η

LISTA DE SÍMBOLOS

Símbolo	– Descrição	Unidade
A	– Matriz dos coeficientes	
b	– Vetor independente do sistema linear	
c_p	– Calor específico calor a pressão constante	[J/KgK]
e	– Erro algébrico (ou apenas erro)	
E_h	– Erro de discretização	
h	– Tamanho do elemento (ou espaçamento) na malha computacional	
\mathbf{I}_h^{2h}	– Operador de restrição	
\mathbf{I}_{2h}^h	– Operador de prolongação	
k	– Número de ondas	
k_t	– Condutividade térmica	[W/mK]
L	– Número de níveis de malhas	
L	– Matriz triangular inferior	
$L_{máx}$	– Maior número possível de níveis de malhas	
$L_{ótimo}$	– Número ótimo de níveis de malhas	
l_2	– Norma euclidiana	
l_∞	– Norma infinito	
p_E	– Ordem efetiva do erro de discretização	
p_L	– Ordem assintótica do erro de discretização	
p_v	– Ordem verdadeira do erro	
p_U	– Ordem aparente do erro de discretização	
\dot{q}	– Taxa de calor gerado	[W/m ³]
R	– Matriz dos resíduos	
r	– Resíduo	

r	– Razão de engrossamento de malhas	
s_p	– Fator de aceleração (ou <i>speedup</i>)	
\mathbf{T}	– Matriz de iteração do método iterativo	
T	– Temperatura	$[K \text{ ou } {}^\circ C]$
\mathbf{U}	– Matriz triangular superior	
u	– Velocidade na direção x	$[m/s]$
v	– Velocidade na direção y	$[m/s]$
x, y, z	– Direções no sistema de coordenadas cartesianas	$[m]$

Símbolos gregos

ν	– Número de iterações internas do MG	
ν_1	– Pré-suavização	
ν_2	– Pós-suavização	
$\nu_{\text{ótimo}}$	– Número ótimo de iterações internas	
ρ	– Massa específica	$[kg/m^3]$
ρ_d	– Densidade	$[kg/m^3]$
ρ_m	– Fator de convergência médio empírico	
σ	– Parâmetro dos <i>solvers</i> MSI e MILU	
ϕ	– Solução numérica da variável de interesse	
$\hat{\phi}$	– Solução aproximada do sistema	
Φ	– Solução analítica da variável de interesse	

Subscritos e superscritos

E	– Volume de controle localizado ao leste do volume de controle geral	
h	– Malha fina	
$2h$	– Malha grossa	
i	– Posição do nó na direção x	
j	– Posição do nó na direção y	

- k – k-ésima iteração
- N – Volume de controle localizado ao norte do volume de controle geral
- NW – Volume de controle localizado ao noroeste do volume de controle geral
- NE – Volume de controle localizado ao nordeste do volume de controle geral
- P – Volume de controle geral
- S – Volume de controle localizado ao sul do volume de controle geral
- SW – Volume de controle localizado ao sudoeste do volume de controle geral
- SE – Volume de controle localizado ao sudeste do volume de controle geral
- W – Volume de controle localizado ao oeste do volume de controle geral

SUMÁRIO

1	INTRODUÇÃO	20
1.1	REVISÃO DE LITERATURA	23
1.1.1	Método <i>multigrid</i>	23
1.1.2	Método <i>multigrid</i> para sistemas de coordenadas curvilíneas	26
1.1.3	Equação de Burgers	28
1.2	MOTIVAÇÃO	32
1.3	OBJETIVOS	33
1.4	ORGANIZAÇÃO DA TESE	34
2	FUNDAMENTAÇÃO TEÓRICA	35
2.1	O MÉTODO DOS VOLUMES FINITOS	35
2.2	CONDIÇÕES DE CONTORNO	38
2.3	MÉTODOS ITERATIVOS DE RESOLUÇÃO PARA SISTEMAS DE EQUAÇÕES LINEARES	39
2.3.1	Método Gauss-Seidel e suas variantes	40
2.3.2	Decomposição LU incompleta modificada (MILU)	42
2.3.3	Método MSI	47
2.4	TRANSFORMAÇÃO DE COORDENADAS	51
2.5	TIPOS DE MAPEAMENTO	55
2.6	GERAÇÃO DO SISTEMA DE COORDENADAS CURVILÍNEAS	56
2.6.1	Métodos algébricos de geração de coordenadas	57
2.6.2	Métodos diferenciais de geração de coordenadas	59
3	MÉTODO <i>MULTIGRID</i>	67
3.1	CARACTERIZAÇÃO DO MÉTODO <i>MULTIGRID</i>	67
3.2	EQUAÇÃO RESIDUAL E ESQUEMAS CS E FAS	68
3.3	ANÁLISE DE ERROS	70
3.4	OPERADORES DE TRANSFERÊNCIAS ENTRE MALHAS	72
3.4.1	Operadores de restrição	73
3.4.2	Operadores de prolongação	74
3.5	CICLOS E ALGORITMO FAS	75
4	MODELOS MATEMÁTICOS E NUMÉRICOS	78
4.1	MODELOS MATEMÁTICOS	79
4.1.1	Equação de Poisson	79
4.1.2	Equações de Burgers	81

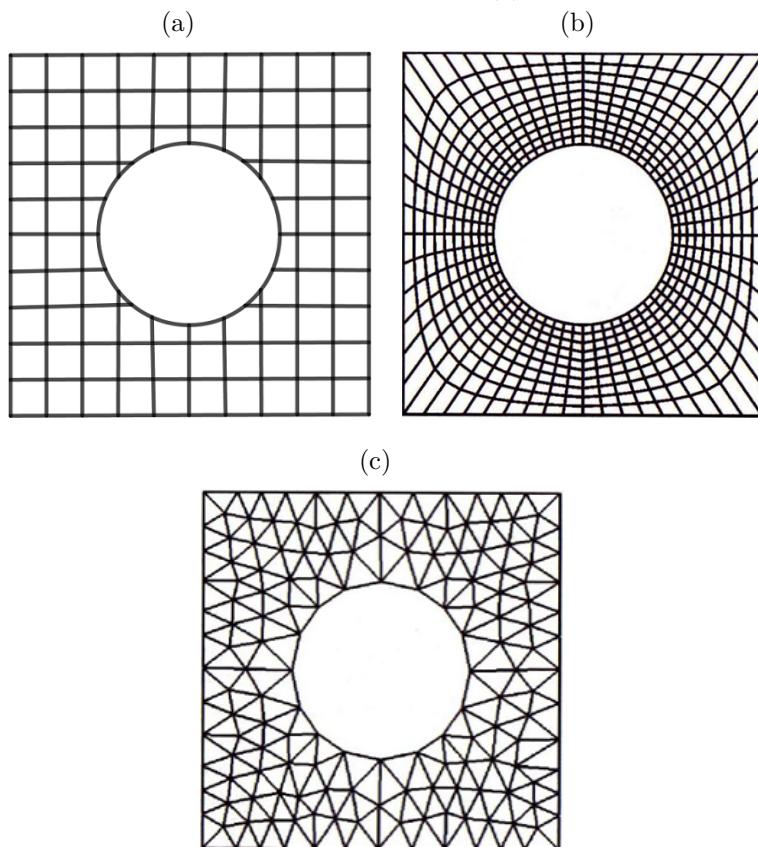
4.2	MODELOS NUMÉRICOS	86
4.2.1	Equação de Poisson	86
4.2.2	Equações de Burgers	91
4.3	CÁLCULO DAS MÉTRICAS DE TRANSFORMAÇÃO DE COORDENADAS NAS MALHAS GROSSAS	94
5	DETALHES COMPUTACIONAIS	96
5.1	FATOR DE CONVERGÊNCIA EMPÍRICO	96
5.2	RESOLUÇÃO DO SISTEMA DE EQUAÇÕES ALGÉBRICAS	96
5.3	DADOS DA IMPLEMENTAÇÃO	97
5.4	DEFINIÇÃO DA TOLERÂNCIA	97
6	RESULTADOS	99
6.1	ANÁLISE DO FATOR DE CONVERGÊNCIA MÉDIO EMPÍRICO	100
6.2	NÚMERO DE ITERAÇÕES INTERNAS DO <i>SOLVER</i>	102
6.3	NÚMERO DE NÍVEIS DE MALHA	105
6.4	ESFORÇO COMPUTACIONAL	107
6.5	COMPARATIVO ENTRE OS <i>SOLVERS</i> MSI E MILU	110
7	CONCLUSÃO	117
7.1	CONTRIBUIÇÕES	118
7.2	PROPOSTA DE TRABALHOS FUTUROS	119
	REFERÊNCIAS	120
	APÊNDICE A – VERIFICAÇÃO DOS CÓDIGOS COMPUTACIONAIS	127
A.1	VERIFICAÇÃO	127
A.2	ERRO DE DISCRETIZAÇÃO	128

1 INTRODUÇÃO

O crescente interesse pela modelagem dos fenômenos físicos, associado ao crescimento exponencial da tecnologia, desencadearam o desenvolvimento de uma grande variedade de técnicas de simulação numérica. A Dinâmica dos Fluidos Computacional (do inglês, *Computational Fluid Dynamics - CFD*) trata dos estudos de métodos computacionais para simulação de fenômenos que envolvem fluidos em movimento com ou sem troca de calor. O principal interesse é obter campos de velocidades, pressões e temperaturas na região de escoamento (FORTUNA, 2000).

Muitos problemas de Engenharia estão relacionados a geometrias complexas. Para essas geometrias o uso de um sistema de coordenadas cartesianas, cilíndricas ou esféricas não se mostra prático ou adequado. Por exemplo, uma discretização utilizando malha estruturada cartesiana para uma geometria com furo, mostrada na FIGURA 1a, não é adequada porque apresenta volumes quebrados na fronteira interna do domínio, sendo preferível uma discretização conforme a FIGURA 1b ou FIGURA 1c .

FIGURA 1 – DISCRETIZAÇÃO UTILIZANDO MALHA (a) ESTRUTURADA CARTESIANA; (b) ESTRUTURADA GENERALIZADA; (c) NÃO ESTRUTURADA



FONTE: Maliska (2010).

As malhas usadas em CFD para geometrias complexas podem ser classificadas fundamentalmente em dois grupos, malhas curvilíneas estruturadas e malhas não estruturadas

(VERSTEEG; MALALASEKERA, 2007). Malhas curvilíneas estruturadas são baseadas no mapeamento do domínio físico sobre o domínio computacional com um formato simples como, por exemplo, um retângulo. Usando esse tipo de malhas (sistema de coordenadas) as condições de contorno existentes nas fronteiras do domínio físico transferem-se exatamente para os contornos do domínio transformado, não necessitando de aproximações. Ainda, usando esse sistema de coordenadas, tem-se um sistema de coordenadas curvilíneas global, e é possível escrever e discretizar as equações diferenciais nesse sistema. Além disso, essas malhas possuem a vantagem de permitir fácil ordenação, uma vez que os volumes da malha possuem uma determinada lei de construção e cada volume tem sempre o mesmo número de vizinhos. Dessa forma, obtém-se matrizes de coeficientes do tipo diagonal do processo de discretização. Esse tipo de matriz de coeficientes permitem métodos de resolução numérica mais fáceis de serem desenvolvidos e mais eficientes (MALISKA, 2010).

Na discretização usando malhas não estruturadas, os volumes de controle conectam-se entre si de forma arbitrária e, consequentemente, o número de vizinhos é variável no domínio (MALISKA, 2010). Essas malhas são versáteis, com facilidade para adaptatividade e refinamento em regiões específicas do domínio. Entretanto, cresce a complexidade dos algoritmos para a solução das equações discretizadas, porque não existe uma lei de formação para a numeração dos volumes elementares e o número de volumes vizinhos pode variar de volume para volume, criando matrizes de coeficientes dependente da natureza do ordenamento. Consequentemente, os métodos de solução de sistemas lineares, incluindo o método *multigrid* são mais elaborados.

Portanto, existem, vantagens e desvantagens em cada um desses métodos de discretização do domínio. Neste trabalho, empregaram-se as malhas curvilíneas estruturadas não ortogonais, porque pode-se utilizar a metodologia numérica desenvolvida para as malhas estruturadas.

Em CFD, os métodos de discretização de modelos matemáticos mais utilizados são os Métodos das Diferenças Finitas (MDF) (ANDERSON; TANNEHILL; PLETCHER, 2013; BURDEN; FAIRES, 2016), dos Volumes Finitos (MVF) (PATANKAR, 1980; MALISKA, 2010) e dos Elementos Finitos (MEF) (HUGHES, 2000; VERSTEEG; MALALASEKERA, 2007; BURDEN; FAIRES, 2016). A discretização de modelos matemáticos com estes métodos aproximam um sistema de equações diferenciais por um sistema de equações algébricas da forma

$$\mathbf{A}\phi = \mathbf{b}, \quad (1.1)$$

sendo \mathbf{A} a matriz dos coeficientes, ϕ o vetor das incógnitas (referente à solução aproximada em cada ponto da malha) e \mathbf{b} o vetor dos termos independentes.

Os métodos de resolução numérica do sistema de equações algébricas dado pela equação (1.1) podem ser classificados como diretos ou iterativos (BURDEN; FAIRES,

2016).

Os métodos diretos fornecem a solução exata do sistema de equações, caso ela exista, exceto por erros de arredondamento, após uma sequência finita de operações (FORTUNA, 2000). Alguns exemplos de métodos diretos são a eliminação de Gauss e TDMA (do inglês, *Tridiagonal Matrix Algorithm*). A resolução para malhas muito refinadas utilizando *solvers* diretos pode ser inviável, devido à necessidade de armazenamento de dados e do alto tempo de CPU (do inglês, *Central Processing Unit*), tempo gasto pela unidade central de processamento. Em geral, os métodos diretos de solução não são empregados na resolução de problemas com grande número de incógnitas ou cujas matrizes de coeficientes sejam esparsas, porque pode haver um acúmulo de erros de arredondamento (FORTUNA, 2000). Nesses problemas utilizam-se os métodos iterativos.

Os métodos iterativos necessitam de uma estimativa inicial para a solução, a partir da qual é gerada a sequência de aproximações que, mediante certas condições, converge para a solução exata do sistema de equações. Neste trabalho, tais métodos são chamados de *solvers*. Para aplicações em CFD, os sistemas obtidos com a aplicação de métodos numéricos são, geralmente, de grande porte, podendo envolver até milhões de incógnitas e com matrizes de coeficientes esparsas. Considerando-se estas características, os métodos iterativos são mais apropriados em relação aos diretos, visto que demandam menor quantidade de memória computacional e o tempo de processamento requerido é muito menor (BURDEN; FAIRES, 2016). Alguns exemplos de métodos iterativos são os métodos de Jacobi, Gauss-Seidel lexicográfico (GS-Lex), SOR (do inglês, *Successive Over-Relaxation*) (BURDEN; FAIRES, 2016) e MSI (do inglês, *Modified Strongly Implicit*) (SCHNEIDER; ZEDAN, 1981).

O método *multigrid* refere-se a uma família de métodos iterativos utilizados para resolver sistemas de equações algébricas dado pela equação (1.1). Esse método percorre, ao longo do processo iterativo, várias malhas com diferentes graus de refinamento, conseguindo suavizar o erro de uma forma mais efetiva. A sequência com que as diversas malhas são visitadas caracteriza um ciclo *multigrid*, que pode ser do tipo V, W, F, entre outros. Para transferir informações da malha fina para a malha imediatamente mais grossa são utilizados operadores de restrição e para transferir informações da malha grossa para a malha imediatamente mais fina são utilizados operadores de prolongação (BRIGGS; HENSON; McCORMICK, 2000; TROTTERBERG; OOSTERLEE; SCHULLER, 2001). Para a solução das equações usando malhas curvilíneas estruturadas, restringiu-se também as métricas de transformação de coordenadas da malha mais fina. Em cada malha executam-se, com um *solver*, iterações até atingir o critério de convergência especificado para o sistema na malha mais fina. Assim, a cada malha, as diferentes componentes do erro são eficientemente suavizadas, acelerando a convergência do processo iterativo empregado na resolução do sistema (ANDERSON; TANNEHILL; PLETCHER, 2013).

O método *multigrid* é distinguido como *multigrid* geométrico (do inglês, *Geometrical Multigrid* - GMG) e o algébrico (do inglês, *Algebraic Multigrid* - AMG), indicados para malhas estruturadas e não estruturadas, respectivamente. Para ambos os tipos, podem ser usados dois tipos de esquemas, o esquema de correção (do inglês, *Correction Scheme* - CS) que é indicado para a resolução de problemas lineares e o esquema de aproximação completa (do inglês, *Full Approximation Scheme* - FAS) que é indicado para problemas não lineares. No esquema CS somente o resíduo é transferido para as malhas grossas, onde é resolvida a equação residual. Já no esquema FAS são transferidos o resíduo e a aproximação da solução para as malhas grossas (BRIGGS; HENSON; McCORMICK, 2000; TROTTERBERG; OOSTERLEE; SCHULLER, 2001).

1.1 REVISÃO DE LITERATURA

1.1.1 Método *multigrid*

Os primeiros estudos sobre o método *multigrid* foram realizados a partir da década de 1960 com os trabalhos de Fedorenko (1964) e Bakhvalov (1966). Fedorenko (1964) formulou um algoritmo *multigrid* para resolução da equação de Poisson, discretizada com um esquema padrão de diferenças finitas de cinco pontos em um domínio quadrado. Em sua pesquisa, Fedorenko (1964), mostrou que o custo computacional para se chegar a uma dada precisão é $O(N)$, sendo N o número de incógnitas. Pouco tempo depois, Bakhvalov (1966) generalizou o trabalho de Fedorenko (1964). Ele considerou uma equação diferencial parcial elíptica de segunda ordem com coeficientes variáveis e contínuos, discretizada com diferenças centrais em um domínio quadrado. Entretanto, as estimativas teóricas do trabalho não foram boas e o método foi deixado de lado, na época.

Na década 70, Brandt (1973; 1976) foram os primeiros trabalhos que reconheceram a eficiência e as características atrativas do método *multigrid*. Brandt (1977) apresentou uma análise teórica e numérica do método que incluiu uma introdução do *multigrid* com esquema FAS, específico para problemas não lineares. O trabalho também apresentou estudos sobre análise local de Fourier (do inglês, *Local Fourier Analysis* - LFA), razões de refinamento, taxa de convergência e a sistematização da ideia do FMG (do inglês, *Full Multigrid*), entre outras análises. Achi Brandt é conhecido como o pai do método *multigrid*. Suas principais publicações podem ser encontradas em <http://www.wisdom.weizmann.ac.il/~achi/>.

Nas décadas seguintes, surgiram muitos trabalhos com o método *multigrid* abordando o aperfeiçoamento do método e a resolução de novos problemas em CFD. Algumas aplicações do *multigrid* podem ser encontradas em Tannehill, Anderson e Pletcher (1997), Trottenberg, Oosterlee e Schuller (2001) e Ferziger e Peric (2002). Ainda nessa época, o método *multigrid* tornou-se um método padrão para a resolução de sistemas em muitas áreas, sendo estendido para a resolução de equações diferenciais parabólicas e hiperbólicas.

O método *multigrid* algébrico, indicado para malhas não estruturadas, foi desenvolvido nessa época na Universidade do Colorado em Denver (DOUGLAS, 1996).

Brandt (1998) buscou compreender as barreiras que impediam os pesquisadores de alcançar a eficiência teórica do método *multigrid*. O trabalho apresenta uma extensa tabela onde são listadas as principais dificuldades computacionais observadas e são apontadas algumas possibilidades para contornar tais dificuldades. Dentre as dificuldades apontadas encontram-se escoamentos recirculantes, características da malha, anisotropias, discretização e relaxação sobre e próximo aos contornos, entre outras. Um repositório de informações sobre o método *multigrid*, também foi criado nessa época pelo pesquisador Craig Douglas, contendo artigos, teses, informações bibliográficas e códigos computacionais, pode ser encontrado no sítio <http://www.mgnet.org/>.

Wesseling e Oosterlee (2001) fizeram uma revisão do desenvolvimento do *multigrid* geométrico, que é indicado para malhas estruturadas, enfatizando aplicações em CFD. Mostraram a eficiência do método *multigrid* em modelos acadêmicos e apresentaram o estado da arte (à época) do método para escoamentos compressíveis e incompressíveis. Nesse mesmo período, Stüben (2001) fez uma análise semelhante, mas enfocando o método *multigrid* algébrico. Afirmou que muitos métodos foram desenvolvidos, mas nenhum deles seria capaz de tratar com eficiência todos os problemas práticos em CFD. Apesar de não ter alcançado a plenitude da eficiência, esses autores apontaram o método *multigrid* como um dos mais importantes desenvolvimentos em análise numérica na segunda metade do século XX.

O método *multigrid* é o mais rápido na resolução de problemas elípticos de valor de contorno. Mas, apesar de não ter alcançado a plenitude da eficiência está entre os mais eficientes para outros tipos de equações diferenciais parciais, tais como parabólicas e hiperbólicas (YAVNEH, 2006).

No presente estudo, observou-se a existência de uma vasta literatura disponível onde o método *multigrid* é empregado com o objetivo de reduzir o tempo de CPU para problemas específicos de CFD. Trottenberg, Oosterlee e Schuller (2001) afirmaram que escolhas de parâmetros, tais como, *solver*, número de iterações internas em cada malha, tipos de ciclos e operadores de restrição e prolongação podem influenciar de forma substancial a taxa de convergência de um algoritmo.

Tannehill, Anderson e Pletcher (1997) fizeram uma análise do *multigrid* com o CS e o *solver* Gauss-Seidel lexicográfico com ciclo V. Para isso, estudaram o problema de Laplace 2D em um domínio quadrado com condições de contorno de Dirichlet, sistema de coordenadas cartesianas com malhas uniformes e o MDF. Verificaram que não houve redução significativa no tempo de CPU com o emprego do número máximo de malhas e que o uso de 4 ou 5 níveis de malhas foram suficientes para problemas de tamanhos

129×129 incógnitas.

Rabi e Lemos (2001) apresentaram alguns parâmetros relativos ao melhor desempenho do método *multigrid* com esquema CS e ciclos V e W, para o problema de advecção-difusão 2D, usando sistema de coordenadas cartesianas e discretizado usando MVF. Usaram uma iteração interna do *solver* para o ciclo V e duas para o ciclo W. Segundo Mesquita e Lemos (2004) os valores ideais para o número de níveis de malhas empregadas e o número de iterações internas do *solver* não são facilmente determinados e podem depender de características específicas do escoamento em questão.

Pinto e Marchi (2006) fizeram uma análise do *multigrid* com CS e FAS, empregando os *solvers* GS-Lex, ADI e MSI. Para isso, estudaram o problema de Laplace bidimensional, usando o sistema de coordenadas cartesianas e o MDF. Constataram que o ideal é empregar o número máximo possível de malhas e que o FAS é mais rápido do que o CS, mesmo para problemas lineares.

Kumar, Kumar e Das (2009) resolveram o clássico problema da cavidade quadrada com tampa móvel na formulação em variáveis primitivas, usando o MVF e MG com FAS. Apresentaram resultados para malhas com $N = 129 \times 129$, 257×257 e 513×513 volumes de controle, para diferentes Re. Afirmaram que o emprego de 2 a 4 malhas são suficientes para problemas de tamanho $N = 513 \times 513$ e que o emprego de um maior número de malhas não resultou em um ganho significativo. No entanto, não esclareceram se o relativo ganho é na taxa de convergência.

Oliveira et al. (2012) investigaram o efeito de alguns parâmetros do *multigrid* com o CS para os problemas de Laplace e Poisson bidimensional, com anisotropia geométrica, discretizados pelo MDF. Verificaram que para problemas isotrópicos e anisotrópicos, o solver Gauss-Seidel *red-black* com restrição por ponderação parcial, resultou em menor tempo de CPU em relação ao GS-Lex. Para o problema de Poisson anisotrópico obtiveram $\nu = 1$ como valor ótimo.

Suero et al. (2012) analisaram alguns parâmetros do AMG para as equações de Laplace e Poisson bidimensionais em malhas quadrangulares e para a equação de Laplace em malhas triangulares. Discretizaram as equações usando o MDF, em malhas quadrangulares, e o MVF em malhas triangulares. Utilizaram o CS, ciclo V e o *solver* GS-Lex. Verificaram que o número ótimo de iterações internas do *solver* é independente do *multigrid*, mas depende da malha. Também concluíram que o emprego do número máximo de malhas resultou no menor tempo de CPU, em ambas as malhas.

Santiago, Marchi e Souza (2015) analisaram a influência do esquema FAS com o *solver* MSI e do número de equações diferenciais sobre o desempenho do método *multigrid* para os problemas bidimensionais de Laplace, Navier e Burgers, discretizadas pelo MDF. Afirmaram que o número de equações diferenciais envolvidas no problema não afetou o

desempenho do *multigrid*.

1.1.2 Método *multigrid* para sistemas de coordenadas curvilíneas

No início de 1970 sistemas de coordenadas curvilíneas começaram a serem empregados entre os pesquisadores do MDF. Foi considerado um dos passos mais importantes dados para o tratamento de problemas de escoamentos de fluidos em geometrias complexas. Até a década de 70, provavelmente, com a preocupação de encontrar maneiras eficientes para tratar o acoplamento pressão-velocidade e as não-linearidades das equações de Navier-Stokes, poucos esforços tinham sido dedicados para resolver o problema da geometria irregular (MALISKA, 2010).

Após os trabalhos de Thompson, Thames e Mastin (1974; 1976; 1977) o uso de sistemas de coordenadas curvilíneas difundiu-se com rapidez. Tiveram o cuidado e a habilidade de apresentar os procedimentos para o uso desse sistema de coordenadas, de forma a mostrar as grandes potencialidades da metodologia. O procedimento usado por eles foi transformar as equações escritas no sistema de coordenadas do plano físico (geometria real do problema) em equações escritas nas coordenadas do plano transformado. As equações transformadas foram, então, resolvidas neste plano transformado. Aplicaram a metodologia em problemas de aerodinâmica bidimensional, usando o MDF.

Vários estudos encontrados na literatura dizem respeito ao método *multigrid* em problemas usando sistemas de coordenadas curvilíneas. Porém, não foram encontrados estudos sistemáticos para obtenção do número ótimo de iterações internas do *solver* e o número ótimo de níveis de malhas a ser utilizado para o método *multigrid* geométrico, usando sistemas de coordenadas curvilíneas não ortogonais, apesar do valor empregado para estes parâmetros estarem disponíveis em alguns trabalhos.

Com o intuito de investigar o desempenho do *multigrid*, Joshi e Vanka (1991) apresentaram algoritmos para problemas de escoamentos laminares e turbulentos bidimensionais de fluidos incompressíveis em regime estacionário, usando sistema de coordenadas curvilíneas não ortogonal. Usaram o FAS-FMG e ciclo V com uma iteração interna do *solver* em malhas com até 96×96 incógnitas, para escoamento laminar, e duas iterações internas do *solver* em malhas com até 80×40 incógnitas, para escoamento turbulento. O *solver* usado foi uma versão ponto-a-ponto do método Gauss-Seidel simétrico (do inglês, *Symmetric Coupled Gauss-Seidel* - SCGS). Concluíram que o algoritmo proposto tem o potencial de calcular eficientemente as soluções para problemas com diferentes tipos de geometrias e para escoamentos laminares e turbulentos.

Visando reduzir o tempo de CPU, Rayner (1991) apresentou um algoritmo *multigrid* para resolver problemas de escoamento de fluidos e transferência de calor que ocorrem no compressor de motores baseados em turbinas a gás. Esses problemas são modelados por

um sistema de equações formado pelas equações de conservação da massa e de conservação da Quantidade de Movimento Linear (QML). Utilizou sistema de coordenadas curvilíneas não ortogonal, gerado usando equações de Laplace transformadas, sendo as malhas não uniformes no domínio computacional retangular e as equações discretizadas usando o MDF. Implementou o *multigrid* com FAS e ciclo V, em malhas com 33×17 , 65×33 e 129×65 nós, usando duas iterações internas para o *solver* nas malhas mais finas e seis na malha mais grossa. Usou o número de níveis de malhas igual a $L_{\max} - 2$ em todas as malhas, sendo L_{\max} o número máximo de níveis de malha. Comparou o tempo de CPU do *multigrid* para sistemas de coordenadas curvilíneas com o *multigrid* para sistemas de coordenadas cartesianas. Verificou que os tempos de CPU são similares, embora comparações diretas sejam difíceis, pois as malhas cartesianas possuem menos nós. Além disso, comparou o tempo de CPU do *multigrid*, usando sistemas de coordenadas curvilíneas com o do *singlegrid* e verificou que o tempo de CPU do *multigrid* é cerca de 93% menor.

Também com o intuito de investigar o desempenho do *multigrid*, Smith, Cope e Vanka (1993) resolveram problemas tridimensionais modelados pelas equações de Navier-Stokes, empregando malhas curvilíneas não ortogonais geradas usando equações elípticas. As equações foram discretizadas usando o MVF e arranjo colocalizado. Apresentaram resultados para números de Reynolds iguais a 500, 1000 e 2000. Usaram o método *multigrid* com FAS combinado com FMG, ciclo V e com uma iteração interna para o *solver*. Calcularam as métricas de transformação de coordenadas na malha grossa restringindo as métricas de transformação da malha mais fina. Apresentaram as taxas de convergência do método para alguns tamanhos de malha, sendo a mais fina $120 \times 64 \times 40$ volumes e compararam a convergência dos algoritmos *multigrid* e *singlegrid* (método de malha única). Verificaram que apesar do custo computacional adicional exigido na restrição das métricas da transformação, a convergência mais rápida do método *multigrid* em relação ao *singlegrid* resultou em um tempo de CPU geral reduzido.

Oosterlee et al. (1993) encontraram soluções *benchmark* para as equações de Navier-Stokes para escoamento bidimensional de fluido incompressível em regime estacionário, em um domínio no formato de L, usando malhas curvilíneas não ortogonais. As equações foram discretizadas com o MVF, usando arranjo desencontrado das variáveis e componentes contravariantes do vetor velocidade. Para isso, empregaram o *multigrid*, com FAS, ciclo F, uma suavização na pré e na pós-suavização e com o número máximo de níveis em malhas com até 512×512 volumes. Apresentaram resultados para números de Reynolds iguais a 100 e 1000. Na malha 256×256 , obtiveram fatores de convergência médio empírico iguais a 0,204 para Reynolds igual a 100 e 0,588 para Reynolds igual a 1000, ou seja, obtiveram melhor fator de convergência médio empírico para número de Reynolds mais baixo. Concluíram que a versão linha do *solver* Gauss-Seidel simétrico (do inglês, *Symmetric Coupled Alternating Lines* - SCAL) é mais robusta que sua versão ponto-a-ponto.

He et al. (1996), visando reduzir o tempo de CPU, propuseram um algoritmo *multigrid* para as equações de Navier-Stokes para fluido newtoniano em escoamento incompressível em regime estacionário. Optaram por não fazer a transformação de coordenadas nas equações de Navier-Stokes. Usaram arranjo desencontrado de variáveis, o *solver* Gauss Seidel por linha e componentes covariantes para o vetor velocidade. Nesse caso, as integrações foram sobre volumes irregulares no domínio físico. Verificaram que a taxa de convergência do método pode melhorar significativamente dependendo do domínio, do tamanho da malha, da distribuição de linhas e do número de Reynolds.

No presente estudo, foram encontrados vários trabalhos entre 1996 e 2015 onde o método *multigrid* foi empregado na resolução de problemas usando sistemas de coordenadas curvilíneas. Entretanto, o foco de tais estudos não foi o estudo sistemático de parâmetros do *multigrid* em malhas curvilíneas, mas apresentar estudos teóricos e/ou testar novas metodologias numéricas.

Li, Yu e Wang (2015) apresentaram resultados numéricos mais precisos que os apresentados em Demirdžić, Lilek e Perić (1992), para o problema da cavidade com formato de paralelogramo, com tampa móvel e números de Reynolds iguais a 100 e 1000. Para isso, estudaram a equação de Convecção-Difusão para escoamento bidimensional de fluido incompressível em regime transiente em cavidades no formato de paralelogramo, trapézio e tipo seno, discretizada pelo MVF, em malhas com até 1024×1024 volumes. Usaram o método *multigrid* com FAS, ciclo V e o *solver Strong Implicit Procedure* (SIP), com 4 iterações internas do *solver* para calcular as componentes do vetor velocidade e 12 iterações para resolver a equação para a pressão. De acordo com Li, Yu e Wang (2015), ainda são poucos os problemas *benchmark* para domínios irregulares discretizados com malhas curvilíneas coincidentes com a fronteira do domínio. A maioria dos problemas *benchmark* ainda empregam sistemas de coordenadas cilíndricas ou polares.

1.1.3 Equação de Burgers

Durante as últimas décadas, consideráveis esforços têm sido feitos para o desenvolvimento de procedimentos computacionais robustos para resolver equações diferenciais parciais não lineares encontradas em várias áreas da Engenharia e vários campos das ciências da natureza. A equação de Burgers foi introduzida em 1915 por Harry Bateman. Mais tarde, em 1948, devido ao extensivo trabalho de Johannes Martinus Burgers, que apresentou o modelo matemático que descreve a turbulência e investigou aspectos espectrais e estatísticos da equação e sistemas de equações relacionados, tal equação ficou conhecida popularmente como equação de Burgers (BONKILE et al., 2018). Essas equações são as mais comuns que envolvem efeitos não lineares e difusivos.

As equações de Burgers descrevem vários problemas importantes da Mecânica dos Fluidos e são amplamente usadas em várias aplicações, tais como, modelagem de dinâmica

de gases, descrição de propagação de ondas em acústica e hidrodinâmica, entre outros. São equações diferenciais parciais não lineares, formuladas a partir de princípios de conservação, que combinam termos advectivos e difusivos. As equações de Burgers são consideradas formas simplificadas das equações de Navier-Stokes, reduzidas às equações de conservação da Quantidade de Movimento Linear (QML). De acordo com certas condições, podem ser classificadas como hiperbólicas, parabólicas ou elípticas. Sua ampla aplicabilidade tem levado muitos pesquisadores ao desenvolvimento de diversos esquemas numéricos para encontrar soluções para as equações de Burgers, sendo ainda um campo de pesquisa bastante ativo (DHAWAN et al., 2012; BONKILE et al., 2018).

No presente estudo, observou-se a existência de uma vasta literatura disponível com diferentes técnicas utilizadas para resolver as equações de Burgers. Nerney, Schmahl e Musielak (1996) encontraram a solução das equações de Burgers tridimensional, usando sistema de coordenadas curvilíneas ortogonal e a transformação de Cole-Hopf. A principal vantagem dessa abordagem é que pode ser usada em diversas aplicações dessas equações.

Foram encontrados vários trabalhos baseados na transformação de Hopf-Cole, na qual as equações de Burgers são reduzidas à equação linear do calor (ZHAO; YU; ZHANG, 2011; KANNAN; WANG, 2012; ZHANLAV; CHULUUNBAATAR; ULZIIBAYAR, 2015). Mas, existe uma variedade de métodos para resolver as equações de Burgers. Alguns exemplos são o método variacional (AKSAN; ÖZDEŞ, 2004), Lattice Boltzmann (LIU; WEIPING, 2011) e esquema de diferenças finitas tipo Crank-Nicolson (XU; SUN, 2009). Dhawan et al. (2012) e Bonkile et al. (2018) apresentaram uma extensa e detalhada revisão das diferentes técnicas empregadas na resolução das equações de Burgers. No trabalho de Bonkile et al. (2018) estão as principais referências acerca do estudo destas equações até os dias atuais, acompanhadas de um resumo das análises e dos importantes resultados e desenvolvimentos obtidos.

Foram encontrados também alguns trabalhos que utilizam o método *multigrid* na resolução de problemas que envolvem as equações de Burgers. Entretanto, em sua maioria, o foco de tais estudos não é a redução do tempo de CPU ou o estudo sistemático de parâmetros do *multigrid*, mas apresentar aplicações, estudos teóricos e/ou testar novas metodologias numéricas.

Mulder (1990) investigou a convergência do método *multigrid* para a equação unidimensional de Burgers para escoamento invíscido. Realizou uma análise teórica e apresentou resultados numéricos com o emprego de um algoritmo FAS-FMG com ciclo F, para duas malhas, e comparou com um problema linear unidimensional, a equação da convecção com coeficientes constantes.

Outro trabalho acerca da convergência do método *multigrid* sob o ponto de vista teórico foi desenvolvido por Ibraheem e Demuren (1996). Propuseram o estudo da

convergência teórica baseado na LFA, para algoritmos de duas malhas para os modelos unidimensionais das equações de advecção-difusão e de Burgers linearizadas. Apresentaram alguns resultados numéricos usando um algoritmo FAS com duas malhas, ciclo V e os solvers LU e ADI, que comprovam a superioridade das previsões realizadas para o método proposto para esses modelos.

Implementaram o método *multigrid* com FAS, ciclo V e os solvers LU e ADI, para duas malhas.

Persson et al. (1999) apresentaram dois teoremas acerca da convergência do método *multigrid*. O primeiro mostra a convergência do *multigrid* com ciclo V, para um problema bidimensional modelado pela equação da advecção-difusão com coeficientes constantes, dominado pela advecção. O segundo teorema prova a convergência do *multigrid* com ciclo V, para algoritmos de duas malhas, para problemas de ondas de choque, descritos pelo modelo bidimensional das equações de Burgers.

Foram encontrados alguns trabalhos envolvendo as equações de Burgers que objetivam melhorar a convergência do método *multigrid*. Os trabalhos encontrados não são para malhas curvilíneas não ortogonais, mas para sistemas de coordenadas cartesianas.

Ferm e Lötstedt (1997) propuseram operadores de restrição para o resíduo para acelerar a convergência do método *multigrid* para problemas com ondas de choque. Um algoritmo FAS foi testado para as equações uni e bidimensional de Burgers e de Euler para duas malhas, e também para três malhas no caso do problema modelado pela equação de Euler. Os resultados mostraram que a taxa de convergência triplicou para a equação de Burgers e dobrou para a equação de Euler, usando três malhas. Confirmando, assim, a superioridade dos operadores de restrição propostos sobre o operador de restrição padrão, obtido tomando-se uma determinada média ponderada dos resíduos relativos aos volumes de controle da malha fina que correspondem àquele da malha grossa.

Visando reduzir o custo computacional, Zhang, Zhang e Xi (2010) propuseram algoritmos *multigrid* com 2 e 3 malhas. Implementaram métodos com malha única (*single-grid* - SG) e *multigrid* com FAS e ciclo V e o FAS-FMG. Avaliaram a precisão e a eficiência dos métodos através das equações unidimensional de Burgers, Taylor-vórtice bidimensional e o problema bidimensional da cavidade quadrada. Concluíram que o algoritmo FAS-FMG com 3 malhas é o que apresentou melhor desempenho, com *speedup*¹ de duas ordens de magnitude e pelo menos uma ordem de magnitude para 2 malhas. Além disso, mostraram que com o aumento do número de Reynolds, de 400 para 1000, a estimativa inicial afetou significativamente a taxa de convergência.

Neveu, Debreu e Dimet (2011) aplicaram o *multigrid* em um problema de assimi-

¹ Medida utilizada para determinar o aumento de velocidade obtido durante a execução de um programa utilizando um algoritmo “A” em relação a sua execução utilizando um algoritmo “B” (TROTENBERG; OOSTERLEE; SCHULLER, 2001).

lação variacional de dados, que resulta na equação unidimensional de Burgers transiente com condições de contorno periódicas. Realizaram testes com um algoritmo FAS para duas malhas com razão de refino 3. Avaliaram a taxa de convergência do *multigrid* com o *singlegrid* e verificaram que o método *multigrid* é 40 vezes mais rápido que o método *singlegrid*.

Gonçalves (2013) analisou a influência de alguns parâmetros do método *multigrid* sobre o tempo de CPU para as equações de Laplace, de Advecção-Difusão e de Burgers, usando sistema de coordenadas cartesianas e discretizadas usando MVF. Para as equações de Burgers, verificou que as componentes algorítmicas que otimizam o desempenho do FAS-FMG foram o número de iterações internas dos *solvers* GS-Lex e GS-RB iguais a 2 ou 3 e o número de malhas igual a $L_{\max} - 1$ em todas as malhas investigadas. Gonçalves (2013) concluiu que a melhor escolha algorítmica do *multigrid* para as equações de Burgers constitui-se do esquema FAS-FMG, com *solver* GS-RB e operador de prolongação bilinear, para interpolar as soluções entre os níveis do FMG e correções da malha grossa para a malha fina no algoritmo do ciclo V.

Santiago, Marchi e Souza (2015) investigaram alguns parâmetros do *multigrid* para os modelos bidimensionais de Laplace, Navier, Burgers e Navier–Stokes, usando sistema de coordenadas cartesianas e equações discretizadas pelo MDF. Para as equações de Burgers, o conjunto de parâmetros que resultou no menor tempo de CPU foi o número de iterações internas do *solver* MSI igual a 5 e o número de malhas igual ao máximo para o tamanho de problema considerado. Observou-se também que o tempo de CPU do *multigrid* cresce à mesma taxa para os problemas com uma ou duas equações e concluiu-se que o número de equações envolvidas no modelo matemático não interfere no desempenho do *multigrid*. Além disso, concluiu-se que não é vantajoso aplicar método *multigrid* em apenas uma equação quando o modelo envolve duas, como no caso das equações de Burgers.

Visando reduzir o tempo computacioanal, Howse et al. (2019) propuseram um algoritmo *multigrid* com redução no tempo (MGRIT) para a equação hiperbólica unidimensional de Burgers. Implementaram um algoritmo FAS e ciclo F para duas malhas e mostraram os *speedups* para os métodos explícitos e implícitos. Segundo eles, o método desenvolvido é o primeiro método convergente com paralelização no tempo para a equação de Burgers com e sem choques.

Esta revisão bibliográfica embasa e justifica os objetivos desta tese e motiva, principalmente, a desenvolver parametrizações e códigos computacionais eficientes, unindo malhas curvilíneas estruturadas não ortogonais com o método *multigrid*.

1.2 MOTIVAÇÃO

Os parâmetros ótimos do *multigrid* podem ter uma forte influência na taxa de convergência do método. Segundo Trottenberg, Oosterlee e Schuller (2001), a eficiência do *multigrid* depende das escolhas de seus componentes, tais como *solver*, número de iterações do *solver* em cada ciclo, tipo de ciclo, operadores de restrição e prolongação. Segundo Ferziger e Peric (2002), no contexto geral do método *multigrid* muitos parâmetros podem ser escolhidos aleatoriamente, mas as escolhas interferem na convergência do método. Wallis (2008) afirma que a escolha do *solver*, estimativa inicial, número de malhas, dentre outras escolhas, são elementos importantes a serem analisados no uso do *multigrid*. Assim, uma boa combinação de parâmetros torna-se essencial na redução do tempo de CPU, o que justifica a importância do estudo e identificação de parâmetros ótimos do método *multigrid*.

As ótimas propriedades teóricas do método *multigrid* com relação à aceleração da convergência dos métodos iterativos têm motivado o desenvolvimento de vários trabalhos com o foco no estudo de parâmetros do método *multigrid*. Algumas análises de componentes do *multigrid* podem ser encontradas em Briggs, Henson e McCormick (2000), Trottenberg, Oosterlee e Schuller (2001), Ferziger e Peric (2002), Wesseling (2004), Mesquita e Lemos (2004), Wienands e Joppich (2004), Thekale et al. (2010), Gonçalves (2013), Santiago, Marchi e Souza (2015), Franco et al. (2018), Oliveira et al. (2018) e Rutz, Pinto e Gonçalves (2019).

A geometria do domínio de cálculo de muitos problemas em CFD são irregulares. O uso de malhas curvilíneas estruturadas não ortogonais é uma alternativa para essas geometrias (VERSTEEG; MALALASEKERA, 2007). Segundo Trottenberg, Oosterlee e Schuller (2001), os volumes distorcidos no domínio de cálculo causam anisotropia nas equações. Problemas com anisotropias frequentemente causam deteriorização na eficiência do *multigrid*, resultando em maior tempo de CPU. Por isso, buscaram-se *solvers* mais eficientes para resolver as equações de Poisson e Burgers usando malhas curvilíneas estruturadas não ortogonais. Não foram encontrados na literatura estudos sistemáticos de componentes do *multigrid* sobre o tempo de CPU utilizando essas malhas.

Para calcular as métricas de transformação nas malhas grossas duas abordagens estão disponíveis em Smith, Cope e Vanka (1993). Uma maneira é calcular as métricas da malha grossa a partir das coordenadas dos vértices dessa malha, outra forma é obter as métricas da malha grossa fazendo a restrição das métricas de transformação de coordenadas da malha fina adjacente. Na primeira abordagem, tem-se métricas de transformação que não resultam nas mesmas medidas para as faces e áreas dos volumes de controle que as medidas e áreas obtidas combinando as métricas dos volumes de controle da malha fina. Segundo Smith, Cope e Vanka (1993), a convergência do método *multigrid* melhora quando

as métricas nas malhas grossas são obtidas por meio da restrição dos valores da malha mais fina adjacente. Contudo, não apresentaram como calcular as métricas de transformação de coordenadas nas malhas mais grossas por meio da restrição dos valores da malha mais fina adjacente.

As equações de Burgers 2D são consideradas um modelo de referência no estudo de escoamento de fluidos. Devido à sua não linearidade e multidimensionalidade, as equações de Burgers 2D são uma boa referência, entre os problemas não lineares, para testes de metodologias numéricas (EFE, 2006). Estas equações possuem muitas aplicações, desde a cosmologia até a modelagem do tráfego (ZHENG, 2016). Dentre os trabalhos encontrados que envolvem o método *multigrid* aplicado às equações de Burgers, são poucos os que tem como objetivo melhorar a convergência do método *multigrid*. Ferm e Lötstedt (1997) analisaram diferentes operadores de restrição. Gonçalves (2013) analisou o efeito do número de iterações internas do *solver*, do número de malhas, do número de incógnitas e diferentes tipos de operadores de prolongação sobre o tempo de CPU. Santiago, Marchi e Souza (2015) analisaram o efeito do número de iterações internas do *solver*, do número de malhas e do número de incógnitas sobre o tempo de CPU. Observa-se, no entanto, que os trabalhos encontrados na literatura não são para malhas curvilíneas estruturadas não ortogonais.

1.3 OBJETIVOS

O objetivo geral deste trabalho é analisar alguns parâmetros do método *multigrid* geométrico para as equações de Poisson e de Burgers, usando malhas curvilíneas estruturadas não ortogonais.

Os parâmetros investigados para fins de redução do tempo de CPU são o número de iterações internas do *solver*, o número de níveis de malhas e o número de incógnitas. Os *solvers* estudados a fim de escolher um *solver* adequado à direção da anisotropia são GS-Lex, η -linha-GS, MSI e MILU.

Para isso, os objetivos específicos são

- Comparar o fator de convergência médio empírico dos *solvers* GS-Lex, η -linha-GS, MSI e MILU para as equações de Poisson e de Burgers;
- Verificar a influência do número de iterações do *solver* sobre o tempo de CPU e identificar o valor ótimo para este parâmetro;
- Verificar a influência do número de níveis sobre o tempo de CPU e identificar o valor ótimo para este parâmetro;

- Verificar a influência do número de incógnitas sobre o tempo de CPU e analisar a eficiência do código implementado para malhas curvilíneas estruturadas não ortogonais;
- Testar as malhas geradas usando interpolação de Lagrange e equações elípticas juntamente com os *solvers* com melhor fator de convergência médio empírico e identificar qual combinação resulta no melhor desempenho do método *multigrid*.

1.4 ORGANIZAÇÃO DA TESE

Além deste primeiro capítulo, esta tese está distribuída em mais seis capítulos e um apêndice. No capítulo 2 são apresentados fundamentos teóricos a respeito do método dos volumes finitos, dos métodos iterativos para resolução de sistemas, da transformação de coordenadas e da geração de coordenadas curvilíneas. O capítulo 3 contém fundamentos teóricos a respeito do método *multigrid*, incluindo componentes do *multigrid* e algoritmo. No capítulo 4 são apresentados os modelos matemáticos, bem como um detalhamento do modelo numérico e do processo de obtenção das equações discretizadas. Alguns aspectos relevantes de implementação do método *multigrid* são apresentados no capítulo 5. Os resultados, comparações e discussões estão descritos no capítulo 6 e as conclusões finais e as sugestões para trabalhos futuros estão descritas no capítulo 7. As soluções numéricas são verificadas comparando-se com soluções analíticas no Apêndice A.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo será descrito o referencial teórico usado nesta tese. Apresenta-se o Método dos Volumes Finitos (MVF) usado para discretizar as equações diferenciais, formas de aplicação das condições de contorno, métodos de resolução de sistemas lineares, o processo de transformação entre o sistema de coordenadas cartesianas e o sistema de coordenadas curvilíneas, tipos de mapeamento de geometrias e métodos de geração do sistema de coordenadas curvilíneas.

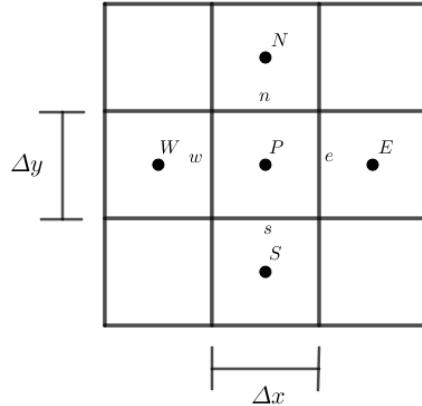
2.1 O MÉTODO DOS VOLUMES FINITOS

A solução numérica de equações ou sistemas de equações diferenciais requer a discretização do domínio de cálculo. Essa discretização gera uma malha em que os termos da equação diferencial são aproximados, resultando em um sistema de equações algébricas que envolve as funções incógnitas. Em CFD, os métodos mais empregados na discretização das equações diferenciais são o MDF, MEF e o MVF (VERSTEEG; MALALASEKERA, 2007; MALISKA, 2010).

No MVF, o domínio de cálculo é dividido em um número finito de subdomínios, denominados volumes de controle, os quais envolvem um único ponto nodal da malha, onde os valores da variável de interesse são calculados (PATANKAR, 1980). Nesse método as equações diferenciais são integradas sobre cada volume de controle (VC) e funções de interpolação em termos dos valores nodais são utilizadas para obter os valores das variáveis nas faces do VC (PATANKAR, 1980). Segundo Patankar (1980), no caso de equações diferenciais escalares, em que se tem apenas uma variável de interesse, um VC pode ser construído com os nós centrados entre as faces do volume ou com as faces centradas entre os nós. Para o caso de equações diferenciais vetoriais, em que se tem mais de uma variável de interesse, há a possibilidade de se posicionar todas as variáveis no centro do volume (este caso é denominado arranjo colocalizado) ou posicionar algumas variáveis no centro do volume e outras em suas faces (este caso é denominado arranjo desencontrado).

A FIGURA 2 exemplifica a disposição dos volumes de controle em uma malha bidimensional uniforme. Nessa figura as letras maiúsculas P , N , S , W e E representam os centros do volume de controle genérico e seus vizinhos norte, sul, leste e oeste, respectivamente, e as letras minúsculas n , s , e e w representam as faces norte, sul, leste e oeste do VC, respectivamente.

FIGURA 2 – DISPOSIÇÃO DE UM VOLUME DE CONTROLE P E SEUS VIZINHOS N (NORTE), S (SUL), E (LESTE) E W (OESTE), EM UMA MALHA BIDIMENSIONAL UNIFORME



Para ilustrar a discretização da equação diferencial parcial, considera-se, por exemplo, a equação que expressa a advecção-difusão de uma determinada propriedade ϕ , em regime permanente, dada por

$$\nabla \cdot (\rho \mathbf{v} \phi) = \nabla \cdot (\Gamma \nabla \phi) + S_\phi, \quad (2.1)$$

em que o termo do lado esquerdo refere-se à advecção da propriedade ϕ , o primeiro termo do lado direito refere-se à difusão e o segundo termo do lado direito é o termo fonte de cada ϕ . Os escalares ρ e Γ referem-se à massa específica do fluido e a um coeficiente de transporte, respectivamente. O vetor velocidade é referenciado por \mathbf{v} e o operador ∇ (nabla) indica o gradiente de ϕ quando assume a operação $\nabla \phi$ e o divergente do vetor velocidade quando assume a operação $\nabla \cdot \mathbf{v}$.

Integrando a equação (2.1) sobre cada VC, tem-se

$$\iiint_{VC} \nabla \cdot (\rho \mathbf{v} \phi) \, dV = \iiint_{VC} \nabla \cdot (\Gamma \nabla \phi) \, dV + \iiint_{VC} S_\phi \, dV, \quad (2.2)$$

sendo dV o elemento de volume do volume de controle P .

Aplicando o teorema da Divergência de Gauss (KREYSZIG, 2009) à equação (2.2), obtém-se

$$\iint_A (\rho \mathbf{v} \phi) \cdot \mathbf{n} \, dA = \iint_A (\Gamma \nabla \phi) \cdot \mathbf{n} \, dA + \iiint_{VC} S_\phi \, dV, \quad (2.3)$$

sendo dA o elemento de área da superfície do volume de controle P e \mathbf{n} o vetor unitário normal à superfície do elemento dA .

Calculando as integrais da equação (2.3), obtém-se através das faces do VC uma expressão contendo as taxas de transferência advectiva e difusiva da propriedade ϕ . A

avaliação da propriedade ϕ e de suas taxas nas interfaces do volume de controle, em função do valor das propriedades armazenadas nos centros dos volumes, é feita mediante o uso de funções de interpolação ou esquemas numéricos. Em Maliska (2010) encontram-se algumas funções de interpolação unidimensionais, tais como os esquemas CDS (*Central Differencing Scheme*), UDS (*Upwind Differencing Scheme*), Esquema Exponencial e WUDS (*Weighted Upstream Differencing Scheme*) e bidimensionais, tais como o SUDS (*Skew Upstream Differencing Scheme*) e o SWUDS (*Skew Weighted Upstream Differencing Scheme*).

Para aproximar as derivadas da propriedade ϕ nas faces w , e , s e n do volume de controle P , pode-se empregar o CDS. Assim,

$$\left[\frac{\partial \phi}{\partial x} \right]_w = \frac{\phi_P - \phi_W}{\Delta x},$$

$$\left[\frac{\partial \phi}{\partial x} \right]_e = \frac{\phi_E - \phi_P}{\Delta x},$$

$$\left[\frac{\partial \phi}{\partial y} \right]_s = \frac{\phi_P - \phi_S}{\Delta y}$$

e

$$\left[\frac{\partial \phi}{\partial y} \right]_n = \frac{\phi_N - \phi_P}{\Delta y},$$

sendo ϕ_P , ϕ_W , ϕ_E , ϕ_S e ϕ_N a propriedade ϕ avaliada nos volumes de controle P , W , E , S e N , respectivamente.

Para avaliar a propriedade ϕ nas faces w , e , s e n do volume de controle P , pode-se empregar o UDS que considera a direção do escoamento. Se $\phi > 0$, ϕ_w , ϕ_e , ϕ_s e ϕ_n passam a serem avaliadas como

$$\phi_w = \phi_W,$$

$$\phi_e = \phi_P,$$

$$\phi_s = \phi_S$$

e

$$\phi_n = \phi_P.$$

No caso de $\phi < 0$, tem-se

$$\phi_w = \phi_P,$$

$$\phi_e = \phi_E,$$

$$\phi_s = \phi_P$$

e

$$\phi_n = \phi_N.$$

Empregando a correção adiada (LILEK; MUZAFERIJA; PERIĆ, 1997; MALISKA, 2010) é possível trabalhar de forma implícita com uma formulação UDS promovendo a correção para um esquema de 2 λ ordem explicitamente. A correção adiada é dada pela seguinte expressão

$$\phi_f = \phi_f^{UDS} + (\phi_f^{CDS})^* - (\phi_f^{UDS})^*, \quad (2.4)$$

sendo f a face do VC, os sobescritos UDS e CDS denotam os esquemas *upwind* e diferenças centrais, respectivamente, e $*$ indica as condições conhecidas do nível iterativo anterior. Na convergência, as avaliações UDS cancelam-se e o CDS é o efetivamente utilizado, resultando em uma aproximação de 2 λ ordem. Na equação (2.4) outros esquemas de primeira ordem e alta ordem podem substituir os esquemas UDS e CDS, respectivamente.

Calculando a integral, na equação (2.3), e aplicando as funções de interpolação para ϕ e suas derivadas nas faces do VC, tem-se para cada propriedade ϕ e para cada VC, a seguinte equação algébrica, de forma mais compacta,

$$a_p \phi_P = a_w \phi_W + a_e \phi_E + a_s \phi_S + a_n \phi_N + b_p, \quad (2.5)$$

sendo a_w , a_e , a_s e a_n os coeficientes obtidos com a aplicação das funções de interpolação e b_p o termo fonte.

Segundo Ferziger e Peric (2002), o MVF pode ser aplicado a qualquer tipo de malha, em geometrias complexas e em diferentes sistemas de coordenadas. A característica atraente do MVF é que o método satisfaz à conservação integral das grandezas, tais como massa, quantidade de movimento linear e energia, por exemplo, para qualquer grupo de volumes de controle e, por consequência, para todo o domínio de cálculo (PATANKAR, 1980).

2.2 CONDIÇÕES DE CONTORNO

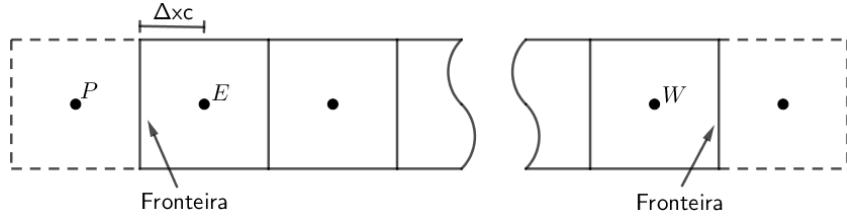
A equação (2.5) é a equação aproximada para um volume de controle interno genérico. Todos os outros volumes internos possuem equações aproximadas idênticas. Para obter o sistema de equações algébricas completo, é necessário também obter as equações algébricas para os volumes de controle que estão na fronteira.

Existem várias formas de aplicação das condições de contorno, tais como usando volumes fictícios, balanço para os volumes da fronteira ou meio-volume (MALISKA, 2010). A seguir, apresenta-se apenas a técnica de volumes fictícios, técnica que será utilizada no decorrer deste trabalho. No final desta subseção apresenta-se a justificativa (vantagens e desvantagens) para a escolha desta técnica.

A técnica dos volumes fictícios consiste em adicionar volumes auxiliares não pertencentes ao domínio. Apenas para ilustrar, na FIGURA 3, os volumes fictícios são mostrados em linhas tracejadas, pois não possuem posição geométrica na malha. Uma vez

criados os volumes fictícios, todos os volumes do domínio, inclusive os de fronteira, são interpretados como internos.

FIGURA 3 – CONDIÇÕES DE CONTORNOS COM VOLUMES FICTÍCIOS



FONTE: Adaptado de Maliska (2010).

Com a criação dos volumes fictícios, geram-se equações para esses volumes em função das condições de contorno existentes. Como, por exemplo, para condição de contorno de Dirichlet, a propriedade ϕ na fronteira, ϕ_c , é conhecida e pode ser escrita como

$$\phi_c = \frac{\phi_P + \phi_E}{2}. \quad (2.6)$$

Reescrevendo a equação (2.6), obtém-se

$$\phi_P = -\phi_E + 2\phi_c. \quad (2.7)$$

Comparando a equação (2.7) com a equação (2.5) determinam-se os seguintes coeficientes e o termo fonte do volume fictício

$$a_p = 1,$$

$$a_e = -1,$$

$$a_w = a_s = a_n = 0$$

e

$$b_p = 2\phi_c.$$

Essa técnica é de fácil aplicação, sendo todos os volumes do domínio de cálculo (incluindo os de fronteira) volumes inteiros, respeitando, portanto, os princípios de conservação para todo o domínio. Além disso, essa técnica também é indicada para problemas com condições de contorno do tipo Dirichlet e que utilizam malhas curvilíneas e transformação de coordenadas. Isso porque os volumes fictícios vizinhos ao volume fictício P não influenciam na equação algébrica correspondente a esse volume (MALISKA, 2010). Uma desvantagem é o aumento do número de incógnitas do sistema algébrico.

2.3 MÉTODOS ITERATIVOS DE RESOLUÇÃO PARA SISTEMAS DE EQUAÇÕES LINEARES

Obtido o sistema de equações algébricas completo, emprega-se um método iterativo para obter a solução do sistema de equações algébricas dado pela equação (1.1), a partir

da estimativa inicial. A partir da estimativa inicial, o método iterativo, gera uma sequência de vetores $\phi^{(1)}, \phi^{(2)}, \dots, \phi^{(k)}$. Se essa sequência satisfaz

$$\lim_{k \rightarrow \infty} \phi^{(k)} = \phi,$$

então o método iterativo converge para a solução do sistema de equações, caso ela exista (FORTUNA, 2000). A seguir, descreve-se como obter a expressão para determinar essa sequência de vetores.

No sistema de equações algébricas dado pela equação (1.1), a matriz dos coeficientes pode ser escrita como

$$\mathbf{A} = \mathbf{M} - \mathbf{N}, \quad (2.8)$$

sendo \mathbf{M} inversível. Substituindo a equação (2.8) na equação (1.1) e criando o processo iterativo, obtém-se

$$\mathbf{M}\phi^{(k+1)} = \mathbf{N}\phi^{(k)} + \mathbf{b}, \quad (2.9)$$

sendo k e $k + 1$ as iteradas.

A solução do sistema de equações algébricas dado pela equação (1.1) é obtida fazendo-se

$$\phi^{(k+1)} = \mathbf{T}\phi^{(k)} + \mathbf{M}^{-1}\mathbf{b}, \quad (2.10)$$

sendo $\mathbf{T} = \mathbf{M}^{-1}\mathbf{N}$ a matriz de iteração do método iterativo e $k = 0, 1, 2, \dots$ é a iteração. A matriz \mathbf{T} depende do método iterativo utilizado. Os métodos iterativos convergem para a solução do sistema de equações, caso ela exista, se certas condições impostas sobre a matriz dos coeficientes do sistema linear são satisfeitas (BURDEN; FAIRES, 2016). O processo iterativo é terminado quando a solução atende a certo critério de parada previamente estabelecido, que pode ser baseado em diversos critérios, entre eles, baseado no erro, no resíduo ou no número de iterações.

O procedimento descrito pelas equações (2.9) e (2.10) é conhecido como método iterativo básico ou estacionário, porque é a base para o desenvolvimento de um *solver* iterativo. Alguns exemplos de *solvers* iterativos são Gauss-Seidel, Jacobi, SOR, ILU, MILU e MSI. Detalhes sobre esses e outros métodos podem ser encontrados em Fortuna (2000), Briggs, Henson e McCormick (2000) e Burden e Faires (2016).

Nas próximas subseções serão tratados alguns métodos iterativos de interesse nesta tese.

2.3.1 Método Gauss-Seidel e suas variantes

O método de Gauss-Seidel (SAAD, 2003; BURDEN; FAIRES, 2016) é um método iterativo que pode ser usado para resolver um sistema de equações dado pela equação (1.1). Este método usa em uma mesma iteração os valores das variáveis que já foram calculadas nesta iteração.

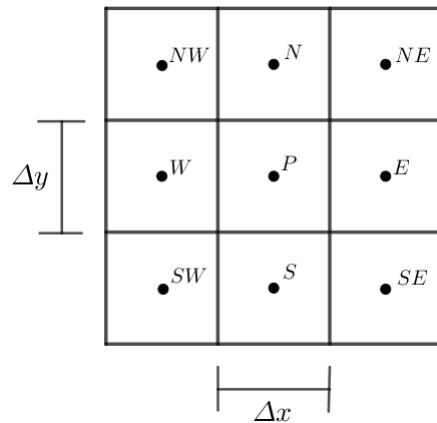
O método de Gauss-Seidel pode ser classificado como suavizador ponto-a-ponto ou por blocos, como, por exemplo, linha-a-linha. A diferença entre cada suavizador é a maneira como as incógnitas são atualizadas.

Nos métodos ponto-a-ponto, cada variável é atualizada individualmente. A FIGURA 4 apresenta a ordenação lexicográfica dos pontos usada neste trabalho. Ao utilizar o método de Gauss-Seidel com esta ordenação, tem-se o método de Gauss-Seidel lexicográfico (GS-Lex). Um recorte de uma malha bidimensional com a disposição de 9 volumes é apresentado na FIGURA 5, sendo P um volume de controle e seus vizinhos N (norte), S (sul), E (leste), W (oeste), SW (sudoeste), SE (sudeste), NW (nordeste) e NE (nordeste).

FIGURA 4 – ORDENAÇÃO LEXICOGRÁFICA EM UMA MALHA BIDIMENSIONAL

• 31	• 32	• 33	• 34	• 35	• 36
• 25	• 26	• 27	• 28	• 29	• 30
• 19	• 20	• 21	• 22	• 23	• 24
• 13	• 14	• 15	• 16	• 17	• 18
• 7	• 8	• 9	• 10	• 11	• 12
• 1	• 2	• 3	• 4	• 5	• 6

FIGURA 5 – NOVE VOLUMES EM UMA MALHA BIDIMENSIONAL UNIFORME



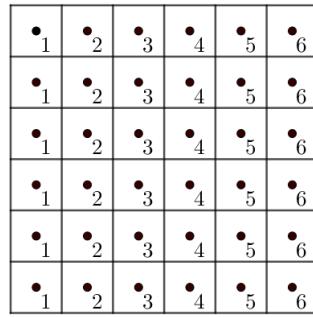
Para resolver cada equação do sistema de equações algébricas dado pela equação (1.1), usando um esquema de 9 volumes e o método GS-Lex, reescrevem-se as equações desse sistema da seguinte maneira

$$\phi_P^{(k+1)} = \frac{\left(\sum_{m_1} a_{m_1} \phi_{m_1}^{(k+1)} + \sum_{m_2} a_{m_2} \phi_{m_2}^{(k)} + b_P \right)}{a_P},$$

sendo o superíndice k a k -ésima iteração e o subíndice a posição na malha, $m_1 = W, S, SW, SE$ e $m_2 = E, N, NW, NE$. Note que na iteração k , utilizando-se a ordenação lexicográfica, são conhecidas as variáveis $\phi_W, \phi_S, \phi_{SW}$ e ϕ_{SE} .

Nos métodos por blocos, cada bloco pode conter as informações, por exemplo, de uma coluna ou linha inteira, dos pontos da malha a serem resolvidos. Cada bloco é atualizado de uma única vez. A FIGURA 6 apresenta a ordenação dos pontos da malha por linha na direção y . Ao utilizar o método de Gauss-Seidel com esta ordenação, tem-se o método de Gauss-Seidel linha na direção y (y -linha-GS).

FIGURA 6 – ORDENAÇÃO DOS PONTOS DA MALHA POR LINHA NA DIREÇÃO Y

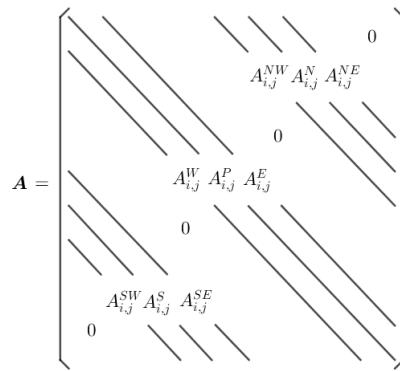


Quando se tem problemas anisotrópicos, a convergência de métodos básicos, como os suavizadores ponto-a-ponto, piora (OOSTERLEE et al., 1993; WIENANDS; JOPPICH, 2005; OLIVEIRA et al., 2018). Uma possível solução para esses problemas é utilizar suavizadores por blocos, porque as incógnitas do forte acoplamento são atualizadas coletivamente, tornando o erro suave.

2.3.2 Decomposição LU incompleta modificada (MILU)

A decomposição LU convencional de uma matriz esparsa \mathbf{A} gera duas matrizes triangulares \mathbf{L} e \mathbf{U} (BURDEN; FAIRES, 2016). Como por exemplo, a FIGURA 7 apresenta a estrutura de uma matriz esparsa \mathbf{A} , sendo $A_{i,j}^{SW} = a_{i,i-N_x-1}$, $A_{i,j}^S = a_{i,i-N_x}$, $A_{i,j}^{SE} = a_{i,i-N_x+1}$, $A_{i,j}^W = a_{i,i-1}$, $A_{i,j}^P = a_{i,i}$, $A_{i,j}^E = a_{i,i+1}$, $A_{i,j}^{NW} = a_{i,i+N_x-1}$, $A_{i,j}^N = a_{i,i+N_x}$, $A_{i,j}^{NE} = a_{i,i+N_x+1}$, em que N_x é o número de volumes na direção x .

FIGURA 7 – ESTRUTURA DA MATRIZ A



A FIGURA 8 apresenta a estrutura das matrizes triangulares \mathbf{L} e \mathbf{U} , obtidas da decomposição convencional LU da matriz \mathbf{A} .

FIGURA 8 – ESTRUTURA DAS MATRIZES \mathbf{L} E \mathbf{U} COM A DECOMPOSIÇÃO LU CONVENTIONAL

(a) Estrutura da matriz \mathbf{L}

$$\mathbf{L} = \begin{bmatrix} & & & \\ & \diagdown & & \\ & & \diagdown & \\ & & & \ddots \\ & & & & \diagdown \\ & & & & & 0 \\ & & & & & & \end{bmatrix}$$

(b) Estrutura da matriz \mathbf{U}

$$\mathbf{U} = \begin{bmatrix} & & & & & \\ & \diagup & & & & \\ & & \diagup & & & \\ & & & \ddots & & \\ & & & & \diagup & \\ & & & & & \ddots \\ & & & & & & 0 \\ & & & & & & & \end{bmatrix}$$

Pode-se observar, na FIGURA 8, que as matrizes \mathbf{L} e \mathbf{U} não possuem a mesma esparsidade da matriz \mathbf{A} e, consequentemente, a decomposição \mathbf{LU} perde as boas propriedades de algoritmos especializados. Uma alternativa para minimizar este dano é usar a decomposição LU incompleta (ILU) (TROTTERBERG; OOSTERLEE; SCHULLER, 2001; WESSELING, 2004; SAAD, 2003). Usando ILU é possível aproximar as matrizes \mathbf{L} e \mathbf{U} com boa esparsidade.

A decomposição LU incompleta é obtida substituindo-se, primeiramente, a matriz \mathbf{M} da equação (2.8) pelo produto

$$\mathbf{M} = \mathbf{LU}. \quad (2.11)$$

Uma fatoração alternativa para \mathbf{M} é

$$\mathbf{M} = \mathbf{LD}^{-1}\mathbf{U}, \quad (2.12)$$

sendo \mathbf{D} uma matriz diagonal ou, ainda, pode-se inserir a diagonal de \mathbf{D} na matriz \mathbf{U} . Em seguida, reescreve-se a matriz \mathbf{A} , dada pela equação (2.8), da seguinte forma

$$\mathbf{A} = \mathbf{LU} - \mathbf{R},$$

sendo \mathbf{R} a matriz dos resíduos.

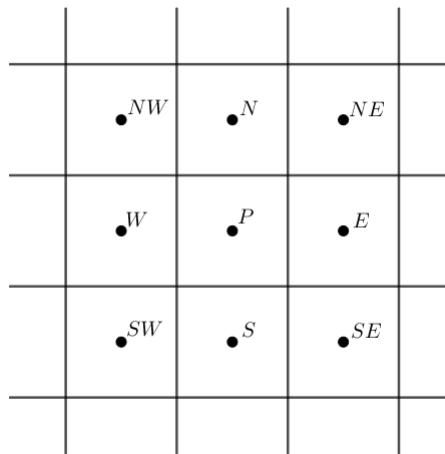
Usando ILU 9 pontos para o esquema com 9 volumes, que é o caso deste trabalho, é possível obter matrizes \mathbf{L} e \mathbf{U} com a mesma esparsidade de \mathbf{A} , introduzindo-se, para isso, a matriz \mathbf{R} (SAAD, 2003). Este caso é denominado de ILU(0). Caso seja adicionado uma diagonal a cada uma das matrizes \mathbf{L} e \mathbf{U} , portanto, dois pontos adicionais na discretização, tem-se ILU 11 pontos para uma matriz original com 9 diagonais. Nesse caso, as matrizes \mathbf{L} e \mathbf{U} são esparsas, mas não com a mesma esparsidade da matriz \mathbf{A} original. Este caso é denominado ILU(1), e assim por diante. A FIGURA 9 apresenta à decomposição usada neste trabalho (ILU(0) para uma matriz com 9 diagonais).

FIGURA 9 – ILU(0) PARA UMA MATRIZ COM 9 DIAGONAIS

$$\begin{bmatrix} & & & 0 \\ & & & 0 \\ & & & 0 \\ & & & 0 \\ & & & 0 \end{bmatrix} = \begin{bmatrix} & & & 0 \\ & & & 0 \\ & & & 0 \\ & & & 0 \\ & & & 0 \end{bmatrix} - \begin{bmatrix} & & & 0 \\ & & & 0 \\ & & & 0 \\ & & & 0 \\ & & & 0 \end{bmatrix}$$

Na FIGURA 9, a matriz \mathbf{R} contém quatro diagonais não nulas (linhas contínuas). As linhas tracejadas representam as posições originais das diagonais não nulas de \mathbf{A} . A FIGURA 10 ilustra a ordenação geográfica dos pontos na malha associada à ILU(0), para o esquema com 9 volumes.

FIGURA 10 – MALHA BIDIMENSIONAL DO PROCEDIMENTO ILU(0) PARA O ESQUEMA COM 9 VOLUMES



Por simplicação, de acordo com Wesseling (2004), denotaram-se os coeficientes não nulos da matriz de coeficientes \mathbf{A} , ilustrada na FIGURA 7, por $z_k = a_{k,k-N_x-1}$, $a_k = a_{k,k-N_x}$, $b_k = a_{k,k-N_x+1}$, $c_k = a_{k,k-1}$, $d_k = a_{k,k}$, $q_k = a_{k,k+1}$, $f_k = a_{k,k+N_x-1}$, $g_k = a_{k,k+N_x}$ $p_k = a_{k,k+N_x+1}$. Desta forma, pode-se reescrever a matriz dos coeficientes \mathbf{A} conforme ilustra a FIGURA 11.

FIGURA 11 – MATRIZ \mathbf{A} REESCRITA

$$\mathbf{A} = \begin{bmatrix} & & & f_k & g_k & p_k & 0 \\ & & & c_k & d_k & q_k & 0 \\ & & & 0 & & & 0 \\ z_k & a_k & b_k & & & & \end{bmatrix}$$

Os elementos não nulos das diagonais de \mathbf{L} , \mathbf{D} e \mathbf{U} são denotados por ω_k , α_k , β_k , γ_k , δ_k , μ_k , ζ_k , η_k e τ_k , respectivamente. A FIGURA 12 apresenta a estrutura das matrizes

L, D e U.

FIGURA 12 – MATRIZES L, D E U

$$(a) \text{ Matriz } \mathbf{L} = \begin{bmatrix} & & & 0 \\ & \gamma_k & \delta_k & & \\ & 0 & & & \\ & \omega_k & \alpha_k & \beta_k & \\ 0 & & & & \end{bmatrix}$$

$$(b) \text{ Matriz } \mathbf{D} = \begin{bmatrix} & & & 0 \\ & & \delta_k & & \\ & 0 & & & \\ & & & & \end{bmatrix}$$

$$(c) \text{ Matriz } \mathbf{U} = \begin{bmatrix} & & & \zeta_k & \eta_k & \tau_k & 0 \\ & \delta_k & \mu_k & & & & \\ & 0 & & & & & \\ & & & & & & \end{bmatrix}$$

Pode-se observar que fazendo a decomposição ILU(0) para a matriz \mathbf{A} , os elementos não nulos das matrizes \mathbf{L} e \mathbf{U} possuem a mesma localização que os elementos não nulos de \mathbf{A} .

Da igualdade $\mathbf{A} = \mathbf{LD}^{-1}\mathbf{U}$, tem-se a decomposição ILU 9 pontos para a matriz com 9 diagonais, na sua forma recursiva

$$\omega_k = z_k,$$

$$\alpha_k + \omega_k \delta_{k-N_x-1}^{-1} \mu_{k-N_x-1} = a_k,$$

$$\beta_k + \alpha_k \delta_{k-N_x}^{-1} \mu_{k-N_x} = b_k,$$

$$\gamma_k + \omega_k \delta_{k-N_x-1}^{-1} \eta_{k-N_x-1} + \alpha_k \delta_{k-N_x}^{-1} \zeta_{k-N_x} = c_k,$$

$$\delta_k + \omega_k \delta_{k-N_x-1}^{-1} \tau_{k-N_x-1} + \alpha_k \delta_{k-N_x}^{-1} \eta_{k-N_x} + \beta_k \delta_{k-N_x+1}^{-1} \zeta_{k-N_x+1} + \gamma_k \delta_{k-1}^{-1} \mu_{k-1} = d_k,$$

$$\mu_k + \alpha_k \delta_{k-N_x}^{-1} \tau_{k-N_x} + \beta_k \delta_{k-N_x+1}^{-1} \eta_{k-N_x+1} = q_k,$$

$$\zeta_k + \gamma_k \delta_{k-1}^{-1} \eta_{k-1} = f_k,$$

$$\eta_k + \gamma_k \delta_{k-1}^{-1} \tau_{k-1} = g_k,$$

$$\tau_k = p_k.$$

Dada a decomposição LU incompleta, $\mathbf{A} = \mathbf{LU} - \mathbf{R}$, tem-se

$$\mathbf{A}\phi = \mathbf{b} \Leftrightarrow (\mathbf{LU} - \mathbf{R})\phi = \mathbf{b} \Leftrightarrow \mathbf{LU}\phi - \mathbf{R}\phi = \mathbf{b}.$$

Com isso, cria-se o processo iterativo

$$\mathbf{LU}\phi^{(k+1)} = \mathbf{b} + \mathbf{R}\phi^{(k)}. \quad (2.13)$$

Substituindo $\mathbf{R} = \mathbf{LU} - \mathbf{A}$ na equação (2.13), tem-se

$$\mathbf{LU}\phi^{(k+1)} = \mathbf{b} + \mathbf{LU}\phi^{(k)} - \mathbf{A}\phi^{(k)} \Leftrightarrow \mathbf{LU}\phi^{(k+1)} - \mathbf{LU}\phi^{(k)} = \mathbf{b} - \mathbf{A}\phi^{(k)},$$

ou seja,

$$\mathbf{L}\mathbf{U}\mathbf{e}^{(k)} = \mathbf{r}^{(k)},$$

sendo $\mathbf{e} = \boldsymbol{\phi}^{(k+1)} - \boldsymbol{\phi}^{(k)}$ a correção da solução na iteração $k + 1$ e $\mathbf{r} = \mathbf{b} - \mathbf{A}\boldsymbol{\phi}^{(k)}$ o resíduo na iteração k . Assim, resolve-se o sistema de equações dado pela equação (1.1) em um processo iterativo composto pelos seguintes sistemas lineares triangulares

1. Resolve-se $L\mathbf{y}^{(k)} = \mathbf{r}^{(k)}$ para obter $\mathbf{y}^{(k)}$;
2. Resolve-se $U\mathbf{e}^{(k)} = \mathbf{y}^{(k)}$ para obter $\mathbf{e}^{(k)}$.

Desta forma, a solução na iteração $k + 1$ é dada por $\boldsymbol{\phi}^{(k+1)} = \boldsymbol{\phi}^{(k)} + \mathbf{e}^{(k)}$.

Na decomposição LU incompleta apresentada, alguns elementos de \mathbf{L} e de \mathbf{U} são descartados durante o processo de decomposição. Esses elementos são colocados na matriz \mathbf{R} , que contém zeros em todas as posições que correspondem à estrutura não nula de \mathbf{L} e \mathbf{U} , inclusive ao longo da diagonal principal. Existem técnicas que tentam reduzir o efeito destes elementos descartados, compensando as entradas descartadas (SAAD, 2003). Uma técnica é usar decomposições que evitam o descarte de todas as diagonais adicionais obtidas no processo de decomposição. O que acontece se, por exemplo, for usado ILU(1). Outra alternativa é modificar a matriz \mathbf{R} , que foi obtida usando ILU(0), de forma a torná-la diagonal dominante, acelerando assim o processo de convergência. Por exemplo, com a introdução de um parâmetro σ pode-se obter $\mathbf{A} = \mathbf{L}^\sigma \mathbf{U}^\sigma - \mathbf{R}^\sigma$, em que a diagonal principal da matriz \mathbf{R} não é mais nula. A diagonal principal de \mathbf{R} e os elementos u_{kk} da matriz \mathbf{U} são modificados de tal forma que

$$r_{kk} \leftarrow \sigma \sum_{j \neq k} |r_{kj}|$$

e

$$u_{kk} \leftarrow u_{kk} + r_{kk},$$

sendo σ um parâmetro usado para reduzir o efeito da matriz \mathbf{R} . Esse parâmetro é um fator de aceleração de convergência (TROTTERBERG; OOSTERLEE; SCHULLER, 2001). Para reduzir o efeito da matriz \mathbf{R} , soma-se na diagonal principal de \mathbf{U} σ vezes os módulos dos elementos que foram descartados (WESSELING, 2004). Essa decomposição é conhecida como decomposição LU incompleta modificada - MILU. Nota-se que $\sigma = 0$ remete ao método ILU sem modificação.

Segundo Oosterlee e Wesseling (1993), o *solver* ILU é robusto e eficiente. Resolveram as equações de Navier-Stokes para escoamentos de fluido incompressível em várias geometrias bidimensionais, dentre elas uma geometria no formato de L, usando o sistema de coordenadas curvilíneas e o MVF. Entretanto, Rutz (2018), estudou o efeito da anisotropia física na equação de difusão bidimensional anisotrópica, empregando os *solvers*

ILU e MILU. Usando o *solver* MILU, com $\sigma = 0,2$, obteve um melhor fator de convergência assintótico do método *multigrid*. Ou seja, o *solver* MILU foi mais robusto e eficiente para resolução do problema de difusão bidimensional anisotrópica, em comparação com o *solver* ILU.

2.3.3 Método MSI

Schneider e Zedan (1981) apresentaram o procedimento MSI (do inglês, *Modified Strongly Implicit Procedure*) usado para resolver a equação de condução de calor bidimensional, em regime permanente e usando o sistema de coordenadas cartesianas. O modelo é dado por

$$\frac{\partial}{\partial x} \left(k_x \frac{\partial \phi}{\partial x} \right) + \frac{\partial}{\partial y} \left(k_y \frac{\partial \phi}{\partial y} \right) = q, \quad (2.14)$$

sendo k_x e k_y condutividades térmicas para o fluxo de calor nas direções x e y , respectivamente, e q o termo fonte responsável pela possível geração de calor.

O algoritmo proposto por Schneider e Zedan (1981) foi desenvolvido para um esquema com 9 volumes, mas trata, como um caso especial, um esquema com 5 volumes. A equação (2.15) apresenta a discretização da equação (2.14), usando um esquema com 9 volumes e o MVF

$$A_{i,j}^S \phi_{i,j-1} + A_{i,j}^W \phi_{i-1,j} + A_{i,j}^P \phi_{i,j} + A_{i,j}^N \phi_{i,j+1} + A_{i,j}^E \phi_{i+1,j} + A_{i,j}^{SW} \phi_{i-1,j-1} + A_{i,j}^{NW} \phi_{i-1,j+1} + A_{i,j}^{NE} \phi_{i+1,j+1} + A_{i,j}^{SE} \phi_{i+1,j-1} = q_{i,j}, \quad (2.15)$$

em que o superíndice com notação indicial e o subíndice com notação geográfica indicam a posição na malha. Considerando $A_{i,j}^{SW} = A_{i,j}^{NW} = A_{i,j}^{NE} = A_{i,j}^{SE} = 0$, tem-se a discretização da equação (2.14) para 5 volumes. As equações algébricas da forma (2.15), obtidas para cada volume de controle do domínio, produz um sistema de equações algébricas da forma

$$\mathbf{A}\boldsymbol{\phi} = \mathbf{b},$$

sendo \mathbf{A} a matriz dos coeficientes, $\boldsymbol{\phi}$ o vetor das incógnitas e \mathbf{b} o vetor dos termos independentes. A estrutura da matriz \mathbf{A} foi apresentada na FIGURA 7.

Neste método, a decomposição *LU* da matriz $\mathbf{M} = \mathbf{A} + \mathbf{N}$ é tal que as matrizes \mathbf{L} e \mathbf{U} são triangular inferior e superior, respectivamente, com a diagonal principal de \mathbf{U} sendo unitária. A estrutura das matrizes \mathbf{L} e \mathbf{U} para o esquema com 9 volumes é apresentada na FIGURA 13. Ao se formar o produto *LU*, existem quatro diagonais adicionais não-nulas em \mathbf{M} , denotadas por $\phi_{i,j}^1$, $\phi_{i,j}^2$, $\phi_{i,j}^3$ e $\phi_{i,j}^4$. Desta forma, a matriz $\mathbf{M} = \mathbf{A} + \mathbf{N}$, sendo \mathbf{N} uma matriz que consiste apenas das diagonais $\phi_{i,j}^1$, $\phi_{i,j}^2$, $\phi_{i,j}^3$ e $\phi_{i,j}^4$.

Comparando a matriz \mathbf{A} com a matriz \mathbf{LU} , obtém-se as equações usadas para determinar os coeficientes das matrizes \mathbf{L} e \mathbf{U}

$$a_{i,j} = A_{i,j}^{SW}, \quad (2.16)$$

$$a_{i,j}f_{i-1,j-1} + b_{i,j} = A_{i,j}^S, \quad (2.17)$$

$$b_{i,j}f_{i,j-1} + c_{i,j} = A_{i,j}^{SE}, \quad (2.18)$$

$$a_{i,j}h_{i-1,j-1} + b_{i,j}g_{i,j-1} + d_{i,j} = A_{i,j}^W, \quad (2.19)$$

$$a_{i,j}u_{i-1,j-1} + b_{i,j}h_{i,j-1} + c_{i,j}g_{i+1,j-1} + d_{i,j}f_{i-1,j} + e_{i,j} = A_{i,j}^P, \quad (2.20)$$

$$b_{i,j}u_{i,j-1} + c_{i,j}h_{i+1,j-1} + e_{i,j}f_{i,j} = A_{i,j}^E, \quad (2.21)$$

$$d_{i,j}h_{i-1,j} + e_{i,j}g_{i,j} = A_{i,j}^{NW}, \quad (2.22)$$

$$d_{i,j}u_{i-1,j} + e_{i,j}h_{i,j} = A_{i,j}^N, \quad (2.23)$$

$$e_{i,j}u_{i,j} = A_{i,j}^{NE} \quad (2.24)$$

e as entradas adicionais não nulas da matriz \mathbf{M}

$$\phi_{i,j}^1 = c_{i,j}f_{i+1,j-1}, \quad (2.25)$$

$$\phi_{i,j}^2 = a_{i,j}g_{i-1,j-1}, \quad (2.26)$$

$$\phi_{i,j}^3 = c_{i,j}u_{i+1,j-1} \quad (2.27)$$

e

$$\phi_{i,j}^4 = d_{i,j}g_{i-1,j}. \quad (2.28)$$

As FIGURAS 14 e 15 ilustram a decomposição LU da matriz $\mathbf{M} = \mathbf{A} + \mathbf{N}$, em que $I = N_x$, e a ordenação dos pontos da malha associada a matriz \mathbf{M} , respectivamente.

FIGURA 13 – MATRIZES \mathbf{L} E \mathbf{U} PARA O MÉTODO MSI

(a) Matriz \mathbf{L}

$$\mathbf{L} = \left[\begin{array}{cccc} & & & 0 \\ & & d_{i,j}e_{i,j} & \\ & & 0 & \\ & a_{i,j}b_{i,j}c_{i,j} & & \\ 0 & & & \end{array} \right]$$

(b) Matriz \mathbf{U}

$$\mathbf{U} = \left[\begin{array}{cccc} & & & 0 \\ & & g_{i,j}h_{i,j}u_{i,j} & \\ & & 0 & \\ & 1 & f_{i,j} & \\ 0 & & & \end{array} \right]$$

FIGURA 14 – MATRIZ M PARA O PROCEDIMENTO MSI

$$\mathbf{M} = \begin{pmatrix} & & & & & 0 \\ & I & & & & \phi_{i,j}^4 A_{i,j}^{NW} A_{i,j}^N A_{i,j}^{NE} \\ & & & & & 0 \\ & & & \phi_{i,j}^2 A_{i,j}^W A_{i,j}^P A_{i,j}^E \phi_{i,j}^3 & & \\ & & & 0 & & \\ & & & A_{i,j}^{SW} A_{i,j}^S A_{i,j}^{SE} \phi_{i,j}^1 & & 0 \end{pmatrix}$$

FIGURA 15 – MALHA BIDIMENSIONAL PARA O ESQUEMA COM 9 VOLUMES PARA O PROCEDIMENTO MSI

	$I - 2$	$I - 1$	I	$I + 1$	$I + 2$	
$J + 1$	$\circ \phi_{i,j}^4$	$\bullet A_{i,j}^{NW}$	$\bullet A_{i,j}^N$	$\bullet A_{i,j}^{NE}$		
J	$\circ \phi_{i,j}^2$	$\bullet A_{i,j}^W$	$\bullet A_{i,j}^P$	$\bullet A_{i,j}^E$	$\circ \phi_{i,j}^3$	
$J - 1$		$\bullet A_{i,j}^{SW}$	$\bullet A_{i,j}^S$	$\bullet A_{i,j}^{SE}$	$\circ \phi_{i,j}^1$	

Para cancelar parcialmente a influência dos termos adicionais na matriz \mathbf{M} , Schneider e Zedan (1981) usaram a expansão em séries de Taylor para obter os valores da propriedade nos locais $(i - 2, j)$, $(i + 2, j)$, $(i + 2, j - 1)$ e $(i - 2, j + 1)$, em função dos valores das propriedades nos volumes da malha original. As aproximações usadas por eles foram

$$\phi_{i-2,j} = -\phi_{i,j} + 2\phi_{i-1,j},$$

$$\phi_{i+2,j} = -\phi_{i,j} + 2\phi_{i+1,j},$$

$$\phi_{i+2,j-1} = -2\phi_{i,j} + 2\phi_{i+1,j} + \phi_{i,j-1}$$

e

$$\phi_{i-2,j+1} = -2\phi_{i,j} + 2\phi_{i-1,j} + \phi_{i,j+1}.$$

Além disso, empregaram um parâmetro σ para implementar o cancelamento parcial da influência dos termos adicionais que aparecem na matriz \mathbf{M} . Fizeram isso escrevendo as

equações algébricas de \mathbf{M} , para o esquema de 9 volumes, da seguinte forma

$$\begin{aligned} A_{i,j}^S \phi_{i,j-1} + A_{i,j}^W \phi_{i-1,j} + A_{i,j}^P \phi_{i,j} + A_{i,j}^N \phi_{i,j+1} + A_{i,j}^E \phi_{i+1,j} + A_{i,j}^{SW} \phi_{i-1,j-1} + A_{i,j}^{NW} u_{i-1,j+1} + \\ A_{i,j}^{NE} \phi_{i+1,j+1} + A_{i,j}^{SE} \phi_{i+1,j-1} + \phi_{i,j}^1 [\phi_{i+2,j-1} - \sigma (-2\phi_{i,j} + 2\phi_{i+1,j} + \phi_{i,j-1})] + \\ \phi_{i,j}^2 [\phi_{i-2,j} - \sigma (-\phi_{i,j} + 2\phi_{i-1,j})] + \phi_{i,j}^3 [\phi_{i+2,j} - \sigma (-\phi_{i,j} + 2\phi_{i+1,j})] + \\ \phi_{i,j}^4 [\phi_{i-2,j+1} - \sigma (-2\phi_{i,j} + 2\phi_{i-1,j} + \phi_{i,j+1})] = q_{i,j}. \end{aligned} \quad (2.29)$$

As equações (2.16) a (2.28) foram modificadas, antes de calcular a solução, para incluir a influência do cancelamento parcial indicado na equação (2.29). Rearranjando os termos, tem-se os elementos de \mathbf{L} e \mathbf{U}

$$a_{i,j} = A_{i,j}^{SW}, \quad (2.30)$$

$$b_{i,j} = \frac{A_{i,j}^S - a_{i,j} f_{i-1,j-1} - \sigma A_{i,j}^{SE} f_{i+1,j-1}}{1 - \sigma f_{i,j-1} f_{i+1,j-1}}, \quad (2.31)$$

$$c_{i,j} = A_{i,j}^{SE} - b_{i,j} f_{i,j-1}, \quad (2.32)$$

$$d_{i,j} = \frac{A_{i,j}^W - a_{i,j} h_{i-1,j-1} - b_{i,j} g_{i,j-1} - 2\sigma a_{i,j} g_{i-1,j-1}}{1 + 2\sigma g_{i-1,j}}, \quad (2.33)$$

$$e_{i,j} = A_{i,j}^P - a_{i,j} u_{i-1,j-1} - b_{i,j} h_{i,j-1} - c_{i,j} g_{i+1,j-1} - d_{i,j} f_{i-1,j} + \quad (2.34)$$

$$\sigma (2\phi_{i,j}^1 + \phi_{i,j}^2 + \phi_{i,j}^3 + 2\phi_{i,j}^4), \quad (2.35)$$

$$f_{i,j} = \frac{A_{i,j}^E - b_{i,j} u_{i,j-1} - c_{i,j} h_{i+1,j-1} - 2\sigma (\phi_{i,j}^1 + \phi_{i,j}^3)}{e_{i,j}}, \quad (2.36)$$

$$g_{i,j} = \frac{A_{i,j}^{NW} - d_{i,j} h_{i-1,j}}{e_{i,j}}, \quad (2.37)$$

$$h_{i,j} = \frac{A_{i,j}^N - d_{i,j} u_{i-1,j} - \sigma \phi_{i,j}^4}{e_{i,j}}, \quad (2.38)$$

e

$$u_{i,j} = \frac{A_{i,j}^{NE}}{e_{i,j}}, \quad (2.39)$$

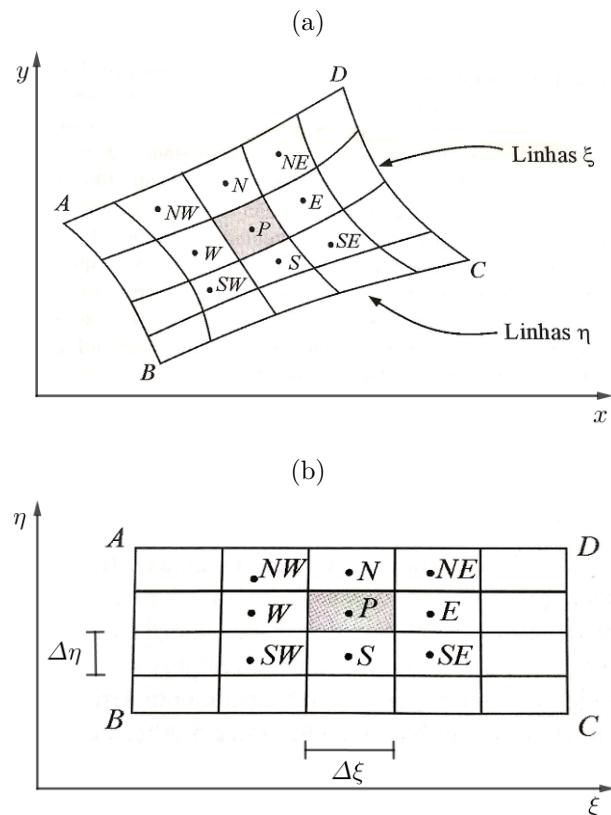
sendo $\phi_{i,j}^1$, $\phi_{i,j}^2$, $\phi_{i,j}^3$ e $\phi_{i,j}^4$ calculadas usando as equações (2.25) a (2.28), em que os valores de $a_{i,j}$, $b_{i,j}$, $c_{i,j}$, $d_{i,j}$, $e_{i,j}$, $f_{i,j}$, $g_{i,j}$, $h_{i,j}$ e $u_{i,j}$ são obtidos das equações (2.30) a (2.39). Assim, feita a decomposição LU da matriz \mathbf{A} , resolve-se o sistema de equações usando o processo iterativo apresentado na subseção 2.3.2.

Schneider e Zedan (1981) também resolveram a equação (2.14) usando o sistema de coordenadas cilíndricas e os solvers MSI, SIP, ADI e SOR. Verificaram que o custo computacional do solver MSI foi consideravelmente menor, sendo 2,426 vezes menor do que o solver SIP, que foi o concorrente mais próximo para este problema, na malha 24×24 . Segundo Anderson, Tannehill e Pletcher (2013), este solver pode ser muito eficiente e empregado para resolver problemas mais complexos.

2.4 TRANSFORMAÇÃO DE COORDENADAS

A transformação (mesmo que numérica) do sistema de coordenadas original (normalmente o cartesiano) para o sistema de coordenadas curvilíneas, permite mapear uma geometria irregular, escrita no sistema de coordenadas cartesianas (x,y), em uma geometria regular no sistema de coordenadas curvilíneas (ξ,η) (THOMPSON; THAMES; MASTIN, 1977; FERZIGER; PERIC, 2002; MALISKA, 2010). A FIGURA 16 mostra um domínio físico e seu respectivo domínio transformado.

FIGURA 16 – DOMÍNIO (a) FÍSICO E (b) TRANSFORMADO



FONTE: Maliska (2010).

Em casos simples, podem-se determinar relações analíticas da transformação de coordenadas do sistema cartesiano para o sistema de coordenadas curvilíneas. Porém, na maior parte dos casos, existirá apenas uma tabela associando os valores discretos do domínio físico (x,y) a valores discretos do plano transformado (ξ,η) (THOMPSON; THAMES; MASTIN, 1977; MALISKA, 2010). As coordenadas curvilíneas de um ponto (x,y) relacionam-se ao sistema cartesiano pelas seguintes equações de transformação

$$\begin{cases} \xi = \xi(x,y) \\ \eta = \eta(x,y) \end{cases} . \quad (2.40)$$

Na transformação de coordenadas é necessário conhecer os parâmetros nos quais estão embutidas as informações da forma e do tamanho real do domínio de cálculo

(MALISKA, 2010). Essas informações sobre a geometria física são fornecidas ao código computacional pelas métricas da transformação de coordenadas, que serão apresentadas a seguir.

As derivadas parciais são transformadas usando a regra da cadeia. Desta forma, considerando $f = f(\xi, \eta)$, tem-se

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial \xi} \frac{\partial \xi}{\partial x} + \frac{\partial f}{\partial \eta} \frac{\partial \eta}{\partial x} \quad (2.41)$$

e

$$\frac{\partial f}{\partial y} = \frac{\partial f}{\partial \xi} \frac{\partial \xi}{\partial y} + \frac{\partial f}{\partial \eta} \frac{\partial \eta}{\partial y}. \quad (2.42)$$

Substituindo f por x e f por y na equação (2.41), tem-se

$$\begin{cases} 1 = x_\xi \xi_x + x_\eta \eta_x \\ 0 = y_\xi \xi_x + y_\eta \eta_x \end{cases}. \quad (2.43)$$

Resolvendo o sistema dado pela equação (2.43), obtém-se

$$\xi_x = y_\eta J \quad (2.44)$$

e

$$\eta_x = -y_\xi J, \quad (2.45)$$

em que os subscritos indicam as derivadas parciais e $J = (x_\xi y_\eta - x_\eta y_\xi)^{-1}$ é o jacobiano da transformação. Ainda, substituindo f por x e f por y na equação (2.42), tem-se

$$\begin{cases} 0 = x_\xi \xi_y + x_\eta \eta_y \\ 1 = y_\xi \xi_y + y_\eta \eta_y \end{cases}. \quad (2.46)$$

Agora, resolvendo o sistema dado pela equação (2.46), obtém-se

$$\xi_y = -x_\eta J \quad (2.47)$$

e

$$\eta_y = x_\xi J. \quad (2.48)$$

A matriz jacobiana da transformação de coordenadas é dada por

$$J_1 = \begin{bmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{bmatrix}, \quad (2.49)$$

cujos elementos são as métricas da transformação do sistema (x, y) para o sistema (ξ, η) , obtidas pelo teorema da função inversa (KREYSZIG, 2009). Ainda, pelo teorema da função inversa, o sistema que fornece a transformação de coordenadas inversa é dado por

$$\begin{cases} x = x(\xi, \eta) \\ y = y(\xi, \eta) \end{cases}.$$

Analogamente ao procedimento algébrico feito anteriormente, tem-se as métricas da transformação inversa dadas por

$$y_\eta = \frac{\xi_x}{J}, \quad (2.50)$$

$$y_\xi = -\frac{\eta_x}{J}, \quad (2.51)$$

$$x_\eta = -\frac{\xi_y}{J} \quad (2.52)$$

e

$$x_\xi = \frac{\eta_y}{J}. \quad (2.53)$$

A matriz jacobiana da inversa da transformação é dada por

$$J_2 = \begin{bmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{bmatrix}. \quad (2.54)$$

As matrizes jacobiana, dadas pelas equações (2.49) e (2.54), se relacionam da seguinte forma

$$J_1 = J_2^{-1}.$$

Assim,

$$J = \det(J_1) = \frac{1}{\det(J_2)} = (x_\xi y_\eta - y_\xi x_\eta)^{-1}. \quad (2.55)$$

Para calcular os comprimentos dL_ξ e dL_η , ilustrados na FIGURA 17, usa-se o teorema de Pitágoras. Dessa forma, obtém-se

$$dL_\xi = \sqrt{(x_\xi \Delta \xi)^2 + (y_\xi \Delta \xi)^2} = \Delta \xi \sqrt{x_\xi^2 + y_\xi^2} = \Delta \xi \sqrt{\gamma}$$

e

$$dL_\eta = \sqrt{(x_\eta \Delta \eta)^2 + (y_\eta \Delta \eta)^2} = \Delta \eta \sqrt{x_\eta^2 + y_\eta^2} = \Delta \eta \sqrt{\alpha},$$

sendo

$$\gamma = x_\xi^2 + y_\xi^2 \quad (2.56)$$

e

$$\alpha = x_\eta^2 + y_\eta^2, \quad (2.57)$$

componentes do tensor métrico. Esses comprimentos podem ser representados pelos seguintes vetores

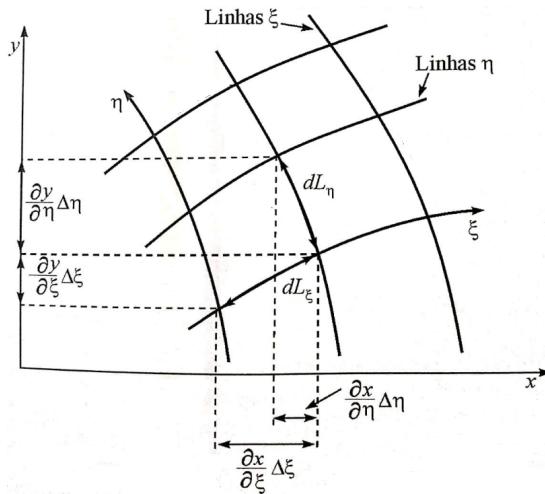
$$\mathbf{dL}_\xi = x_\xi \Delta \xi \mathbf{i} + y_\xi \Delta \xi \mathbf{j} \quad (2.58)$$

e

$$\mathbf{dL}_\eta = x_\eta \Delta \eta \mathbf{i} + y_\eta \Delta \eta \mathbf{j}, \quad (2.59)$$

sendo $\mathbf{i} = (1,0,0)$ e $\mathbf{j} = (0,1,0)$ vetores unitários.

FIGURA 17 – COMPRIMENTOS NO DOMÍNIO FÍSICO



FONTE: Adaptado de Maliska (2010).

Para calcular a área do paralelogramo formado pelos vetores dados pelas equações (2.58) e (2.59), calcula-se a norma do vetor resultante do produto vetorial entre esses vetores (STEINBRUCH; WINTERLE, 1987). Assim, como o produto vetorial desses vetores resulta no seguinte vetor

$$\mathbf{dS} = \mathbf{dL}_\xi \times \mathbf{dL}_\eta = \begin{bmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ x_\xi \Delta\xi & y_\xi \Delta\xi & 0 \\ x_\eta \Delta\eta & y_\eta \Delta\eta & 0 \end{bmatrix} = \mathbf{k}(x_\xi y_\eta - x_\eta y_\xi) \Delta\xi \Delta\eta,$$

sendo $\mathbf{k} = (1,0,0)$, a área do paralelogramo formado por esses vetores é dada por

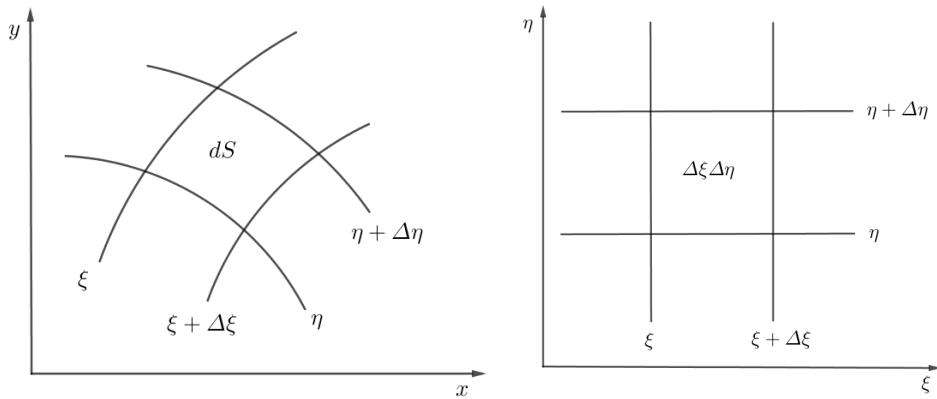
$$dS = \|\mathbf{dS}\| = (x_\xi y_\eta - x_\eta y_\xi) \Delta\xi \Delta\eta. \quad (2.60)$$

Comparando a equação (2.60) com a equação (2.55), constata-se que a expressão entre parênteses na equação (2.60) é igual a $\frac{1}{J}$. Com isso, tem-se

$$\frac{dS}{\Delta\xi \Delta\eta} = \frac{1}{J}, \quad (2.61)$$

ou seja, a relação entre as áreas do domínio físico e do plano transformado é igual a $\frac{1}{J}$. Na equação (2.61), podem-se considerar $\Delta\xi$ e $\Delta\eta$ valores arbitrários. Neste trabalho, considerou-se $\Delta\xi = \Delta\eta = 1$, pois é comum usar esse valor para $\Delta\xi$ e $\Delta\eta$ (FERZIGER; PERIC, 2002). Desta forma, o inverso do jacobiano é exatamente a área do elemento no domínio físico. A FIGURA 18 mostra a área do domínio físico e seu mapeamento no plano transformado.

FIGURA 18 – ÁREAS DO DOMÍNIO FÍSICO (a) E PLANO TRANSFORMADO (b)



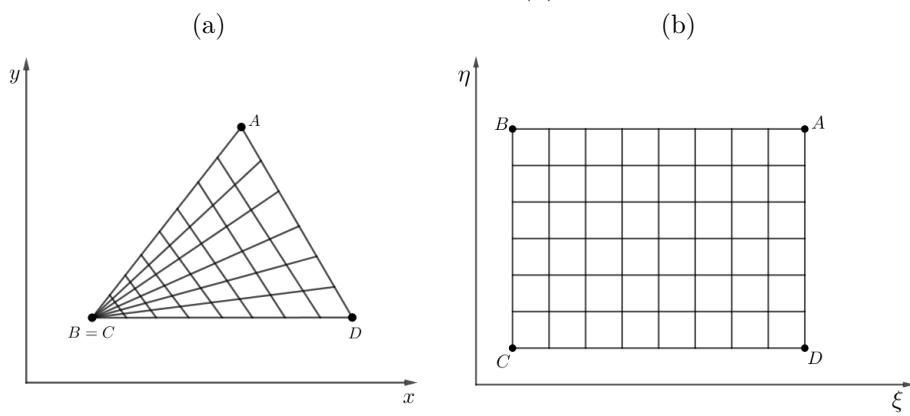
FONTE: Maliska (2010).

2.5 TIPOS DE MAPEAMENTO

O mapeamento de geometrias permite obter diferentes sistemas de coordenadas curvilíneas para a mesma geometria física. A escolha de qual sistema de coordenadas curvilíneas é o mais adequado depende do problema físico (THOMPSON; WARSI; MASTIN, 1985; MALISKA, 2010).

No caso de geometria simplesmente conexa (conforme FIGURA 19), para a geração da malha com mapeamento em bloco único, as fronteiras do domínio transformado são exatamente as fronteiras do domínio físico (conforme FIGURA 19b), ou seja, as condições de contorno no plano transformado são aquelas do plano físico (MALISKA, 2010).

FIGURA 19 – MALHA PARA O TRIÂNGULO (a) E PLANO TRANSFORMADO (b)

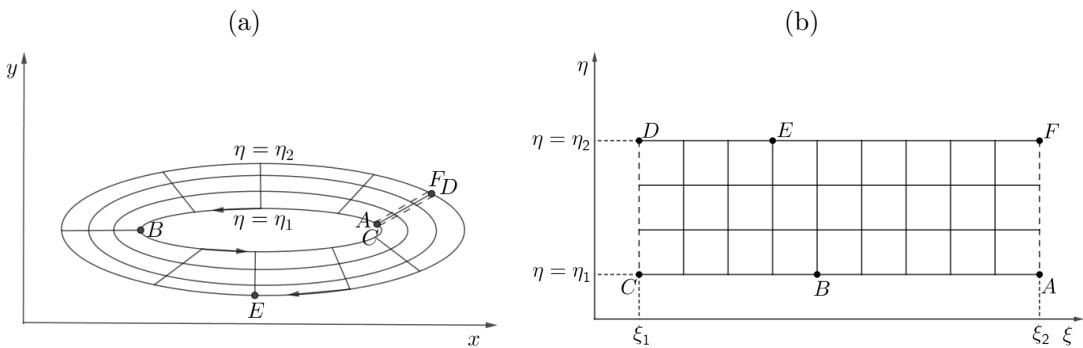


FONTE: Maliska (2010).

As geometrias multiplamente conexas podem ser mapeadas em um bloco único ou usando multiblocos, dependendo de sua complexidade (THOMPSON; WARSI; MASTIN, 1985; MALISKA, 2010). A FIGURA 20a mostra um exemplo de uma geometria

multiplamente conexa que foi mapeada em um único bloco. O mapeamento foi possível fazendo-se um corte na linha ξ , que liga a fronteira interna com a externa, abrindo-se a geometria neste local e representando-a no plano transformado. A FIGURA 20b mostra o plano transformado associado a FIGURA 20a.

FIGURA 20 – MALHA PARA UMA GEOMETRIA MULTIPLAMENTE CONEXA (a) E PLANO TRANSFORMADO (b)



FONTE: Maliska (2010).

Para o mapeamento de geometrias multiplamente conexas em um bloco único, é necessário percorrer todas as fronteiras do domínio físico sem levantar o lápis do papel. Esse processo determina os cortes exigidos. Por exemplo, na FIGURA 20a, partindo com o lápis do ponto A , percorre-se toda a fronteira interna, passando pelo ponto B e chegando em C , que é o próprio ponto A . Para percorrer a fronteira externa sem levantar o lápis do papel, corta-se a geometria ao longo de \overline{CD} , percorre-se toda a superfície externa, passando por E , chegando em F e voltando pelo corte, até chegar em A , concluindo assim o processo. Passando duas vezes pelo mesmo corte, geram-se as duas fronteiras, \overline{CD} e \overline{AF} , no plano transformado. Como no plano físico os segmentos \overline{CD} e \overline{AF} são linhas coincidentes e internas ao domínio, essas fronteiras, no plano transformado, devem ser tratadas como fronteiras internas.

Para exemplificar como tratar as fronteiras internas, considera-se a FIGURA 21, que é o plano transformado da FIGURA 20a. Ao integrar a equação diferencial transformada sobre P , é necessário especificar os valores da função incógnita e de sua derivada em e . Como no domínio físico os segmentos \overline{CD} e \overline{AF} são coincidentes, tem-se $E = E'$. Com isso, os valores da função incógnita e de suas derivadas em e podem ser obtidos através de funções de interpolação que envolvem P e E' .

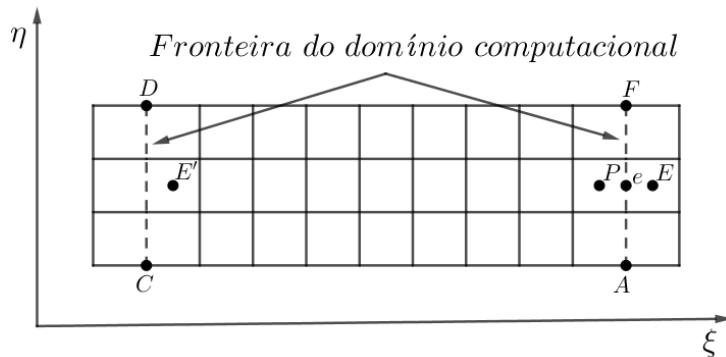
2.6 GERAÇÃO DO SISTEMA DE COORDENADAS CURVILÍNEAS

A geração da malha, ou seja, a discretização do domínio computacional é uma tarefa complexa do processo de simulação numérica. Apesar dos algoritmos de geração de malha serem conceitualmente simples, são de difícil implementação computacional em

função das características das espaciais das geometrias. Uma malha bem gerada respeita a concentração de linhas onde o problema físico requer, não contém elementos excessivamente distorcidos e possui variação suave de espaçamentos entre elementos (MALISKA, 2010). Atender a todos esses requisitos não é uma tarefa fácil de realizar.

Na literatura existem vários métodos para a geração de malhas. Fundamentalmente, esses métodos podem ser classificados em algébricos e diferenciais (THOMPSON; WARSI; MASTIN, 1985; HOFFMAN; CHIANG, 2000). Os métodos algébricos utilizam-se de diferentes tipos de interpolação, são versáteis e rápidos computacionalmente. Os métodos diferenciais, por sua vez, empregam sistemas de equações diferenciais, são mais gerais, porém, apresentam tempo de computação e elaboração matemática maiores (HOFFMANN; CHIANG, 2000). Nesta seção, serão apresentadas as duas categorias de métodos. Primeiramente, o método algébrico (um método que usa interpolação de Lagrange) e, na sequência, o que emprega equações diferenciais (um método que são usadas equações diferenciais elípticas).

FIGURA 21 – TRATAMENTO DOS CORTES DA FIGURA 20a



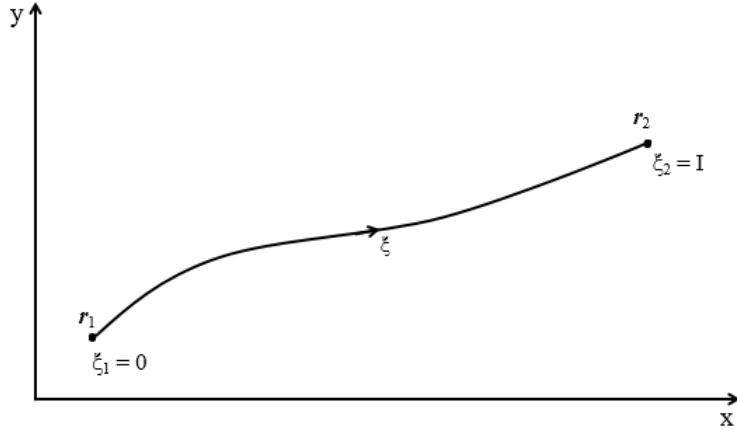
FONTE: Maliska (2010).

2.6.1 Métodos algébricos de geração de coordenadas

Os métodos algébricos de geração de coordenadas são bastante poderosos e largamente empregados. Eles apresentam grandes vantagens, tais como a simplicidade na implementação do código computacional e tempo computacional reduzido (HOFFMAN; CHIANG, 2000).

O método algébrico usado neste trabalho é o que emprega interpolação de Lagrange para gerar as malhas. Neste caso, considera-se $\mathbf{r} = x\mathbf{i} + y\mathbf{j}$, o vetor posição de um ponto genérico, \mathbf{r}_1 e \mathbf{r}_2 os vetores de posição de dois pontos pertencentes a uma mesma linha ξ e I o número de linhas coordenadas que interceptam a linha ξ , conforme mostrado na FIGURA 22.

FIGURA 22 – INTERPOLAÇÃO UNIDIMENSIONAL



FONTE: Maliska (2010).

O polinômio de interpolação genérico é dado por

$$\mathbf{r}(\xi) = \sum_{n=1}^N \phi_n \left(\frac{\xi}{I} \right) \mathbf{r}_n, \quad (2.62)$$

sendo $\mathbf{r}(\xi)$ os vetores posições dos pontos interpolados, \mathbf{r}_n os pontos fornecidos por onde passará o polinômio, $0 \leq \xi \leq I$, N a quantidade de pontos usados para a interpolação e ϕ_n definida por (THOMPSON; WARSI; MASTIN, 1985)

$$\phi_n \left(\frac{\xi}{I} \right) = \prod_{\substack{k=1 \\ k \neq n}}^N \frac{\xi - \xi_k}{\xi_n - \xi_k}. \quad (2.63)$$

A função ϕ possui as seguintes propriedades

1. $\phi_n \left(\frac{\xi_m}{I} \right) = \delta_{mn}$, sendo δ_{mn} denominado delta de Kronecker e definido por

$$\delta_{mn} = \begin{cases} 1 & \text{se } m = n \\ 0 & \text{se } m \neq n \end{cases};$$

2. $\mathbf{r}(\xi_m) = \sum_{n=1}^N \phi_n \left(\frac{\xi_m}{I} \right) \mathbf{r}_n = \sum_{n=1}^N \delta_{mn} \mathbf{r}_n = \mathbf{r}_m$.

Para $N = 2$ (polinômios lineares), caso empregado neste trabalho, tem-se

$$\phi_1 \left(\frac{\xi}{I} \right) = \frac{\xi - \xi_2}{\xi_1 - \xi_2} = \frac{\xi - I}{0 - I} = 1 - \frac{\xi}{I} \quad (2.64)$$

e

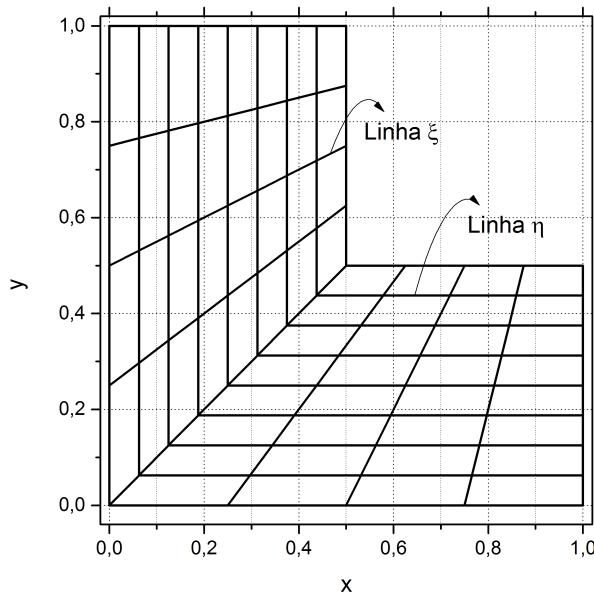
$$\phi_2 \left(\frac{\xi}{I} \right) = \frac{\xi - \xi_1}{\xi_2 - \xi_1} = \frac{\xi - 0}{I - 0} = \frac{\xi}{I}. \quad (2.65)$$

Deste modo, o polinômio de Lagrange é dado por

$$\mathbf{r}(\xi) = \left(1 - \frac{\xi}{I}\right) \mathbf{r}_1 + \frac{\xi}{I} \mathbf{r}_2. \quad (2.66)$$

A vantagem de usar o polinômio de Lagrange na forma dada pela equação (2.66) é que esse polinômio está normalizado em relação à variável ξ . Desta forma, substituindo ξ pelos valores inteiros de 1 a $I - 1$, tem-se os os valores de x e y ao longo da reta que une ξ_1 e ξ_2 . A FIGURA 23 apresenta uma malha gerada empregando-se interpolações de Lagrange para um domínio no formato de L, que é o domínio de interesse.

FIGURA 23 – MALHA GERADA UTILIZANDO INTERPOLAÇÃO DE LAGRANGE



A seguir, apresenta-se o método que usa equações diferenciais elípticas para gerar as malhas. Esse método também é usado neste trabalho, uma vez que é largamente empregado porque suaviza e melhora a não ortogonalidade da malha (HOFFMAN; CHIANG, 2000; ANDERSON; TANNEHILL; PLETCHER, 2013).

2.6.2 Métodos diferenciais de geração de coordenadas

O método diferencial consiste em resolver um sistema de equações diferenciais definido na geometria de interesse. É um método apropriado porque as soluções do sistema diferencial elíptico são isolinhas, que são suaves. Além disso, empregando equações de Laplace para gerar as malhas, o princípio do máximo das funções harmônicas é garantido (THOMPSON; WARSI; MASTIN, 1985; ANDERSON; TANNEHILL; PLETCHER, 2013). Com isso, tem-se a garantia que o jacobiano da transformação não se anula no domínio e da unicidade das funções ξ e η a serem determinadas.

Neste trabalho, o sistema de equações diferenciais elípticas usado para determinar o sistema de coordenadas curvilíneas, é dado por

$$\begin{cases} \frac{\partial^2 \xi}{\partial x^2} + \frac{\partial^2 \xi}{\partial y^2} = 0 \\ \frac{\partial^2 \eta}{\partial x^2} + \frac{\partial^2 \eta}{\partial y^2} = 0 \end{cases}. \quad (2.67)$$

As condições de contorno de Dirichlet para a variável ξ , em todas as fronteiras, são as seguintes

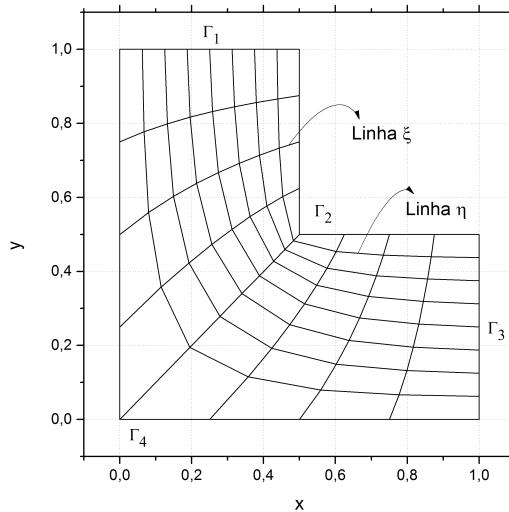
$$\begin{cases} \xi = \xi_1 = \text{constante em } \Gamma_1 \\ \xi = \xi_N = \text{constante em } \Gamma_3 \\ \xi = \text{distribuição especificada } \Gamma_2 \\ \xi = \text{distribuição especificada } \Gamma_4 \end{cases}, \quad (2.68)$$

e para variável η são dadas por

$$\begin{cases} \eta = \eta_1 = \text{constante em } \Gamma_4 \\ \eta = \eta_M = \text{constante em } \Gamma_2 \\ \eta = \text{distribuição especificada } \Gamma_1 \\ \eta = \text{distribuição especificada } \Gamma_3 \end{cases}. \quad (2.69)$$

A FIGURA 24 apresenta a malha obtida através da solução do sistema de equações diferenciais elípticas dado pela equação (2.67), juntamente com as condições de contorno, dadas pelos sistemas de equações (2.68) e (2.69).

FIGURA 24 – MALHA GERADA UTILIZANDO A SOLUÇÃO DO SISTEMA DE EQUAÇÕES ELÍPTICAS



As equações de Laplace, como equações geradoras, dão origem a coordenadas que apresentam maior grau de uniformidade possível. Longe das fronteiras, a tendência é a obtenção de quadriláteros curvilíneos formados pelas linhas ξ e η . Em superfícies

convexas, a tendência é concentrar as linhas coordenadas, ocorrendo o contrário em superfícies côncavas. Portanto, se for necessária a concentração de linhas, por exemplo, na superfície côncava, são introduzidos termos fontes no sistema dado pela equação (2.67). Neste caso, o sistema gerador tem a seguinte forma (THOMPSON; THAMES; MASTIN, 1977; ANDERSON; TANNEHILL; PLETCHER, 2013)

$$\begin{cases} \nabla^2\xi = P(\xi, \eta) \\ \nabla^2\eta = Q(\xi, \eta) \end{cases}, \quad (2.70)$$

sendo

$$P(\xi, \eta) = -\sum_{i=1}^N a_i \text{sign}(\xi - \xi_i) e^{-c_i |\xi - \xi_i|} - \sum_{j=1}^M b_j \text{sign}(\xi - \xi_j) e^{-d_j [(\xi - \xi_j)^2 + (\eta - \eta_j)^2]^{\frac{1}{2}}} \quad (2.71)$$

e

$$Q(\xi, \eta) = -\sum_{i=1}^N a_i \text{sign}(\eta - \eta_i) e^{-c_i |\eta - \eta_i|} - \sum_{j=1}^M b_j \text{sign}(\eta - \eta_j) e^{-d_j [(\xi - \xi_j)^2 + (\eta - \eta_j)^2]^{\frac{1}{2}}}. \quad (2.72)$$

Para a função P , ξ_i e η_i são as linhas para as quais as linhas ξ serão atraídas e (ξ_j, η_j) são os pontos para os quais as linhas ξ serão atraídas. O efeito da amplitude a_i é para atrair as linhas ξ em torno da linha ξ_i e o efeito da amplitude b_j é para atrair as linhas ξ em torno do ponto (ξ_j, η_j) . Os coeficientes c_i e d_j podem ser ajustados para aumentar ou diminuir a atração. A função representada por $\text{sign}(\cdot)$ é a função sinal. O efeito da função Q sobre as linhas η é análogo. O primeiro termo do lado direito das equações (2.71) e (2.72) são responsáveis pela atração entre as linhas coordenadas e o segundo termo, pela atração das linhas aos pontos escolhidos (THOMPSON; THAMES; MASTIN, 1977).

A maneira largamente difundida para a resolução do sistema dado pela equação (2.70) é resolvê-lo no plano transformado (THOMPSON; THAMES; MASTIN, 1974; MALISKA, 2010). Dessa forma, o sistema gerador de malhas e de equações que modela o problema físico são resolvidos no plano transformado.

Para transformar as equações geradoras do plano físico para o plano transformado, consideram-se as equações de transformações dadas pela equação (2.40). É possível obter as expressões para as derivadas de primeira ordem e de segunda ordem da função f através da regra da cadeia

$$\begin{cases} f_x = f_\xi \xi_x + f_\eta \eta_x \\ f_y = f_\xi \xi_y + f_\eta \eta_y \end{cases}.$$

Assim,

$$f_{xx} = f_\xi \xi_{xx} + f_\eta \eta_{xx} + \xi_x^2 f_{\xi\xi} + \eta_x^2 f_{\eta\eta} + 2\xi_x \eta_x f_{\xi\eta} \quad (2.73)$$

e

$$f_{yy} = f_\xi \xi_{yy} + f_\eta \eta_{yy} + \xi_y^2 f_{\xi\xi} + \eta_y^2 f_{\eta\eta} + 2\xi_y \eta_y f_{\xi\eta}. \quad (2.74)$$

Considerando $f = x$ e $f = y$ na equação (2.73), obtém-se

$$\begin{cases} 0 = x_\xi \xi_{xx} + x_\eta \eta_{xx} + E_1 \\ 0 = y_\xi \xi_{xx} + y_\eta \eta_{xx} + F_1 \end{cases}, \quad (2.75)$$

sendo

$$E_1 = \xi_x^2 x_{\xi\xi} + \eta_x^2 x_{\eta\eta} + 2\xi_x \eta_x x_{\xi\eta} \quad (2.76)$$

e

$$F_1 = \xi_x^2 y_{\xi\xi} + \eta_x^2 y_{\eta\eta} + 2\xi_x \eta_x y_{\xi\eta}. \quad (2.77)$$

Analogamente, considerando-se $f = x$ e $f = y$ na equação (2.74), obtém-se

$$\begin{cases} 0 = x_\xi \xi_{yy} + x_\eta \eta_{yy} + E_2 \\ 0 = y_\xi \xi_{yy} + y_\eta \eta_{yy} + F_2 \end{cases}, \quad (2.78)$$

sendo

$$E_2 = \xi_y^2 x_{\xi\xi} + \eta_y^2 x_{\eta\eta} + 2\xi_y \eta_y x_{\xi\eta} \quad (2.79)$$

e

$$F_2 = \xi_y^2 y_{\xi\xi} + \eta_y^2 y_{\eta\eta} + 2\xi_y \eta_y y_{\xi\eta}. \quad (2.80)$$

Resolvendo o sistema dado pela equação (2.75), tem-se

$$\left(x_\xi - \frac{x_\eta}{y_\eta} y_\xi \right) \xi_{xx} = -E_1 + F_1 \frac{x_\eta}{y_\eta}. \quad (2.81)$$

Para simplificar a equação (2.81), são empregadas as equações (2.50), (2.51), (2.52), (2.53) e (2.55). Desse modo, obtém-se

$$\xi_{xx} = -(E_1 \xi_x + F_1 \xi_y). \quad (2.82)$$

Analogamente, obtém-se

$$\eta_{xx} = -(E_1 \eta_x + F_1 \eta_y). \quad (2.83)$$

Do sistema dado pela equação (2.78), obtém-se

$$\xi_{yy} = -(E_2 \xi_x + F_2 \xi_y) \quad (2.84)$$

e

$$\eta_{yy} = -(E_2 \eta_x + F_2 \eta_y). \quad (2.85)$$

Aplicando-se as equações (2.82) a (2.85) no sistema dado pela equação (2.70), tem-se

$$\begin{cases} -(E_1 + E_2) \xi_x - (F_1 + F_2) \xi_y = P(\xi, \eta) \\ -(E_1 + E_2) \eta_x - (F_1 + F_2) \eta_y = Q(\xi, \eta) \end{cases}. \quad (2.86)$$

Considerando

$$E = E_1 + E_2 \quad (2.87)$$

e

$$F = F_1 + F_2, \quad (2.88)$$

pode-se escrever o sistema dado pela equação (2.86) da seguinte forma

$$\begin{cases} E\xi_x + F\xi_y = -P(\xi, \eta) \\ E\eta_x + F\eta_y = -Q(\xi, \eta) \end{cases}.$$

Resolvendo esse sistema, tem-se

$$E = -\frac{P\eta_y - Q\xi_x}{J} \quad (2.89)$$

e

$$F = -\frac{-P\eta_x + Q\xi_x}{J}. \quad (2.90)$$

Substituindo as equações (2.76) e (2.79) na equação (2.87), tem-se

$$(\xi_x^2 + \xi_y^2)x_{\xi\xi} + (\eta_x^2 + \eta_y^2)x_{\eta\eta} + 2(\xi_x\eta_x + \xi_y\eta_y)x_{\xi\eta} = E$$

ou seja,

$$ax_{\xi\xi} + bx_{\eta\eta} + 2dx_{\xi\eta} = E, \quad (2.91)$$

sendo

$$\begin{aligned} a &= \xi_x^2 + \xi_y^2, \\ b &= \eta_x^2 + \eta_y^2 \end{aligned}$$

e

$$d = \xi_x\eta_x + \xi_y\eta_y.$$

De forma análoga, substituindo as equações (2.77) e (2.80) na equação (2.88), tem-se

$$(\xi_x^2 + \xi_y^2)y_{\xi\xi} + (\eta_x^2 + \eta_y^2)y_{\eta\eta} + 2(\xi_x\eta_x + \xi_y\eta_y)y_{\xi\eta} = F$$

ou seja,

$$ay_{\xi\xi} + by_{\eta\eta} + 2dy_{\xi\eta} = F. \quad (2.92)$$

As equações (2.91) e (2.92) formam o seguinte sistema de equações

$$\begin{cases} ax_{\xi\xi} + bx_{\eta\eta} + 2dx_{\xi\eta} = E \\ ay_{\xi\xi} + by_{\eta\eta} + 2dy_{\xi\eta} = F \end{cases},$$

no qual ao serem empregados os valores de E e F , equações (2.89) e (2.90), obtém-se

$$\begin{cases} ax_{\xi\xi} + bx_{\eta\eta} + 2dx_{\xi\eta} = -\frac{P\eta_y - Q\xi_x}{J} \\ ay_{\xi\xi} + by_{\eta\eta} + 2dy_{\xi\eta} = -\frac{-P\eta_x + Q\xi_x}{J} \end{cases}. \quad (2.93)$$

Substituindo as equações (2.44), (2.45), (2.47), e (2.48) no sistema dado pela equação (2.93) tem-se

$$\begin{cases} ax_{\xi\xi} + bx_{\eta\eta} + 2dx_{\xi\eta} + (Px_\xi + Qx_\eta) = 0 \\ ay_{\xi\xi} + by_{\eta\eta} + 2dy_{\xi\eta} + (Py_\xi + Qy_\eta) = 0 \end{cases}. \quad (2.94)$$

E ainda, como

$$a = \alpha J^2, \quad (2.95)$$

$$b = \gamma J^2, \quad (2.96)$$

e

$$d = -\beta J^2, \quad (2.97)$$

sendo α e γ dados pelas equações (2.57) e (2.56), respectivamente, e

$$\beta = x_\xi x_\eta + y_\xi y_\eta. \quad (2.98)$$

Substituindo as equações (2.95), (2.96) e (2.97) no sistema dado pela equação (2.94), tem-se o sistema

$$\begin{cases} \alpha J^2 x_{\xi\xi} + \gamma J^2 x_{\eta\eta} - 2\beta J^2 x_{\xi\eta} + (Px_\xi + Qx_\eta) = 0 \\ \alpha J^2 y_{\xi\xi} + \gamma J^2 y_{\eta\eta} - 2\beta J^2 y_{\xi\eta} + (Py_\xi + Qy_\eta) = 0 \end{cases},$$

ou seja,

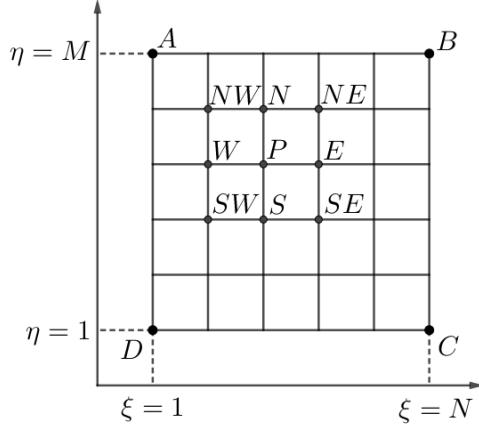
$$\begin{cases} \alpha x_{\xi\xi} + \gamma x_{\eta\eta} - 2\beta x_{\xi\eta} + \frac{1}{J^2}(Px_\xi + Qx_\eta) = 0 \\ \alpha y_{\xi\xi} + \gamma y_{\eta\eta} - 2\beta y_{\xi\eta} + \frac{1}{J^2}(Py_\xi + Qy_\eta) = 0 \end{cases}. \quad (2.99)$$

Este sistema é o sistema gerador de malhas transformado, sendo as variáveis dependentes as coordenadas x e y e independentes ξ e η . Dessa forma, as condições de contorno devem ser especificadas para as variáveis x e y (que aparecem naturalmente na geometria do problema). Como ξ e η são variáveis independentes, pode-se atribuir valores a essas variáveis, como por exemplo na FIGURA 25, atribui-se $\xi = 1$ a N e $\eta = 1$ a M , sendo N e M números inteiros.

Considerando $P = Q = 0$ no sistema dado pela equação (2.99), tem-se o sistema dado pela equação (2.67) transformado. Sendo esse o sistema transformado, usado neste trabalho, para a geração de malhas.

Para resolver numericamente o sistema dado pela equação (2.99), reescrevem-se as equações desse sistema para a variável genérica ϕ , que pode representar x ou y . Assim, tem-se

$$\alpha \phi_{\xi\xi} + \gamma \phi_{\eta\eta} - 2\beta \phi_{\xi\eta} + \frac{1}{J^2}(P\phi_\xi + Q\phi_\eta) = 0. \quad (2.100)$$

FIGURA 25 – PONTOS VIZINHOS DO PONTO P NO PLANO TRANSFORMADO

É conveniente resolver a equação (2.100) numericamente empregando-se, para isso, aproximações de segunda ordem pelo MDF (TANNEHILL; ANDERSON; PLETCHER, 1997; BURDEN; FAIRES, 2016). Assim,

$$\phi_{\xi\xi} = \frac{\phi_E + \phi_W - 2\phi_P}{(\Delta\xi)^2}, \quad (2.101)$$

$$\phi_{\eta\eta} = \frac{\phi_N + \phi_S - 2\phi_P}{(\Delta\eta)^2}, \quad (2.102)$$

$$\phi_{\xi\eta} = \frac{\phi_{NE} + \phi_{SW} - \phi_{SE} - \phi_{NW}}{4\Delta\xi\Delta\eta}, \quad (2.103)$$

$$\phi_\xi = \frac{\phi_E - \phi_W}{2\Delta\xi}, \quad (2.104)$$

e

$$\phi_\eta = \frac{\phi_N - \phi_S}{2\Delta\eta}, \quad (2.105)$$

sendo P o nó obtido pela interseção das linhas de malhas $\xi = 3$ e $\eta = 4$, conforme mostra a FIGURA 25, e W , E , S , N , SW , NW , SE e NE são os nós vizinhos ao nó P , que representam as posições oeste, leste, sul, norte, sudoeste, noroeste, sudeste e nordeste, respectivamente.

Substituindo as equações (2.101) a (2.105) na equação (2.100) e rearranjando os termos, obtém-se uma equação na forma

$$a_p\phi_P = a_e\phi_E + a_w\phi_W + a_n\phi_N + a_s\phi_S + a_{ne}\phi_{NE} + a_{se}\phi_{SE} + a_{nw}\phi_{NW} + a_{sw}\phi_{SW}, \quad (2.106)$$

sendo

$$a_p = 2(\alpha + \gamma),$$

$$a_e = \alpha + \frac{P}{2J^2},$$

$$a_w = \alpha - \frac{P}{2J^2},$$

$$a_n = \gamma + \frac{Q}{2J^2},$$

$$a_s = \gamma - \frac{Q}{2J^2},$$

$$a_{ne} = -\frac{\beta}{2},$$

$$a_{se} = \frac{\beta}{2},$$

$$a_{nw} = \frac{\beta}{2},$$

e

$$a_{sw} = -\frac{\beta}{2},$$

com α , γ e β dados pelas equações (2.57), (2.56) e (2.98), respectivamente.

Com as condições de contorno especificadas no plano transformado, resolve-se o sistema dado pela equação (2.106) empregando o Algoritmo 1 apresentado a seguir.

Algoritmo 1: Geração de malhas empregando equações elípticas

1. Estimar valores para x e y para todos os pontos internos;
 2. Calcular as métricas α , β e γ , dadas pelas equações (2.57), (2.56) e (2.98), respectivamente. Bem como, o jacobiano, equação (2.55);
 3. Resolver o sistema dado pela equação (2.106), obtendo novos valores para x e y ;
 4. Voltar ao item 2, iterando até que uma distribuição de x e y adequada seja encontrada.
-

Com os pontos (x,y) calculados para todas as interseções das linhas ξ e η , tem-se a malha. Consequentemente, pode-se calcular as componentes do tensor métrico, as métricas e o jacobiano da transformação.

3 MÉTODO *MULTIGRID*

Neste capítulo apresentam-se a ideia do método *multigrid*, a definição de equação residual, a análise de convergência e de erros, as principais componentes do *multigrid*, ciclos e algoritmo FAS.

O método *multigrid* é uma técnica muito utilizada para acelerar a convergência de sistemas de equações do tipo $\mathbf{A}\phi = \mathbf{b}$. O princípio de funcionamento desse método está relacionado ao comportamento de convergência dos métodos iterativos básicos. Os métodos iterativos básicos, tais como Gauss-Seidel e Jacobi, são eficientes na suavização dos erros de alta frequência (componentes oscilatórias do erro), porém as baixas frequências são mantidas praticamente inalteradas. Esses métodos, nas primeiras iterações, reduzem rapidamente as componentes oscilatórias do erro. Assim, após algumas iterações, o erro começa a cair mais lentamente, indicando a predominância das componentes suaves do erro (BRANDT, 1977; WESSELING, 2004; BRIGGS; HENSON; McCORMICK, 2000; TROTTERBERG; OOSTERLEE; SCHULLER, 2001). Como o método *multigrid* percorre um conjunto de malhas com diferentes espaçamentos, após a suavização das componentes oscilatórias na malha fina, as componentes dos erros (suaves) são transferidas para a malha imediatamente mais grossa, tornando-se mais oscilatórias. Esse processo pode ser repetido até outras malhas mais grossas, fazendo com que o método iterativo não perca a eficiência, pois agirá nas diversas componentes dos erros.

3.1 CARACTERIZAÇÃO DO MÉTODO *MULTIGRID*

A ideia principal do método *multigrid* consiste em usar uma discretização de malha adequada para que, usando um suavizador (Gauss-Seidel, Jacobi entre outros), as componentes oscilatórias do erro sejam suavizadas nas primeiras iterações. Após a suavização, o método *multigrid* percorre uma sequência de malhas cada vez mais grossas, de tal forma que os comprimentos de ondas do erro, que agora são longos na malha fina, sejam transformados em curtos na malha imediatamente mais grossa, tornando-se oscilatórios, situação na qual as componentes do erro são suavizadas mais rapidamente.

A razão de engrossamento de malhas é dada por

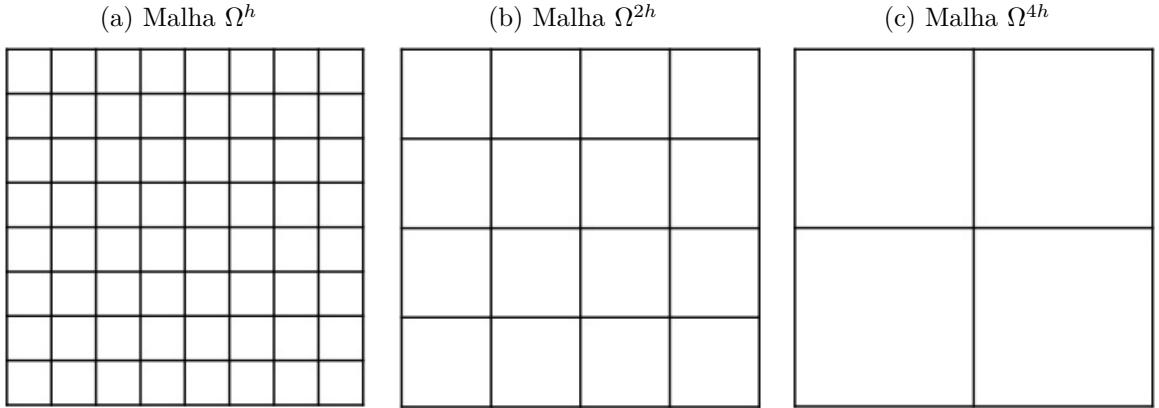
$$r = \frac{H}{h}, \quad (3.1)$$

em que h e H são os tamanhos dos elementos das malhas fina (também chamado de espaçamento da malha fina) e imediatamente mais grossa, respectivamente.

Dada a razão de engrossamento, pode-se determinar o maior número de malhas possível (L_{\max}) para cada tamanho de problema. Por exemplo, considerando-se um problema com $r = 2$, razão de engrossamento utilizada neste trabalho, e $N = 128 \times 128$

incógnitas, o maior número possível de malhas é $L_{\max} = 7$. São elas 128×128 , 64×64 , 32×32 , 16×16 , 8×8 , 4×4 e 2×2 . A FIGURA 26 apresenta uma sequência de três malhas com razão de engrossamento padrão em ambas as direções.

FIGURA 26 – SEQUÊNCIA DE TRÊS MALHAS COM RAZÃO DE ENGROSSAMENTO PADRÃO



3.2 EQUAÇÃO RESIDUAL E ESQUEMAS CS E FAS

Considerando Φ a solução exata do sistema linear dado pela equação (1.1), se $\tilde{\phi}$ é uma aproximação de Φ , o erro algébrico e , ou apenas erro, pode ser calculado por

$$\mathbf{e} = \Phi - \tilde{\phi}. \quad (3.2)$$

A magnitude do erro, dado pela equação (3.2), pode ser medida com alguma norma de vetores. As mais utilizadas são a norma l_1 , a norma euclidiana, l_2 , e a norma infinito, l_∞ , definidas, respectivamente, pelas equações

$$\|\mathbf{e}\|_1 = \sum_{i=1}^n |e_i|,$$

$$\|\mathbf{e}\|_2 = \left(\sum_{i=1}^n e_i^2 \right)^{\frac{1}{2}}$$

e

$$\|\mathbf{e}\|_\infty = \max_{1 \leq i \leq n} |e_i|,$$

sendo e_i a i -ésima componente do vetor erro. Outras normas vetoriais podem ser encontradas em Burden e Faires (2016).

Em geral, a solução exata do sistema linear é desconhecida, por essa razão o erro também o é. Para verificar o quanto a aproximação $\tilde{\phi}$ dista da solução exata do sistema linear, ao longo do processo iterativo, calcula-se o resíduo \mathbf{r} do sistema linear dado pela equação (1.1). O resíduo é definido por

$$\mathbf{r} = \mathbf{b} - \mathbf{A}\tilde{\phi}. \quad (3.3)$$

Com a substituição da equação (1.1) na equação do resíduo, equação (3.3), e utilizando a equação do erro, equação (3.2), pode-se estabelecer uma importante relação entre o erro e o resíduo. Fazendo a substituição, obtém-se

$$\mathbf{r} = \mathbf{A}\Phi - \mathbf{A}\tilde{\phi},$$

que, para o caso linear, pode ser reescrita da seguinte forma

$$\mathbf{r} = \mathbf{A}(\Phi - \tilde{\phi}). \quad (3.4)$$

Com a substituição do erro, equação (3.2), na equação (3.4), tem-se

$$\mathbf{A}\mathbf{e} = \mathbf{r}. \quad (3.5)$$

Essa equação é chamada de equação do resíduo ou equação residual. Ela desempenha um papel importante no âmbito do método *multigrid* e indica que o erro satisfaz o mesmo conjunto de equações que a solução Φ quando \mathbf{b} é substituído pelo resíduo \mathbf{r} (BRIGGS; HENSON; McCORMICK, 2000).

Para o caso não linear, a equação residual é definida por

$$\mathbf{r} = \mathbf{A}\Phi - \mathbf{A}\tilde{\phi}. \quad (3.6)$$

Diferentemente do que acontece no caso linear, no caso não linear $\mathbf{A}\mathbf{e} = \mathbf{A}(\Phi - \tilde{\phi}) \neq \mathbf{A}\Phi - \mathbf{A}\tilde{\phi}$.

Considerando que tenha sido obtida uma aproximação $\tilde{\phi}$ para a solução do sistema linear dado pela equação (1.1), por meio de algum método iterativo, o resíduo pode ser calculado com a equação (3.3). Para melhorar a aproximação $\tilde{\phi}$, calcula-se o erro com a equação (3.5) e, então, corrige-se a solução atual, obtendo uma nova aproximação, através da expressão

$$\phi = \tilde{\phi} + \mathbf{e}. \quad (3.7)$$

Ou seja, a solução numérica é a aproximação atual acrescida do erro. O procedimento descrito acima é um dos princípios fundamentais do *multigrid*.

No capítulo 1 deste texto, foram mencionados dois tipos de esquemas com os quais o *multigrid* pode ser implementado, o CS e o FAS. No CS, o resíduo, equação (3.3), é transferido para a malha grossa subsequente, onde a equação residual, equação (3.5), é suavizada. Na malha mais grossa possível ou desejável, a equação (3.5) é resolvida e o erro é transferido para a malha fina subsequente para corrigir a solução. No FAS, além do resíduo, a aproximação da solução também é transferida para a malha grossa subsequente, na qual a equação residual, equação (3.6), é suavizada. Na malha mais grossa ou desejável percorrida, a equação (3.6) é resolvida e o erro é transferido para a malha fina subsequente para corrigir a solução. De acordo com estas características, o CS e o FAS são indicados,

respectivamente, para problemas lineares e não lineares (BRANDT, 1977). Embora a equação de Laplace bidimensional seja linear, Pinto e Marchi (2006) concluíram que o FAS é mais rápido que o CS para esse problema. Santiago (2010) também concluiu que o esquema FAS é mais rápido do que o CS para a equação de Laplace e para as equações de Navier, para $N = 5 \times 5$ até $N = 2049 \times 2049$. Desta forma, neste trabalho, emprega-se o método *multigrid* com FAS tanto para a equação de Poisson como para as equações de Burgers.

3.3 ANÁLISE DE ERROS

Para analisar o comportamento dos erros é suficiente utilizar o sistema linear homogêneo $\mathbf{A}\phi = 0$ (BRIGGS; HENSON; McCORMICK, 2000). Neste caso, tem-se a vantagem da solução exata ($\phi = \mathbf{0}$) ser conhecida e o erro na aproximação é $\mathbf{e} = -\tilde{\phi}$.

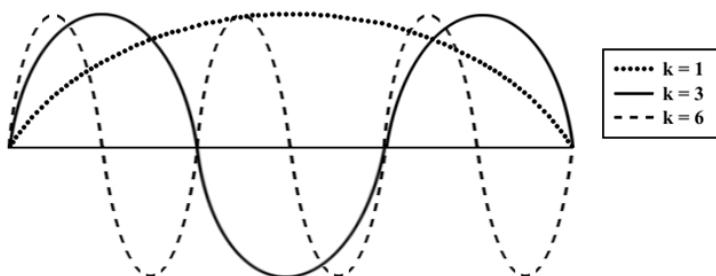
Para ilustrar os efeitos da suavização do erro inicial com os métodos iterativos, considera-se, primeiramente, o caso unidimensional da equação de Laplace com condições de contorno de Dirichlet e estimativa inicial, ou modos de Fourier, \mathbf{v} , cuja a j -ésima componente é dada por

$$v_j = \sin\left(\frac{jk\pi}{N}\right), \quad 0 \leq j \leq N \text{ e } 0 \leq k \leq N - 1,$$

sendo N o número de pontos da malha e k é o número de ondas ou modos de Fourier.

A FIGURA 27 ilustra os modos de Fourier para valores de $k = 1$, $k = 3$, $k = 6$. Os valores de k indicam quantos meios-senos constituem o vetor \mathbf{v} no domínio do problema.

FIGURA 27 – MODOS DE FOURIER

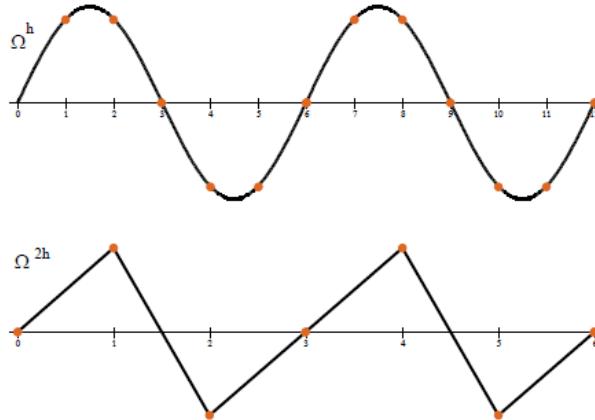


FONTE: Briggs, Henson e McCormick (2000).

Nesta figura, nota-se que pequenos valores de k correspondem a ondas mais longas e suaves, enquanto grandes valores de k correspondem a ondas oscilatórias. Os modos de Fourier da parte inferior do espectro, $1 \leq k < \frac{N}{2}$, são chamados de modos de Fourier de baixa frequência ou modos suaves. Os modos localizados na parte superior do espectro, $\frac{N}{2} \leq k \leq N - 1$, são chamados de modos de Fourier de alta frequência ou modos oscilatórios (BRIGGS; HENSON; McCORMICK, 2000).

A FIGURA 28 ilustra, para o caso unidimensional da equação de Laplace, a representação de um modo do erro na malha Ω^h com $N = 12$ pontos e sua projeção na malha Ω^{2h} com $N = 6$ pontos. O modo de Fourier na malha com $N = 12$ tem $k = 4$. Assim, pela definição, corresponde a um modo suave nesta malha. Na malha com $N = 6$, tem-se também $k = 4$ que, pela definição, corresponde a um modo oscilatório na malha mais grossa. Assim, tem-se que o erro projetado na malha mais grossa é oscilatório.

FIGURA 28 – MODO DO ERRO NA MALHA Ω^h PROJETADO NA MALHA GROSSA Ω^{2h}



FONTE: Briggs, Henson e McCormick (2000).

Para ilustrar os modos de Fourier no caso bidimensional, Santiago (2010) resolveu a equação de Laplace bidimensional, que tem solução exata conhecida. Realizou algumas iterações com o método Gauss-Seidel e estimativa inicial \mathbf{v} dada por uma combinação de modos de Fourier, com altas e baixas frequências na direção y , cujas componentes são dadas por

$$v_{i,j} = \frac{1}{2} \left[\operatorname{sen} \left(\frac{k_1 i \pi}{N_y} \right) + \operatorname{sen} \left(\frac{k_2 j \pi}{N_y} \right) \right], \quad 0 \leq i, j \leq N_y \text{ e } 0 \leq k_1, k_2 \leq N_y - 1,$$

sendo N_y o número de pontos na direção y . As iterações foram realizadas para dois tamanhos de malha, $N = 33 \times 33$ com $k_1 = 2$ e $k_2 = 16$ e $N = 17 \times 17$ com $k_1 = 2$ e $k_2 = 8$, sendo k_1 e k_2 os modos de baixa e alta frequência, respectivamente. Os resultados são mostrados nas FIGURAS 29 e 30. Essas figuras, ilustram as superfícies que apresentam o erro na estimativa inicial e após 2, 4 e 10 iterações nas malhas $N = 33 \times 33$ e $N = 17 \times 17$, respectivamente.

Na FIGURA 29, observa-se que as componentes de alta frequência são suavizadas ao longo do processo iterativo, enquanto as componentes de baixa frequência são pouco influenciadas ao longo das iterações. Quando o mesmo problema é iterado na malha mais grossa ($N = 17 \times 17$), observa-se, na FIGURA 30, que tanto as componentes de alta frequência quanto as de baixa frequência são suavizadas mais rapidamente.

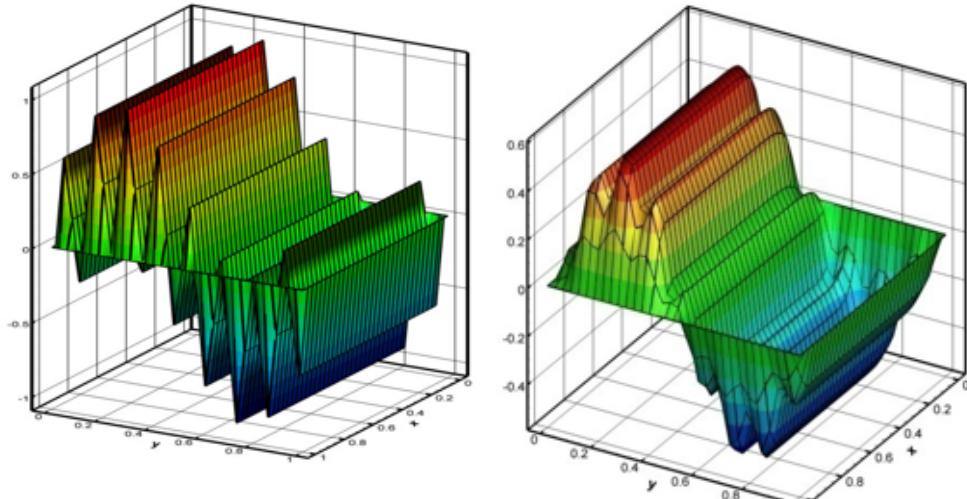
Em geral, as componentes suaves do erro aparecem mais oscilatórias em malhas mais grossas. Isso sugere que quando um processo iterativo começa a tornar-se lento, o

problema seja transferido para a malha imediatamente mais grossa (WESSELING, 2004; TROTTERBERG; OOSTERLEE; SCHULLER, 2001).

FIGURA 29 – SUAVIZAÇÃO DO ERRO COM O MÉTODO GAUSS-SEIDEL NA MALHA 33×33

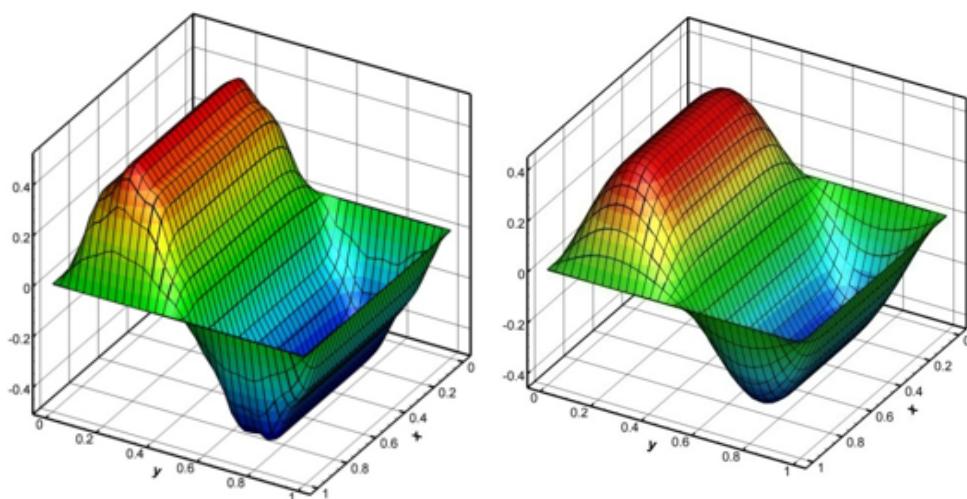
(a) Erro na estimativa inicial

(b) Erro após 2 iterações



(c) Erro após 4 iterações

(d) Erro após 10 iterações



FONTE: Santiago (2010).

3.4 OPERADORES DE TRANSFERÊNCIAS ENTRE MALHAS

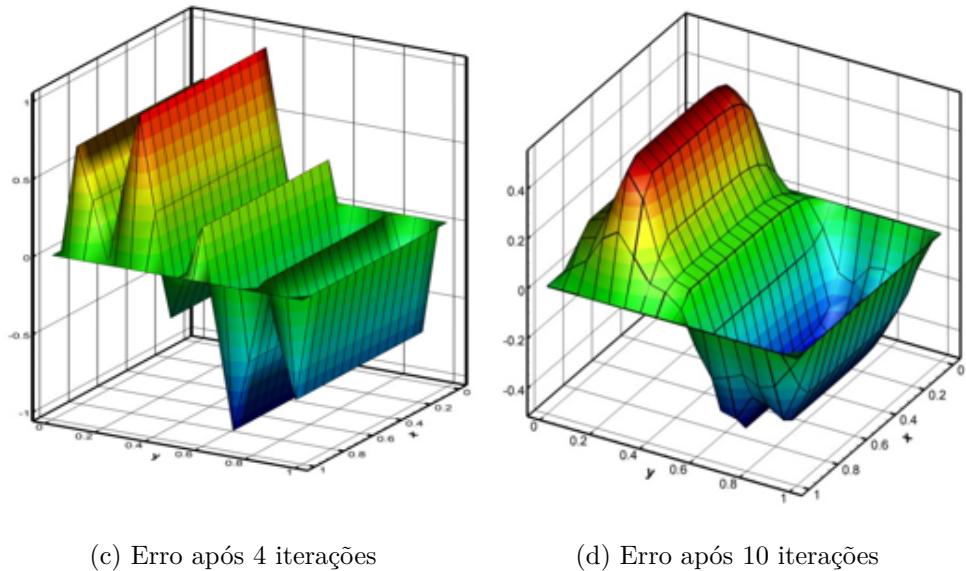
Entre as componentes essenciais do método *multigrid* estão os operadores de restrição e de prolongação, denotados genericamente por I_h^{2h} e I_{2h}^h , respectivamente, com h denotando a malha fina e $2h$ a malha grossa. Com o problema representado na malha fina Ω^h é necessário transferir as informações para às malhas mais grossas. Isso é feito usando um operador denominado operador de restrição. Para fazer o caminho contrário, ou seja, para transferir as informações da malha grossa para as malhas mais finas, utiliza-se um

operador denominado operador de prolongação.

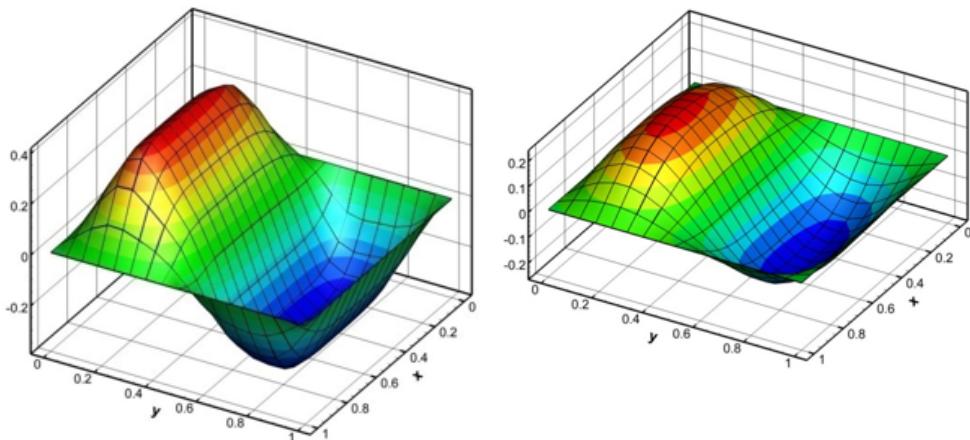
FIGURA 30 – SUAVIZAÇÃO DO ERRO COM O MÉTODO GAUSS-SEIDEL NA MALHA 17×17

(a) Erro na estimativa inicial

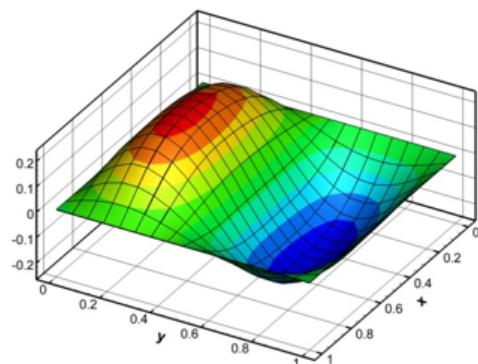
(b) Erro após 2 iterações



(c) Erro após 4 iterações



(d) Erro após 10 iterações



FONTE: Santiago (2010).

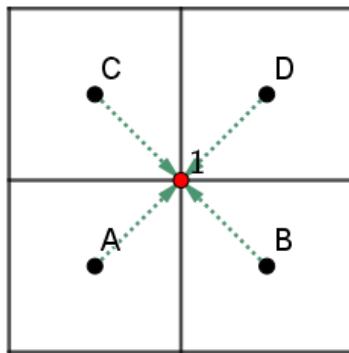
3.4.1 Operadores de restrição

O operador de restrição (I_h^{2h}) é responsável por transferir as informações (resíduo e/ou aproximação da solução) de uma determinada malha Ω^h para malhas mais grossas. A restrição das informações pode ser feita por injeção, meia ponderação ou ponderação completa, entre outras (WESSELING, 2004; BRIGGS; HENSON; McCORMICK, 2000; TROTTERBERG; OOSTERLEE; SCHULLER, 2001).

O operador de restrição por injeção transfere informações dos valores nodais da malha fina Ω^h para os nós coincidentes da malha imediatamente mais grossa Ω^{2h} . Pela sua simplicidade, o operador de restrição por injeção é um dos procedimentos bastante utilizados no MDF. Considerando o MVF, com razão de engrossamento $r = 2$, cada

volume de controle da malha grossa corresponde a quatro volumes de controle na malha fina, conforme mostra a FIGURA 31. Nesta figura, A , B , C e D representam os nós que pertencem à malha mais fina e 1 representa o nó que pertence à malha imediatamente mais grossa. Neste caso, não existem nós coincidentes entre as duas malhas e, portanto, não é possível aplicar o operador por injeção.

FIGURA 31 – DISPOSIÇÃO DO VOLUME DA MALHA FINA E DOS VOLUMES DA MALHA IMEDIATAMENTE MAIS GROSSA



No MVF, método usado neste trabalho, um operador de restrição frequentemente empregado é obtido fazendo-se a média aritmética dos valores da propriedade dos quatro volumes da malha fina que correspondem àquele da malha grossa (JOSHI; VANKA, 1991; TROTTERBERG; OOSTERLEE; SCHULLER, 2001; VERSTEEG; MALALASEKERA, 2007)

$$\mathbf{I}_h^{2h} \phi^h = \frac{1}{4} (\phi_A^h + \phi_B^h + \phi_C^h + \phi_D^h). \quad (3.8)$$

Uma prática bastante comum para restringir o resíduo, considerando-se o MVF, é através da soma dos resíduos relativos aos quatro volumes de controle da malha fina que correspondem àquele da malha grossa (SMITH; COPE; VANKA, 1993; KUMAR; KUMAR; DAS, 2009)

$$\mathbf{I}_h^{2h} \mathbf{r}^h = (r_A^h + r_B^h + r_C^h + r_D^h). \quad (3.9)$$

Como o resíduo pode ser encarado como uma propriedade extensiva, este procedimento proporciona o balanço corretamente. Porém, não é uma prática unânime. Em Versteeg e Malalasekera (2007), encontra-se um exemplo do uso do operador definido como na equação (3.8) para restringir os resíduos.

3.4.2 Operadores de prolongação

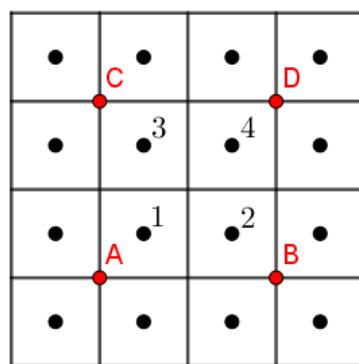
O operador de prolongação ou interpolação (I_{2h}^h) é responsável por transferir informações (correção) da malha grossa para a malha imediatamente mais fina. A interpolação pode ser interpolação linear (para o caso 1D) ou bilinear (para o caso 2D), entre outras. A interpolação bilinear é um dos operadores de prolongação mais utilizados na

literatura (BRIGGS; HENSON; McCORMICK, 2000; TROTTERBERG; OOSTERLEE; SCHULLER, 2001). As expressões para a prolongação bilinear da propriedade ϕ são dadas por

$$I_{2h}^h \phi^{2h} = \begin{cases} \frac{9\phi_A^{2h} + 3\phi_B^{2h} + 3\phi_C^{2h} + \phi_D^{2h}}{16} & \text{para } \phi_1^h \\ \frac{3\phi_A^{2h} + 9\phi_B^{2h} + \phi_C^{2h} + 3\phi_D^{2h}}{16} & \text{para } \phi_2^h \\ \frac{3\phi_A^{2h} + \phi_B^{2h} + 9\phi_C^{2h} + 3\phi_D^{2h}}{16} & \text{para } \phi_3^h \\ \frac{\phi_A^{2h} + 3\phi_B^{2h} + 3\phi_C^{2h} + 9\phi_D^{2h}}{16} & \text{para } \phi_4^h \end{cases}, \quad (3.10)$$

sendo A, B, C e D os nós que pertencem à malha mais grossa e 1, 2, 3 e 4 os nós que pertencem à malha imediatamente mais fina. A FIGURA 32 ilustra o esquema de malhas para o operador de prolongação com a interpolação bilinear.

FIGURA 32 – ESQUEMA DE MALHAS PARA OPERADOR DE PROLONGAÇÃO DA MALHA GROSSA (PONTOS A, B, C E D) PARA A MALHA IMEDIATAMENTE MAIS FINA (PONTOS 1, 2, 3 E 4)



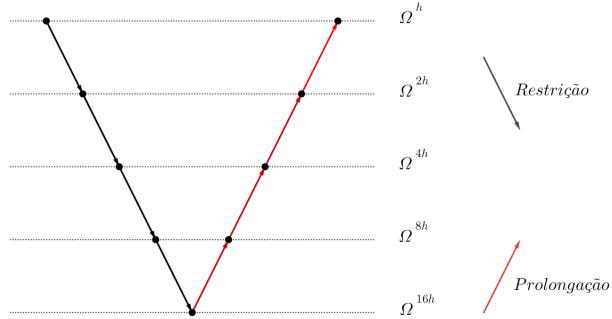
Em Trottenberg, Oosterlee e Schuller (2001), encontram-se outros operadores de prolongação, como o quadrático e o cúbico, para os casos uni e bidimensional.

3.5 CICLOS E ALGORITMO FAS

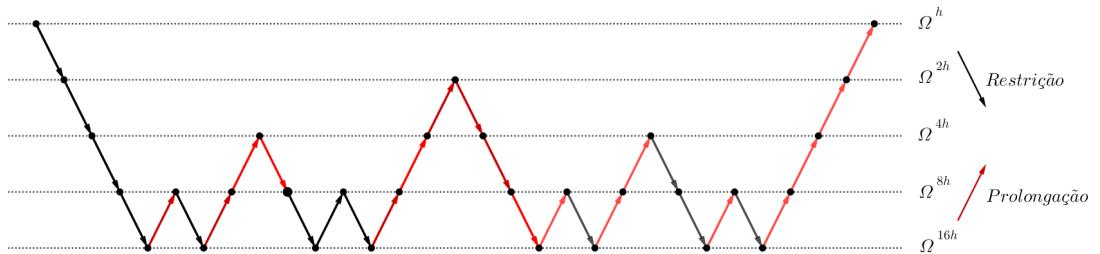
A sequência com que as malhas são percorridas dá origem aos chamados ciclos de suavização. Trottenberg, Oosterlee e Schuller (2001) e Wesseling (2004), apresentam detalhes dos ciclos V , W e F . Os ciclos V e W são os tipos de ciclos padrão, sendo o ciclo W o mais robusto, porém computacionalmente mais caro. Dessa forma, neste trabalho, usa-se o ciclo V , que é um dos mais utilizados (RAYNER, 1991; SMITH; COPE; VANKA, 1993; LI; YU; WANG, 2015). A FIGURA 33 ilustra os ciclos V , W e F . Uma propriedade importante do método *multigrid* é que o número de ciclos necessários para alcançar a convergência na malha mais fina é independente do número de pontos da malha (FERZIGER; PERIC, 2002; BRIGGS; HENSON; McCORMICK, 2000).

FIGURA 33 – CICLOS V, W E F

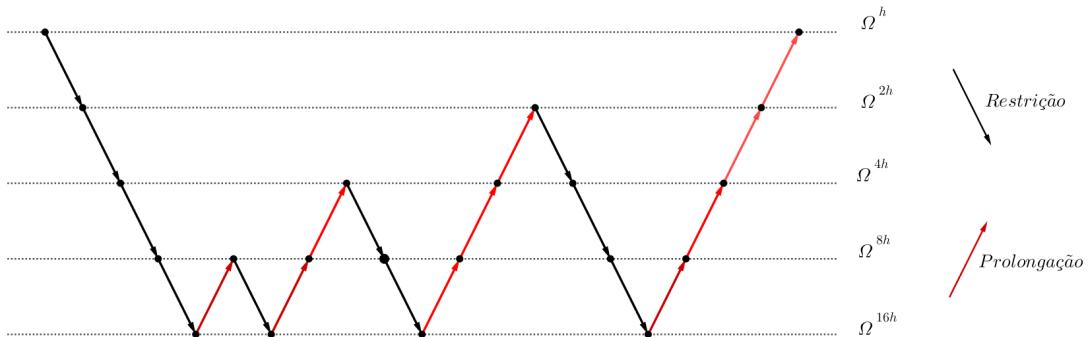
(a) Ciclo V com 5 níveis de malhas



(b) Ciclo W com 5 níveis de malhas



(c) Ciclo F com 5 níveis de malhas



Usa-se $V(\nu_1, \nu_2)$ para indicar ciclo V com ν_1 pré-suavizações (suavização no processo de restrição) e ν_2 pós-suavizações (suavização no processo de prolongação), a mesma notação de Trottenberg, Oosterlee e Schuller (2001). O número de suavizações, ou iterações do suavizador (ν_1, ν_2), geralmente depende do suavizador e não precisa ser necessariamente o mesmo em todos os níveis de malha, tanto no processo de restrição quanto na prolongação. Briggs, Henson e McCormick (2000) afirmam que para problemas que não apresentam dificuldades de convergência, são empregadas 1, 2 ou 3 suavizações.

Os passos básicos do algoritmo FAS, algoritmo utilizado neste trabalho, para um único ciclo V , generalizado para L malhas, $2 \leq L \leq L_{\max}$, são descritos no Algoritmo 2. Na prática, vários ciclos V são realizados, sucessivamente, até que o critério de parada seja alcançado. Para fins de implementação computacional, usa-se uma notação específica. O vetor do lado direito da equação residual é denotado por \mathbf{f} , pois sempre será um vetor do lado direito e o vetor solução dessa equação é denotado por ϕ , pois sempre será um

vetor solução.

Algoritmo 2: Esquema FAS com ciclo V para L malhas

Entrada: FAS($\nu_1, \nu_2, \tilde{\phi}^h, \mathbf{f}^h$)

1. Suavizar $\mathbf{A}^h \phi^h = \mathbf{f}^h$ ν_1 vezes com estimativa inicial $\tilde{\phi}^h$;
2. Calcular $\mathbf{r}^{2h} = I_h^{2h}(\mathbf{f}^h - \mathbf{A}^h \phi^h)$ e $\tilde{\phi}^{2h} = I_h^{2h} \phi^h$;
3. Calcular $\mathbf{f}^{2h} = \mathbf{A}^{2h} \tilde{\phi}^{2h} + \mathbf{r}^{2h}$;
4. Suavizar $\mathbf{A}^{2h} \phi^{2h} = \mathbf{f}^{2h}$ ν_1 vezes com estimativa inicial $\tilde{\phi}^{2h}$;
5. Calcular $\mathbf{r}^{4h} = I_{2h}^{4h}(\mathbf{f}^{2h} - \mathbf{A}^{2h} \phi^{2h})$ e $\tilde{\phi}^{4h} = I_{2h}^{4h} \phi^{2h}$;
6. Calcular $\mathbf{f}^{4h} = \mathbf{A}^{4h} \tilde{\phi}^{4h} + \mathbf{r}^{4h}$;
7. Suavizar $\mathbf{A}^{4h} \phi^{4h} = \mathbf{f}^{4h}$ ν_1 vezes com estimativa inicial $\tilde{\phi}^{4h}$;
8. Calcular $\mathbf{r}^{8h} = I_{4h}^{8h}(\mathbf{f}^{4h} - \mathbf{A}^{4h} \phi^{4h})$ e $\tilde{\phi}^{8h} = I_{4h}^{8h} \phi^{4h}$;
9. Calcular $\mathbf{f}^{8h} = \mathbf{A}^{8h} \tilde{\phi}^{8h} + \mathbf{r}^{8h}$;
- ⋮
10. Resolver $\mathbf{A}^{lh} \phi^{lh} = \mathbf{f}^{lh}$, sendo $l = 2^{L-1}$;
11. Aproximar o erro $\mathbf{e}^{lh} = \phi^{lh} - \tilde{\phi}^{lh}$;
- ⋮
12. Corrigir a solução $\tilde{\phi}^{4h} \leftarrow \phi^{4h} + I_{8h}^{4h} \mathbf{e}^{8h}$;
13. Suavizar $\mathbf{A}^{4h} \phi^{4h} = \mathbf{f}^{4h}$ ν_2 vezes com estimativa inicial $\tilde{\phi}^{4h}$;
14. Corrigir a solução $\tilde{\phi}^{2h} \leftarrow \phi^{2h} + I_{4h}^{2h} \mathbf{e}^{4h}$;
15. Suavizar $\mathbf{A}^{2h} \phi^{2h} = \mathbf{f}^{2h}$ ν_2 vezes com estimativa inicial $\tilde{\phi}^{2h}$;
16. Corrigir a solução $\tilde{\phi}^h \leftarrow \phi^h + I_{2h}^h \mathbf{e}^{2h}$;
17. Suavizar $\mathbf{A}^h \phi^h = \mathbf{f}^h$ ν_2 vezes com estimativa inicial $\tilde{\phi}^h$.

Saída: FAS($\nu_1, \nu_2, \phi^h, \mathbf{f}^h$)

Para o caso de duas equações, como o caso das equações de Burgers, o método *multigrid* pode ser aplicado de formas diferentes. Uma maneira é usar o ciclo V nas equações separadamente, uma depois a outra, ao longo do processo iterativo. Outra maneira, é usar o ciclo V em todas as equações do sistema simultaneamente. Santiago (2010) mostrou que a segunda maneira é a mais eficiente, sendo essa empregada neste trabalho.

4 MODELOS MATEMÁTICOS E NUMÉRICOS

Neste capítulo apresentam-se os modelos matemáticos e numéricos empregados neste trabalho. Foram escolhidos dois modelos matemáticos que fazem parte do escopo da dinâmica de fluidos computacional: as equações de Poisson e de Burgers. Desta forma, têm-se um modelo linear puramente difusivo e um modelo não linear que apresenta advecção e difusão.

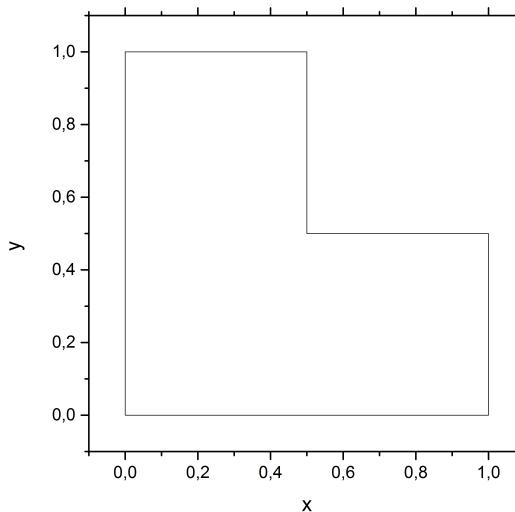
Inicialmente, apresenta-se o domínio de cálculo, ilustrado na FIGURA 34, definido por

$$\begin{cases} 0 \leq x \leq 1, & \text{se } 0 \leq y \leq 0,5; \\ 0 \leq x \leq 0,5, & \text{se } 0,5 < y \leq 1 \end{cases}$$

e

$$\begin{cases} 0 \leq y \leq 1, & \text{se } 0 \leq x \leq 0,5; \\ 0 \leq y \leq 0,5, & \text{se } 0,5 < x \leq 1 \end{cases} .$$

FIGURA 34 – DOMÍNIO DE CÁLCULO



O domínio L é interessante porque pode ser mapeado usando sistemas de coordenadas curvilíneas, sendo o domínio computacional correspondente formado por bloco único, onde a implementação do código computacional é mais simples (FLETCHER, 1991; MALISKA, 2010). Além disso, como ocorre uma mudança de 90° na direção das linhas da malha no domínio físico, pode-se testar diferentes aproximações numéricas para as métricas da transformação de coordenadas, a fim de obter discretizações mais acuradas. Ainda, para mapear esse domínio, é possível usar de multiblocos (OOSTERLEE et al., 1993).

4.1 MODELOS MATEMÁTICOS

4.1.1 Equação de Poisson

A partir da equação da difusão de calor, em coordenadas cartesianas, equação (4.1), considerando-se regime permanente, propriedades físicas constantes e modelo bidimensional, obtém-se uma equação simplificada do tipo Poisson. Na equação tridimensional da difusão de calor

$$\frac{\partial}{\partial x} \left(k_t \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left(k_t \frac{\partial T}{\partial y} \right) + \frac{\partial}{\partial z} \left(k_t \frac{\partial T}{\partial z} \right) + \dot{q} = \rho_d c_p \frac{\partial T}{\partial t}, \quad (4.1)$$

T é variável dependente, que representa a temperatura; x , y e z são variáveis independentes, que representam as direções no sistema de coordenadas cartesianas; k_t é a condutividade térmica do material; \dot{q} é a taxa na qual a energia é gerada, por unidade de volume do meio; e $\rho_d c_p \frac{\partial T}{\partial t}$ é a taxa de variação com o tempo da energia térmica do meio, por unidade de volume, sendo ρ_d a densidade e c_p o calor específico a pressão constante. A dedução dessa equação consta em Incropera et al. (2008).

A equação de Poisson (equação diferencial elíptica) é dada pela seguinte equação

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = S, \quad (4.2)$$

em que S representa o termo fonte, que pode ser uma constante ou uma função de x e y . Se $S > 0$, há geração de calor e se $S < 0$, há absorção de calor.

Neste trabalho, propõe-se a seguinte solução analítica para a temperatura T

$$T(x,y) = \sin\left(\frac{\pi x}{2}\right) \sin\left(\frac{\pi y}{2}\right). \quad (4.3)$$

Essa solução analítica foi obtida com o Método das soluções fabricadas (ROACHE, 2002). Em geral, as soluções fabricadas são utilizadas para verificar a existência de eventuais erros de programação e erros numéricicos.

A substituição das derivadas da função T , equação (4.3), na equação (4.2) resulta na seguinte expressão para o termo fonte

$$S = -\frac{\pi^2}{2} \sin\left(\frac{\pi x}{2}\right) \sin\left(\frac{\pi y}{2}\right).$$

Com a substituição dos valores de x e y sobre o contorno na solução analítica, equação (4.3), as condições de contorno para o domínio, representado pela FIGURA 34, são dadas por

$$\begin{cases} T(x,0) = 0 \\ T(x,1) = \sin\left(\frac{\pi x}{2}\right), & \text{se } 0 \leq x \leq \frac{1}{2} \\ T\left(x,\frac{1}{2}\right) = \frac{\sqrt{2}}{2} \sin\left(\frac{\pi x}{2}\right), & \text{se } \frac{1}{2} \leq x \leq 1 \end{cases}$$

e

$$\begin{cases} T(0,y) = 0 \\ T(1,y) = \operatorname{sen}\left(\frac{\pi y}{2}\right), & \text{se } 0 \leq y \leq \frac{1}{2} \\ T\left(\frac{1}{2},y\right) = \frac{\sqrt{2}}{2} \operatorname{sen}\left(\frac{\pi y}{2}\right), & \text{se } \frac{1}{2} \leq y \leq 1 \end{cases}.$$

Para transformar a equação (4.2), escrita no domínio físico (x,y) para o domínio computacional (ξ,η) , consideram-se as equações de transformação do sistema dado pela equação (2.40).

Como

$$T_x = T_\xi \xi_x + T_\eta \eta_x,$$

então

$$T_{xx} = (T_\xi \xi_x + T_\eta \eta_x)_\xi \xi_x + (T_\xi \xi_x + T_\eta \eta_x)_\eta \eta_x. \quad (4.4)$$

Substituindo as equações (2.44) e (2.45) na equação (4.4), tem-se

$$T_{xx} = [T_\xi y_\eta J + T_\eta (-y_\xi J)]_\xi y_\eta J + [T_\xi y_\eta J + T_\eta (-y_\xi J)]_\eta (-y_\xi J).$$

Considerando

$$A = T_\xi y_\eta J - T_\eta y_\xi J, \quad (4.5)$$

tem-se

$$T_{xx} = A_\xi y_\eta J - A_\eta y_\xi J,$$

que pode ser reescrita como

$$T_{xx} = (Ay_\eta)_\xi J - (Ay_\xi)_\eta J, \quad (4.6)$$

uma vez que

$$(Ay_\eta)_\xi J - (Ay_\xi)_\eta J = J(A_\xi y_\eta + Ay_{\eta\xi} - A_\eta y_\xi - Ay_{\xi\eta}).$$

Substituindo a equação (4.5) na equação (4.6), obtém-se

$$T_{xx} = \left[J \left(T_\xi y_\eta^2 - T_\eta y_\xi y_\eta \right) \right]_\xi J - \left[T_\xi y_\eta y_\xi J - T_\eta y_\xi^2 J \right]_\eta J. \quad (4.7)$$

Como

$$T_y = T_\xi \xi_y + T_\eta \eta_y,$$

então

$$T_{yy} = (T_\xi \xi_y + T_\eta \eta_y)_\xi \xi_y + (T_\xi \xi_y + T_\eta \eta_y)_\eta \eta_y. \quad (4.8)$$

Substituindo as equações (2.47) e (2.48) na equação (4.8), tem-se

$$T_{yy} = [T_\xi (-x_\eta J) + T_\eta (x_\xi J)]_\xi (-x_\eta J) + [T_\xi (-x_\eta J) + T_\eta (x_\xi J)]_\eta (x_\xi J),$$

ou seja,

$$T_{yy} = - \left[J \left(T_\eta x_\xi x_\eta - T_\xi x_\eta^2 \right) \right]_\xi J + \left[J \left(T_\eta x_\xi^2 - T_\xi x_\eta x_\xi \right) \right]_\eta J. \quad (4.9)$$

Ainda, substituindo as equações (4.7) e (4.9) na equação (4.2) e multiplicando o resultado por $\frac{1}{J}$, obtém-se

$$\begin{aligned} & \left[J \left(T_\xi y_\eta^2 - T_\eta y_\xi y_\eta \right) \right]_\xi - \left[J \left(T_\xi y_\eta y_\xi - T_\eta y_\xi^2 \right) \right]_\eta - \left[J \left(T_\eta x_\xi x_\eta - T_\xi x_\eta^2 \right) \right]_\xi + \\ & \left[J \left(T_\eta x_\xi^2 - T_\xi x_\eta x_\xi \right) \right]_\eta = \frac{S}{J}. \end{aligned}$$

Agrupando os termos dessa equação, tem-se

$$\begin{aligned} & \left\{ J \left[T_\xi \left(y_\eta^2 + x_\eta^2 \right) - T_\eta \left(y_\xi y_\eta + x_\xi x_\eta \right) \right] \right\}_\xi + \\ & \left\{ J \left[T_\eta \left(x_\xi^2 + x_\eta^2 \right) - T_\xi \left(x_\xi x_\eta + y_\xi y_\eta \right) \right] \right\}_\eta = \frac{S}{J}. \end{aligned} \quad (4.10)$$

Assim, a equação transformada é dada por

$$\frac{\partial}{\partial \xi} \left[J \left(\alpha \frac{\partial T}{\partial \xi} - \beta \frac{\partial T}{\partial \eta} \right) \right] + \frac{\partial}{\partial \eta} \left[J \left(\gamma \frac{\partial T}{\partial \eta} - \beta \frac{\partial T}{\partial \xi} \right) \right] = \frac{S}{J}, \quad (4.11)$$

sendo $\alpha = x_\eta^2 + y_\eta^2$, $\beta = x_\xi x_\eta + y_\xi y_\eta$ e $\gamma = x_\xi^2 + y_\xi^2$ as componentes do tensor métrico, para o caso bidimensional (FLETCHER, 1991). Essa transformação, como já foi escrito, possui um domínio computacional ou transformado fixo. Os termos do lado esquerdo da equação (4.11) representam a difusão de T através das faces ξ e η , respectivamente, e o termo do lado direito representa o termo fonte.

4.1.2 Equações de Burgers

As equações de Burgers são dadas por um sistema de equações diferenciais parciais não lineares, e representam fenômenos do tipo advecção-difusão. Consistem em uma forma simplificada das equações de Navier-Stokes, reduzidas às equações QML (Quantidade de Movimento Linear), porque o campo de pressões é prescrito e, assim, a equação de conservação da massa não é necessária. Devido às suas não linearidades e pelo fato de serem uma simplificação das equações de Navier-Stokes, constituem um importante problema teste. O modelo considerado neste trabalho tem soluções analíticas obtidas pelo Método das soluções fabricadas e campo de pressões prescrito, os quais serão apresentados na sequência.

Considerando propriedades constantes, regime permanente e coordenadas cartesianas, as equações de Burgers (SHIH; TAN; HWANG, 1989; ZHAO; YU; ZHANG, 2011) são escritas da seguinte forma

$$\frac{\partial u^2}{\partial x} + \frac{\partial uv}{\partial y} = - \frac{\partial p}{\partial x} + \frac{1}{Re} \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \quad (4.12)$$

e

$$\frac{\partial uv}{\partial x} + \frac{\partial v^2}{\partial y} = -\frac{\partial p}{\partial y} + \frac{1}{Re} \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) - S(x, y, Re). \quad (4.13)$$

As variáveis nessas equações são frequentemente usadas na forma adimensional e são relacionadas com as variáveis dimensionais da seguinte forma (FORTUNA, 2000)

$$x = \frac{\hat{x}}{l} \text{ e } y = \frac{\hat{y}}{l},$$

e

$$u = \frac{\hat{u}}{U_\infty}, v = \frac{\hat{v}}{U_\infty} \text{ e } p = \frac{\hat{p}}{U_\infty^2},$$

sendo p a pressão estática, u e v as componentes da velocidade nas direções coordenadas x e y , respectivamente, U_∞ a velocidade de referência, l o comprimento de referência e $Re = \frac{U_\infty l}{\nu_c}$ o número de Reynolds, sendo $\nu_c = \frac{\mu_d}{\rho}$ a viscosidade cinemática, em que μ_d é a viscosidade dinâmica e ρ é a massa específica. O número de Reynolds expressa a razão entre as forças iniciais (devido à velocidade) e as forças viscosas. Basicamente, o número de Reynolds define o comportamento do escoamento de um fluido: laminar ou turbulento. Escoamentos abaixo de um valor Re crítico são laminares; acima desse valor, os escoamentos tornam-se turbulentos.

Neste trabalho, considerou-se $Re = 1$ (SHIH; TAN; HWANG, 1989). Além disso, propõem-se as seguintes soluções analíticas para as componentes do vetor velocidade u e v (SHIH; TAN; HWANG, 1989)

$$u(x, y) = f(x)g'(y) \quad (4.14)$$

e

$$v(x, y) = -f'(x)g(y), \quad (4.15)$$

sendo

$$f(x) = x^5 - \frac{5}{2}x^4 + \frac{35}{16}x^3 - \frac{25}{32}x^2 + \frac{3}{32}x$$

e

$$g(y) = y^5 - \frac{5}{2}y^4 + \frac{35}{16}y^3 - \frac{25}{32}y^2 + \frac{3}{32}y.$$

Desta forma,

$$u(x, y) = \left(x^5 - \frac{5}{2}x^4 + \frac{35}{16}x^3 - \frac{25}{32}x^2 + \frac{3}{32}x \right) \left(5y^4 - 10y^3 + \frac{105}{16}y^2 - \frac{25}{16}y + \frac{3}{32} \right) \quad (4.16)$$

e

$$v(x, y) = \left(-5x^4 + 10x^3 - \frac{105}{16}x^2 + \frac{25}{16}x - \frac{3}{32} \right) \left(y^5 - \frac{5}{2}y^4 + \frac{35}{16}y^3 - \frac{25}{32}y^2 + \frac{3}{32}y \right). \quad (4.17)$$

Substituindo as derivadas das funções u e v , equações (4.14) e (4.15), nas equações (4.12) e (4.13), obtém-se pelo Método das soluções fabricadas o campo de pressões

$$p(x, y, Re = 1) = -\frac{1}{2}f^2(x)[g'(y)]^2 + \frac{1}{2}f^2(x)g(y)g''(y) + f'(x)g'(y) + F(x)g'''(y),$$

e o termo fonte

$$S(x, y, Re = 1) = \frac{1}{2}f^2(x)[g(y)g'''(y) - g'(y)g''(y)] + \{[f'(x)]^2 - f''(x)f(x)\}g(y)g'(y) + 2f'(x)g''(y) + g^{iv}(y)F(x) + f'''(x)g(y),$$

sendo $F(x) = \int f(x)dx$.

Com a substituição dos valores de x e y sobre o contorno nas soluções analíticas, equações (4.16) e (4.17), obtém-se as condições de contorno, do tipo Dirichlet, dadas por

$$v(x, 0) = 0$$

e

$$u(x, 0) = \frac{3}{32}x^5 - \frac{15}{64}x^4 + \frac{105}{512}x^3 - \frac{75}{1024}x^2 + \frac{9}{1024}x,$$

para $0 \leq x \leq 1$,

$$v(0, y) = -\frac{3}{32}y^5 + \frac{15}{64}y^4 - \frac{105}{512}y^3 + \frac{75}{1024}y^2 - \frac{9}{1024}y$$

e

$$u(0, y) = 0,$$

para $0 \leq y \leq 1$,

$$v(x, 1) = 0$$

e

$$u(x, 1) = \frac{3}{32}x^5 - \frac{15}{64}x^4 + \frac{105}{512}x^3 - \frac{75}{1024}x^2 + \frac{9}{1024}x,$$

para $0 \leq x \leq \frac{1}{2}$,

$$v\left(\frac{1}{2}, y\right) = -\frac{1}{64}y^5 + \frac{5}{128}y^4 - \frac{35}{1024}y^3 + \frac{25}{2048}y^2 - \frac{3}{2048}y$$

e

$$u\left(\frac{1}{2}, y\right) = 0,$$

para $\frac{1}{2} \leq y \leq 1$,

$$v\left(x, \frac{1}{2}\right) = 0,$$

e

$$u\left(x, \frac{1}{2}\right) = \frac{1}{64}x^5 - \frac{5}{128}x^4 + \frac{35}{1024}x^3 - \frac{25}{2048}x^2 + \frac{3}{2048}x.$$

para $\frac{1}{2} \leq x \leq 1$,

$$v(1,y) = -\frac{3}{32}y^5 + \frac{15}{64}y^4 - \frac{105}{512}y^3 + \frac{75}{1024}y^2 - \frac{9}{1024}y$$

e

$$u(1,y) = 0,$$

para $0 \leq y \leq \frac{1}{2}$.

De uma maneira em geral, as equações (4.12) e (4.13) podem ser escritas na forma

$$\frac{\partial}{\partial x}(u\phi) + \frac{\partial}{\partial y}(v\phi) = p^\phi + \frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} - S^\phi, \quad (4.18)$$

sendo ϕ , u ou v , $p^u = -\frac{\partial p}{\partial x}$ e $p^v = -\frac{\partial p}{\partial y}$.

Para transformar a equação (4.18), escrita no domínio físico para o domínio computacional, utilizaram-se as equações de transformação do sistema dado pela equação (2.40).

Considerando

$$\text{Termo I} = \frac{\partial}{\partial x}(u\phi) + \frac{\partial}{\partial y}(v\phi),$$

como

$$\frac{\partial}{\partial x}(u\phi) = (u\phi)_\xi \xi_x + (u\phi)_\eta \eta_x$$

e

$$\frac{\partial}{\partial y}(v\phi) = (v\phi)_\xi \xi_y + (v\phi)_\eta \eta_y,$$

então

$$\text{Termo I} = (u\phi)_\xi \xi_x + (u\phi)_\eta \eta_x + (v\phi)_\xi \xi_y + (v\phi)_\eta \eta_y. \quad (4.19)$$

Sabe-se que

$$\xi_x = y_\eta J, \quad (4.20)$$

$$\eta_x = -y_\xi J, \quad (4.21)$$

$$\xi_y = -x_\eta J \quad (4.22)$$

e

$$\eta_y = x_\xi J. \quad (4.23)$$

Substituindo as equações (4.20) a (4.23) na equação (4.19), tem-se

$$\text{Termo I} = (u\phi)_\xi [y_\eta J] + (u\phi)_\eta [-y_\xi J] + (v\phi)_\xi [-x_\eta J] + (v\phi)_\eta [x_\xi J],$$

ou seja,

$$\frac{\text{Termo I}}{J} = (u\phi)_\xi y_\eta - (u\phi)_\eta y_\xi - (v\phi)_\xi x_\eta + (v\phi)_\eta x_\xi.$$

Como

$$\begin{aligned} \frac{\text{Termo I}}{J} &= (u\phi)_\xi y_\eta + u\phi y_{\eta\xi} - u\phi y_{\eta\xi} - \\ &\quad (u\phi)_\eta y_\xi - u\phi y_{\xi\eta} + u\phi y_{\xi\eta} - \\ &\quad (v\phi)_\xi x_\eta - v\phi x_{\eta\xi} + v\phi x_{\eta\xi} + \\ &\quad (v\phi)_\eta x_\xi + v\phi x_{\xi\eta} - v\phi x_{\xi\eta}, \end{aligned}$$

então

$$\begin{aligned} \frac{\text{Termo I}}{J} &= (u\phi)_\xi y_\eta - u\phi y_{\eta\xi} - \\ &\quad (u\phi)_\eta y_\xi + u\phi y_{\xi\eta} - \\ &\quad (v\phi)_\xi x_\eta + v\phi x_{\eta\xi} + \\ &\quad (v\phi)_\eta x_\xi - v\phi x_{\xi\eta}, \end{aligned}$$

ou seja,

$$\begin{aligned} \frac{\text{Termo I}}{J} &= (u\phi y_\eta)_\xi - (u\phi y_\xi)_\eta - (v\phi x_\eta)_\xi + (v\phi x_\xi)_\eta \\ &= [(uy_\eta - vx_\eta)\phi]_\xi + [(vx_\xi - uy_\xi)\phi]_\eta \\ &= (U\phi)_\xi + (V\phi)_\eta, \end{aligned} \tag{4.24}$$

sendo $U = uy_\eta - vx_\eta$ e $V = vx_\xi - uy_\xi$ as componentes contravariantes do vetor velocidade, representadas em função das velocidades cartesianas (RAYNER, 1991). As componentes contravariantes U e V são vetores normais às linhas η e ξ , respectivamente, o que as torna atraentes para uso nas equações de conservação transformadas (ANDERSON; TANNEHILL; PLETCHER, 2013).

Ainda, considerando

$$\text{Termo II} = \frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2}, \tag{4.25}$$

a transformação dessa equação segue o procedimento usado para a equação de Poisson 2D. Assim,

$$\frac{\text{Termo II}}{J} = \frac{\partial}{\partial \xi} \left[J \left(\alpha \frac{\partial \phi}{\partial \xi} - \beta \frac{\partial \phi}{\partial \eta} \right) \right] + \frac{\partial}{\partial \eta} \left[J \left(\gamma \frac{\partial \phi}{\partial \eta} - \beta \frac{\partial \phi}{\partial \xi} \right) \right]. \tag{4.26}$$

Também, considerando o termo III de u

$$p^u = -\frac{\partial p}{\partial x},$$

a transformação desse termo resulta em

$$\begin{aligned} p^u &= -p_\xi \xi_x - p_\eta \eta_x \\ &= -p_\xi (y_\eta J) - p_\eta (-y_\xi J), \end{aligned}$$

ou seja,

$$\begin{aligned}\frac{p^u}{J} &= -y_\eta p_\xi + y_\xi p_\eta \\ &= -y_\eta p_\xi - py_{\eta\xi} + py_{\eta\xi} + y_\xi p_\eta + py_{\xi\eta} + py_{\xi\eta} \\ &= -(py_\eta)_\xi + py_{\eta\xi} + (py_\xi)_\eta - py_{\xi\eta}.\end{aligned}$$

Então,

$$\frac{p^u}{J} = -\frac{\partial}{\partial \xi}(py_\eta) + \frac{\partial}{\partial \eta}(py_\xi). \quad (4.27)$$

Agora, considerando o termo III de v

$$p^v = -\frac{\partial p}{\partial y},$$

a transformação desse termo resulta em

$$\begin{aligned}p^v &= -p_\xi \xi_y - p_\eta \eta_y \\ &= -p_\xi (-x_\eta J) - p_\eta (x_\xi J),\end{aligned}$$

ou seja,

$$\begin{aligned}\frac{p^v}{J} &= x_\eta p_\xi - x_\xi p_\eta \\ &= x_\eta p_\xi + px_{\eta\xi} - px_{\eta\xi} - x_\xi p_\eta - px_{\xi\eta} + px_{\xi\eta} \\ &= (px_\eta)_\xi - px_{\eta\xi} - (px_\xi)_\eta + px_{\xi\eta}.\end{aligned}$$

Então,

$$\frac{p^v}{J} = \frac{\partial}{\partial \xi}(px_\eta) - \frac{\partial}{\partial \eta}(px_\xi). \quad (4.28)$$

Juntando os termos transformados, equações (4.24), (4.26), (4.27) e (4.28), obtém-se a equação (4.18) transformada

$$\frac{\partial}{\partial \xi}(U\phi) + \frac{\partial}{\partial \eta}(V\phi) = +\frac{p^\phi}{J} + \frac{\partial}{\partial \xi} \left[J \left(\alpha \frac{\partial \phi}{\partial \xi} - \beta \frac{\partial \phi}{\partial \eta} \right) \right] + \frac{\partial}{\partial \eta} \left[J \left(\gamma \frac{\partial \phi}{\partial \eta} - \beta \frac{\partial \phi}{\partial \xi} \right) \right] - \frac{S^\phi}{J}, \quad (4.29)$$

em que os termos do lado esquerdo da equação (4.29) representam a advecção de ϕ através das faces ξ e η , respectivamente. No lado direito, o primeiro termo representa a pressão e os outros termos representam os fluxos difusivos de ϕ através das faces ξ e η , respectivamente, e o termo fonte.

4.2 MODELOS NUMÉRICOS

4.2.1 Equação de Poisson

Integrando a equação (4.11) sobre o volume de controle P , no plano transformado, conforme mostra a FIGURA 35, obtém-se

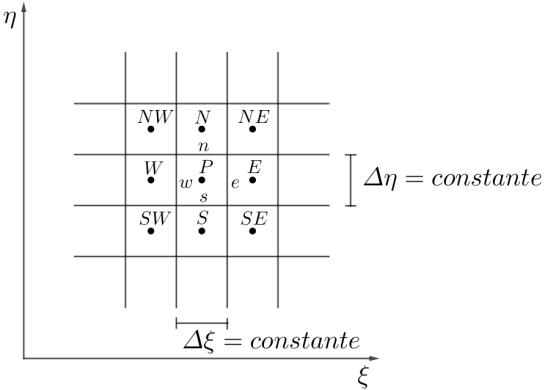
$$\int_s^n \int_w^e \left\{ \frac{\partial}{\partial \xi} \left[J \left(\alpha \frac{\partial T}{\partial \xi} - \beta \frac{\partial T}{\partial \eta} \right) \right] + \frac{\partial}{\partial \eta} \left[J \left(\gamma \frac{\partial T}{\partial \eta} - \beta \frac{\partial T}{\partial \xi} \right) \right] - \frac{S}{J} \right\} d\xi d\eta = 0,$$

que resulta em

$$\left[J \left(\alpha \frac{\partial T}{\partial \xi} - \beta \frac{\partial T}{\partial \eta} \right) \right]_w^e \Delta \eta + \left[J \left(\gamma \frac{\partial T}{\partial \eta} - \beta \frac{\partial T}{\partial \xi} \right) \right]_s^n \Delta \xi = \frac{S}{J} \Delta \xi \Delta \eta, \quad (4.30)$$

sendo w, e, s e n as faces oeste, leste, sul e norte do volume P , respectivamente.

FIGURA 35 – VOLUME DE CONTROLE E SEUS VIZINHOS USADOS PARA A DISCRETIZAÇÃO DA EQUAÇÃO TRANSFORMADA



Aproximando as derivadas da equação (4.30) com diferenças centrais, tem-se

$$\left[\frac{\partial T}{\partial \xi} \right]_e \simeq \frac{T_E - T_P}{\Delta \xi}, \quad (4.31)$$

$$\left[\frac{\partial T}{\partial \xi} \right]_w \simeq \frac{T_P - T_W}{\Delta \xi}, \quad (4.32)$$

$$\left[\frac{\partial T}{\partial \xi} \right]_n \simeq \frac{T_E + T_{NE} - T_W - T_{NW}}{4 \Delta \xi}, \quad (4.33)$$

$$\left[\frac{\partial T}{\partial \xi} \right]_s \simeq \frac{T_E + T_{SE} - T_W - T_{SW}}{4 \Delta \xi}, \quad (4.34)$$

$$\left[\frac{\partial T}{\partial \eta} \right]_n \simeq \frac{T_N - T_P}{\Delta \eta}, \quad (4.35)$$

$$\left[\frac{\partial T}{\partial \eta} \right]_s \simeq \frac{T_P - T_S}{\Delta \eta}, \quad (4.36)$$

$$\left[\frac{\partial T}{\partial \eta} \right]_e \simeq \frac{T_N + T_{NE} - T_S - T_{SE}}{4 \Delta \eta} \quad (4.37)$$

e

$$\left[\frac{\partial T}{\partial \eta} \right]_w \simeq \frac{T_N + T_{NW} - T_S - T_{SW}}{4 \Delta \eta}. \quad (4.38)$$

Substituindo as equações (4.31) a (4.38) na equação (4.30), obtém-se

$$\begin{aligned} J_e \alpha_e \frac{T_E - T_P}{\Delta\xi} \Delta\eta - J_w \alpha_w \frac{T_P - T_W}{\Delta\xi} \Delta\eta - J_e \beta_e \frac{T_N + T_{NE} - T_S - T_{SE}}{4\Delta\eta} \Delta\eta + \\ J_w \beta_w \frac{T_N + T_{NW} - T_S - T_{SW}}{4\Delta\eta} \Delta\eta + J_n \gamma_n \frac{T_N - T_P}{\Delta\eta} \Delta\xi - J_s \gamma_s \frac{T_P - T_S}{\Delta\eta} \Delta\xi - \\ J_n \beta_n \frac{T_E + T_{NE} - T_W - T_{NW}}{4\Delta\xi} \Delta\xi + J_s \beta_s \frac{T_E + T_{SE} - T_W - T_{SW}}{4\Delta\xi} \Delta\xi = S_p \frac{\Delta\xi \Delta\eta}{J_p} \end{aligned}, \quad (4.39)$$

sendo J_e , J_w , J_s e J_n o jacobiano avaliado nas faces leste, oeste, sul e norte do volume de controle P , respectivamente.

A equação (4.39) pode ser reescrita da seguinte forma

$$\begin{aligned} a_p T_p &= a_w T_W + a_e T_E + a_s T_S + a_n T_N + a_{sw} T_{SW} + a_{se} T_{SE} + a_{nw} T_{NW} + \\ a_{ne} T_{NE} + b_p &= \sum_{nb} a_{nb} T_{NB} + b_P, \end{aligned} \quad (4.40)$$

sendo

$$\begin{aligned} a_w &= J_w \alpha_w \frac{\Delta\eta}{\Delta\xi} + \frac{1}{4} J_n \beta_n - \frac{1}{4} J_s \beta_s, \\ a_e &= J_e \alpha_e \frac{\Delta\eta}{\Delta\xi} - \frac{1}{4} J_n \beta_n + \frac{1}{4} J_s \beta_s, \\ a_s &= J_s \gamma_s \frac{\Delta\xi}{\Delta\eta} + \frac{1}{4} J_e \beta_e - \frac{1}{4} J_w \beta_w, \\ a_n &= J_n \gamma_n \frac{\Delta\xi}{\Delta\eta} - \frac{1}{4} J_e \beta_e + \frac{1}{4} J_w \beta_w, \\ a_{sw} &= -\frac{1}{4} J_w \beta_w - \frac{1}{4} J_s \beta_s, \\ a_{se} &= \frac{1}{4} J_e \beta_e + \frac{1}{4} J_s \beta_s, \\ a_{nw} &= \frac{1}{4} J_w \beta_w + \frac{1}{4} J_n \beta_n, \\ a_{ne} &= -\frac{1}{4} J_e \beta_e - \frac{1}{4} J_n \beta_n, \\ a_p &= J_e \alpha_e \frac{\Delta\eta}{\Delta\xi} + J_w \alpha_w \frac{\Delta\eta}{\Delta\xi} + J_n \gamma_n \frac{\Delta\xi}{\Delta\eta} + J_s \gamma_s \frac{\Delta\xi}{\Delta\eta} = \sum_{nb=1}^8 a_{nb}, \end{aligned}$$

e

$$b_p = -S_p \frac{\Delta\xi \Delta\eta}{J_p},$$

sendo NB os 8 volumes vizinhos a P .

Aplicando as condições de contorno com a técnica dos volumes fictícios para o contorno leste, conforme mostra a FIGURA 36, tem-se

$$\frac{T_P + T_W}{2} = T_w,$$

sendo T_w a temperatura no contorno. Com isso,

$$T_P = T_W + 2T_w,$$

que na forma da equação (4.40), resulta em

$$a_e = a_s = a_n = a_{ne} = a_{nw} = a_{se} = a_{sw} = 0,$$

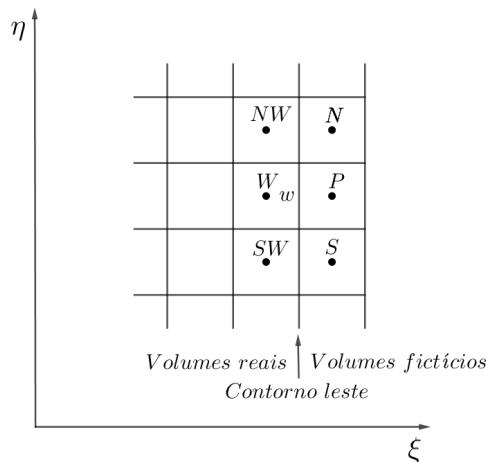
$$a_w = -1,$$

$$a_p = 1$$

e

$$b_p = 2T_w.$$

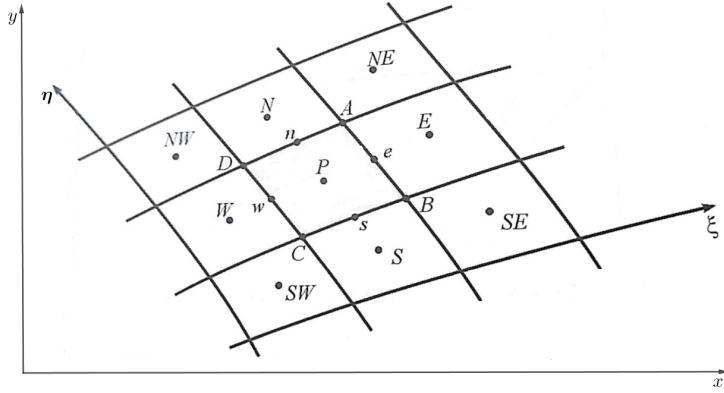
FIGURA 36 – CONTORNO LESTE COM OS VOLUMES FICTÍCIOS



O procedimento para obter os coeficientes e o termo fonte para os demais contornos é análogo.

Com a transformação das equações de conservação para o domínio (ξ, η) , é necessário conhecer as métricas da transformação, ou seja, as grandezas $(x_\eta)_P^e$, $(y_\eta)_P^e$, $(x_\xi)_P^n$ e $(y_\xi)_P^n$, no caso bidimensional. Por facilidade de notação, usa-se a face como superíndice e o ponto de referência como subíndice. Com essas grandezas é possível calcular as componentes do tensor métrico e o jacobiano da transformação.

A FIGURA 37 mostra um volume de controle P , com seus respectivos vértices A , B , C e D , usados para calcular as métricas da transformação numericamente.

FIGURA 37 – VOLUME P COM SEUS RESPECTIVOS VÉRTICES A, B, C E D 

Assim,

$$(x_\eta)_P^e = \frac{x_A - x_B}{\Delta\eta}, \quad (4.41)$$

$$(y_\eta)_P^e = \frac{y_A - y_B}{\Delta\eta}, \quad (4.42)$$

$$(x_\xi)_P^n = \frac{x_A - x_D}{\Delta\xi} \quad (4.43)$$

e

$$(y_\xi)_P^n = \frac{y_A - y_D}{\Delta\xi}. \quad (4.44)$$

As componentes do tensor métricos são calculadas da seguinte forma

$$\alpha_P^e = [(x_\eta)_P^e]^2 + [(y_\eta)_P^e]^2, \quad (4.45)$$

$$\gamma_P^n = [(x_\xi)_P^n]^2 + [(y_\xi)_P^n]^2, \quad (4.46)$$

$$\beta_P^e = (x_\xi)_P^n (x_\eta)_P^e + (y_\xi)_P^n (y_\eta)_P^e \quad (4.47)$$

com

$$(\psi_\xi)_P^e = \frac{1}{4} [(\psi_\xi)_P^n + (\psi_\xi)_E^n + (\psi_\xi)_S^n + (\psi_\xi)_{SE}^n], \quad (4.48)$$

sendo $\psi = x$ ou y , e

$$\beta_P^n = (x_\xi)_P^n (x_\eta)_P^n + (y_\xi)_P^n (y_\eta)_P^n, \quad (4.49)$$

com

$$(\psi_\eta)_P^n = \frac{1}{4} [(\psi_\eta)_P^e + (\psi_\eta)_N^e + (\psi_\eta)_W^e + (\psi_\eta)_{NW}^e]. \quad (4.50)$$

O jacobiano da transformação é dado por

$$J_P^c = [(x_\xi)_P^c (y_\eta)_P^c - (x_\eta)_P^c + (y_\xi)_P^c]^{-1}, \quad (4.51)$$

sendo c o centro do volume de controle e

$$(\psi_\xi)_P^c = \frac{1}{2} [(\psi_\xi)_P^n + (\psi_\xi)_S^n] \quad (4.52)$$

e

$$(\psi_\eta)_P^c = \frac{1}{2} [(\psi_\eta)_P^e + (\psi_\eta)_W^e]. \quad (4.53)$$

Também,

$$(J)_P^e = \frac{1}{2} [(J)_P^c + (J)_E^c] \quad (4.54)$$

e

$$(J)_P^n = \frac{1}{2} [(J)_P^c + (J)_N^c]. \quad (4.55)$$

4.2.2 Equações de Burgers

Integrando a equação (4.29) sobre o volume de controle P , no plano transformado, obtém-se

$$\int_s^n \int_w^e \left[\frac{\partial}{\partial \xi} (U\phi) + \frac{\partial}{\partial \eta} (V\phi) \right] d\xi d\eta = \int_s^n \int_w^e \left\{ \frac{p^\phi}{J} + \frac{\partial}{\partial \xi} \left[J \left(\alpha \frac{\partial \phi}{\partial \xi} - \beta \frac{\partial \phi}{\partial \eta} \right) \right] + \frac{\partial}{\partial \eta} \left[J \left(\gamma \frac{\partial \phi}{\partial \eta} - \beta \frac{\partial \phi}{\partial \xi} \right) \right] - \frac{S^\phi}{J} \right\} d\xi d\eta,$$

que resulta em

$$(U\phi)_w^e \Delta\eta + (V\phi)_s^n \Delta\xi = \left[J \left(\alpha \frac{\partial \phi}{\partial \xi} - \beta \frac{\partial \phi}{\partial \eta} \right) \right]_w^e \Delta\eta + \left[J \left(\gamma \frac{\partial \phi}{\partial \eta} - \beta \frac{\partial \phi}{\partial \xi} \right) \right]_s^n \Delta\xi + L \left[\frac{p^\phi}{J} \right]_P - \frac{S^\phi}{J} \Delta\xi \Delta\eta,$$

ou seja,

$$\begin{aligned} U_e \phi_e \Delta\eta - U_w \phi_w \Delta\eta + V_n \phi_n \Delta\xi - V_s \phi_s \Delta\xi &= J_e \left\{ \alpha_e \left[\frac{\partial \phi}{\partial \xi} \right]_e - \beta_e \left[\frac{\partial \phi}{\partial \eta} \right]_e \right\} \Delta\eta - \\ &\quad J_w \left\{ \alpha_w \left[\frac{\partial \phi}{\partial \xi} \right]_w - \beta_w \left[\frac{\partial \phi}{\partial \eta} \right]_w \right\} \Delta\eta + \\ &\quad J_n \left\{ \gamma_n \left[\frac{\partial \phi}{\partial \eta} \right]_n - \beta_n \left[\frac{\partial \phi}{\partial \xi} \right]_n \right\} \Delta\xi - \\ &\quad J_s \left\{ \gamma_s \left[\frac{\partial \phi}{\partial \eta} \right]_s - \beta_n \left[\frac{\partial \phi}{\partial \xi} \right]_s \right\} \Delta\xi - \\ &\quad L \left[\frac{p^\phi}{J} \right]_P - \frac{S^\phi}{J} \Delta\xi \Delta\eta, \end{aligned} \quad (4.56)$$

sendo $L \left[\frac{p^\phi}{J} \right]_P$ a aproximação da integral em ξ e η de p^ϕ no volume P e w, e, s e n as faces oeste, leste, sul e norte do volume P , respectivamente. Neste trabalho, usa-se arranjo colocalizado (variáveis localizadas no centro do volume de controle), conforme mostra a

FIGURA 35. Isso porque o uso de arranjo desencontrado (algumas variáveis localizadas no centro do volume e outras localizadas nas faces do mesmo) é mais complexo quando usa-se sistemas de coordenadas curvilíneas, devido ao armazenamento computacional extra necessário para as informações geométricas sobre os diferentes volumes de controle (MARCHI; MALISKA, 1994).

Aproximando-se as derivadas diretas e cruzadas do lado direito da equação (4.56) com diferenças centrais, tem-se

$$\left[\frac{\partial \phi}{\partial \xi} \right]_e \simeq \frac{\phi_E - \phi_P}{\Delta \xi}, \quad (4.57)$$

$$\left[\frac{\partial \phi}{\partial \xi} \right]_w \simeq \frac{\phi_P - \phi_W}{\Delta \xi}, \quad (4.58)$$

$$\left[\frac{\partial \phi}{\partial \xi} \right]_n \simeq \frac{\phi_E + \phi_{NE} - \phi_W - \phi_{NW}}{4 \Delta \xi}, \quad (4.59)$$

$$\left[\frac{\partial \phi}{\partial \xi} \right]_s \simeq \frac{\phi_E + \phi_{SE} - \phi_W - \phi_{SW}}{4 \Delta \xi}, \quad (4.60)$$

$$\left[\frac{\partial \phi}{\partial \eta} \right]_n \simeq \frac{\phi_N - \phi_P}{\Delta \eta}, \quad (4.61)$$

$$\left[\frac{\partial \phi}{\partial \eta} \right]_s \simeq \frac{\phi_P - \phi_S}{\Delta \eta}, \quad (4.62)$$

$$\left[\frac{\partial \phi}{\partial \eta} \right]_e \simeq \frac{\phi_N + \phi_{NE} - \phi_S - \phi_{SE}}{4 \Delta \eta} \quad (4.63)$$

e

$$\left[\frac{\partial \phi}{\partial \eta} \right]_w \simeq \frac{\phi_N + \phi_{NW} - \phi_S - \phi_{SW}}{4 \Delta \eta}. \quad (4.64)$$

Os termos advectivos (lado esquerdo da equação (4.56)) foram aproximados com UDS (MALISKA, 2010), sendo

$$\phi_e = \left(\frac{1}{2} + \lambda_e \right) \phi_P + \left(\frac{1}{2} - \lambda_e \right) \phi_E \quad (4.65)$$

$$\phi_w = \left(\frac{1}{2} + \lambda_w \right) \phi_W + \left(\frac{1}{2} - \lambda_w \right) \phi_P \quad (4.66)$$

$$\phi_n = \left(\frac{1}{2} + \lambda_n \right) \phi_P + \left(\frac{1}{2} - \lambda_n \right) \phi_N \quad (4.67)$$

e

$$\phi_s = \left(\frac{1}{2} + \lambda_s \right) \phi_S + \left(\frac{1}{2} - \lambda_s \right) \phi_P, \quad (4.68)$$

em que

$$\lambda_e = \frac{1}{2}\text{sign}(U_e) \quad (4.69)$$

$$\lambda_w = \frac{1}{2}\text{sign}(U_w) \quad (4.70)$$

$$\lambda_n = \frac{1}{2}\text{sign}(V_n) \quad (4.71)$$

e

$$\lambda_s = \frac{1}{2}\text{sign}(V_s), \quad (4.72)$$

sendo sign a função sinal.

Substituindo as equações (4.57) a (4.68) na equação (4.56), tem-se

$$\begin{aligned} U_e & \left[\left(\frac{1}{2} + \lambda_e \right) \phi_P + \left(\frac{1}{2} - \lambda_e \right) \phi_E \right] \Delta\eta - U_w \left[\left(\frac{1}{2} + \lambda_w \right) \phi_W + \left(\frac{1}{2} - \lambda_w \right) \phi_P \right] \phi_w \Delta\eta + \\ V_n & \left[\left(\frac{1}{2} + \lambda_n \right) \phi_P + \left(\frac{1}{2} - \lambda_n \right) \phi_N \right] \Delta\xi - V_s \left[\left(\frac{1}{2} + \lambda_s \right) \phi_S + \left(\frac{1}{2} - \lambda_s \right) \phi_P \right] \Delta\xi = \\ J_e & \left[\alpha_e \frac{\phi_E - \phi_P}{\Delta\xi} - \beta_e \frac{\phi_N + \phi_{NE} - \phi_S - \phi_{SE}}{4\Delta\eta} \right] \Delta\eta - \\ J_w & \left[\alpha_w \frac{\phi_P - \phi_W}{\Delta\xi} - \beta_w \frac{\phi_N + \phi_{NW} - \phi_S - \phi_{SW}}{4\Delta\eta} \right] \Delta\eta + \\ J_n & \left[\gamma_n \frac{\phi_N - \phi_P}{\Delta\eta} - \beta_n \frac{\phi_E + \phi_{NE} - \phi_W - \phi_{NW}}{4\Delta\xi} \right] \Delta\xi - \\ J_s & \left[\gamma_s \frac{\phi_P - \phi_S}{\Delta\eta} - \beta_n \frac{\phi_E + \phi_{NE} - \phi_W - \phi_{NW}}{4\Delta\xi} \right] \Delta\xi + L \left[\frac{p^\phi}{J} \right]_P - \frac{S^\phi}{J} \Delta\xi \Delta\eta \end{aligned}, \quad (4.73)$$

que pode ser reescrita como

$$\begin{aligned} a_p \phi_p &= a_w \phi_W + a_e \phi_E + a_s \phi_S + a_n \phi_N + a_{sw} \phi_{SW} + a_{se} \phi_{SE} + a_{nw} \phi_{NW} + \\ & a_{ne} \phi_{NE} + b_p = \sum_{nb} a_{nb} \phi_{NB} + b_P, \end{aligned} \quad (4.74)$$

sendo

$$a_w = J_w \alpha_w \frac{\Delta\eta}{\Delta\xi} + \frac{1}{4} J_n \beta_n - \frac{1}{4} J_s \beta_s + U_w \left(\frac{1}{2} + \lambda_w \right) \Delta\eta,$$

$$a_e = J_e \alpha_e \frac{\Delta\eta}{\Delta\xi} - \frac{1}{4} J_n \beta_n + \frac{1}{4} J_s \beta_s - U_e \left(\frac{1}{2} - \lambda_e \right) \Delta\eta,$$

$$a_s = J_s \gamma_s \frac{\Delta\xi}{\Delta\eta} + \frac{1}{4} J_e \beta_e - \frac{1}{4} J_w \beta_w + V_s \left(\frac{1}{2} + \lambda_s \right) \Delta\xi,$$

$$a_n = J_n \gamma_n \frac{\Delta\xi}{\Delta\eta} - \frac{1}{4} J_e \beta_e + \frac{1}{4} J_w \beta_w + V_n \left(\frac{1}{2} - \lambda_s \right) \Delta\xi,$$

$$a_{sw} = -\frac{1}{4} J_w \beta_w - \frac{1}{4} J_s \beta_s,$$

$$a_{se} = \frac{1}{4} J_e \beta_e + \frac{1}{4} J_s \beta_s,$$

$$\begin{aligned} a_{nw} &= \frac{1}{4}J_w\beta_w + \frac{1}{4}J_n\beta_n, \\ a_{ne} &= -\frac{1}{4}J_e\beta_e - \frac{1}{4}J_n\beta_n, \end{aligned}$$

e

$$a_p = a_w + a_e + a_s + a_n.$$

Os termos fontes de u e v são diferentes, dados por

$$\begin{aligned} b_p^u &= \left[\frac{(p_N + p_P)}{2}(y_\xi)_P^n - \frac{(p_P + p_S)}{2}(y_\xi)_P^s \right] \Delta\xi - \\ &\quad \left[\frac{(p_P + p_E)}{2}(y_\eta)_P^e - \frac{(p_P + p_W)}{2}(y_\eta)_P^w \right] \Delta\eta \end{aligned}$$

e

$$\begin{aligned} b_p^v &= -S_P \frac{\Delta\xi\Delta\eta}{J_P} + \left[\frac{(p_P + p_E)}{2}(x_\eta)_P^e - \frac{(p_P + p_W)}{2}(x_\eta)_P^w \right] \Delta\eta - \\ &\quad \left[\frac{(p_P + p_N)}{2}(x_\xi)_P^n - \frac{(p_P + p_S)}{2}(x_\xi)_P^s \right] \Delta\xi. \end{aligned}$$

As métricas e o jacobiano da transformação são calculados como na subseção 4.2.1.

4.3 CÁLCULO DAS MÉTRICAS DE TRANSFORMAÇÃO DE COORDENADAS NAS MALHAS GROSSAS

Para os problemas resolvidos, as métricas de transformação de coordenadas nas malhas grossas foram obtidas fazendo-se a restrição das métricas de transformação obtidas na malha mais fina, que é uma abordagem sugerida por Smith, Cope e Vanka (1993). Neste trabalho, conseguiu-se a convergência do método *multigrid* calculando as métricas de transformação de coordenadas da seguinte forma

$$(\psi_\eta)_F^e = (\psi_\eta)_{16}^e, \quad (4.75)$$

e

$$(\psi_\xi)_F^n = (\psi_\xi)_{21}^n, \quad (4.76)$$

sendo $\psi = x$ ou y , F ponto da malha grossa e 16 e 21 pontos da malha fina, conforme mostra a FIGURA 38.

As componentes do tensor métrico na malha grossa são calculadas usando as seguintes expressões

$$\alpha_F^e = [(x_\eta)_F^e]^2 + [(y_\eta)_F^e]^2, \quad (4.77)$$

$$\gamma_F^n = [(x_\xi)_F^n]^2 + [(y_\xi)_F^n]^2 \quad (4.78)$$

e

$$\beta_F^e = (x_\xi)_F^e (x_\eta)_F^e + (y_\xi)_F^e (y_\eta)_F^e \quad (4.79)$$

com

$$(\psi_\xi)_F^e = \frac{1}{4} [(\psi_\xi)_F^n + (\psi_\xi)_G^n + (\psi_\xi)_B^n + (\psi_\xi)_C^n] \quad (4.80)$$

e

$$(\psi_\eta)_F^n = \frac{1}{4} [(\psi_\eta)_F^e + (\psi_\eta)_I^e + (\psi_\eta)_D^e + (\psi_\eta)_H^e], \quad (4.81)$$

sendo G, B, C, I, D e H pontos da malha grossa.

O jacobiano da transformação na malha grossa é calculado da seguinte forma

$$J_F^c = [(x_\xi)_F^c (y_\eta)_F^c - (x_\eta)_F^c + (y_\xi)_F^c]^{-1}, \quad (4.82)$$

sendo c o centro do volume de controle,

$$(\psi_\xi)_F^c = \frac{1}{2} [(\psi_\xi)_F^n + (\psi_\xi)_B^n] \quad (4.83)$$

e

$$(\psi_\eta)_F^c = \frac{1}{2} [(\psi_\eta)_F^e + (\psi_\eta)_D^e]. \quad (4.84)$$

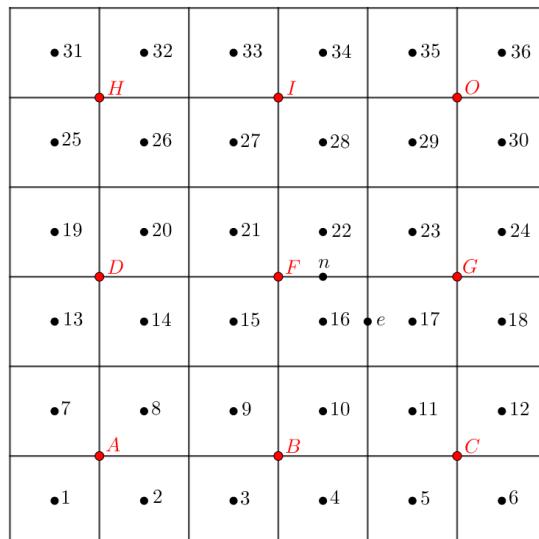
Também,

$$(J)_F^e = \frac{1}{2} [(J)_F^c + (J)_G^c] \quad (4.85)$$

e

$$(J)_F^n = \frac{1}{2} [(J)_F^c + (J)_I^c]. \quad (4.86)$$

FIGURA 38 – ESQUEMA DE MALHAS NO DOMÍNIO COMPUTACIONAL PARA RESTRIÇÃO DAS MÉTRICAS DA TRANSFORMAÇÃO DE COORDENADAS DA MALHA FINA (PONTOS 1, 2, 3, ..., 36) PARA A MALHA IMEDIATAMENTE MAIS GROSSA (PONTOS A, B, C, D, F, G, H, I E O)



5 DETALHES COMPUTACIONAIS

Neste capítulo são apresentados os detalhes computacionais adotados para cada problema estudado.

5.1 FATOR DE CONVERGÊNCIA EMPÍRICO

O fator de convergência do método iterativo dado pela equação (2.10) é definido por

$$\rho(\mathbf{T}) = \max |\lambda(\mathbf{T})|,$$

em que $\lambda(\mathbf{T})$ representa os autovalores da matriz \mathbf{T} (matriz de iteração). O fator de convergência $\rho(\mathbf{T})$ é o raio espectral da matriz \mathbf{T} e indica, aproximadamente, a pior redução do erro a cada passo iterativo (BRIGGS; HENSON; McCORMICK, 2000). O método iterativo dado pela equação (2.10) é convergente se e somente se $\rho(\mathbf{T}) < 1$, sendo $\rho(\mathbf{T})$ o fator de convergência do método iterativo (ISAACSON; KELLER, 1994). Segundo Briggs, Henson e McCormick (2000), quando $\rho(\mathbf{T})$ se aproxima de 1, a taxa de convergência diminui e quando os valores de $\rho(\mathbf{T})$ são pequenos (isto é, $\rho(\mathbf{T})$ positivo e próximo de zero), tem-se uma alta taxa de convergência dos métodos iterativos.

A convergência do método *multigrid* pode ser analisada usando-se o fator de convergência empírico, que é calculado com base no resíduo. Tal fator, pode ser aproximado numericamente como descrito em Trottenberg, Oosterlee e Schuller (2001) por

$$\rho^{(k)} = \frac{\|\mathbf{r}^{(k)}\|}{\|\mathbf{r}^{(k-1)}\|}, \quad (5.1)$$

sendo $k = 1, 2, \dots$ o número de iteradas ou ciclos *multigrid* realizados ou fazendo-se a média geométrica dos fatores de convergência dados pela equação (5.1), que é definida por

$$\rho_m = \sqrt[k]{\rho^{(1)} \rho^{(2)} \cdots \rho^{(k-1)} \rho^{(k)}} = \sqrt[k]{\frac{\|\mathbf{r}^{(k)}\|}{\|\mathbf{r}^{(0)}\|}}. \quad (5.2)$$

Esse fator ρ_m é denominado fator de convergência médio empírico e representa o fator médio de redução do resíduo sobre k iterações.

5.2 RESOLUÇÃO DO SISTEMA DE EQUAÇÕES ALGÉBRICAS

Os sistemas de equações algébricas obtidos para cada um dos modelos matemáticos (Poisson e Burgers), com o domínio dado pela FIGURA 34, foram resolvidos com o método *multigrid* geométrico, usando o esquema FAS e ciclo V. Primeiramente, verificou-se o fator de convergência médio empírico para todos os *solvers* empregados neste estudo (GS-Lex, η -linha-GS, MSI e MILU). Posteriormente, cada um dos modelos matemáticos foram

resolvidos empregando o método *multigrid* geométrico, usando o esquema FAS, ciclo V e o *solver* MSI. Em seguida, para o modelo dado pelas equações de Burgers, o desempenho do MSI foi comparado com o do MILU. Para a geração de malhas curvilíneas estruturadas não ortogonais empregam-se os métodos algébrico, que emprega interpolação de Lagrange e diferencial, que utiliza sistema de equações elípticas.

O operador de restrição das aproximações da solução foi obtido fazendo-se a média aritmética dos valores da propriedade dos quatro volumes da malha fina. O processo de restrição do resíduo foi feito somando-se os resíduos relativos dos volumes de controle da malha fina que correspondem àquele da malha mais grossa. A restrição das métricas de transformação foi feita conforme descrito na seção 4.3. A prolongação da correção foi feita através de interpolação bilinear.

Neste trabalho, utiliza-se a razão de engrossamento padrão das malhas, ou seja, $r = 2$. Nesse caso, o tamanho do volume da malha imediatamente mais grossa é duas vezes o tamanho do volume da malha fina. O número de iterações internas do *solver* empregadas na pré-suavização e pós-suavização são iguais, ou seja, $\nu = \nu_1 = \nu_2$.

5.3 DADOS DA IMPLEMENTAÇÃO

Todos os códigos deste trabalho foram implementados na linguagem FORTRAN 95 com o compilador versão 11.1 da Intel. Para cada problema, foi escrito um código computacional compilado na versão *release*, projeto tipo *Console Application*.

Todas as simulações foram realizadas no microcomputador CFD36 do laboratório de Experimentação numérica (LENA) da UFPR, com processador Intel(R) Core(TM) i7 – 3770, CPU 3.40 GHz, 16 GB RAM e sistema operacional Windows xp 64 bits, usando aritmética de dupla precisão.

Mesmo nas malhas mais refinadas, as simulações foram realizadas sem a necessidade do uso de memória virtual. A memória computacional foi monitorada através do gerenciador de tarefas do Windows. O tempo de CPU foi medido usando-se a função TIMEF da biblioteca PORTLIB do FORTRAN 95 e corresponde ao tempo para atribuir estimativa inicial, calcular os coeficientes e termos fonte, resolver os sistemas de equações e verificar o critério de convergência.

5.4 DEFINIÇÃO DA TOLERÂNCIA

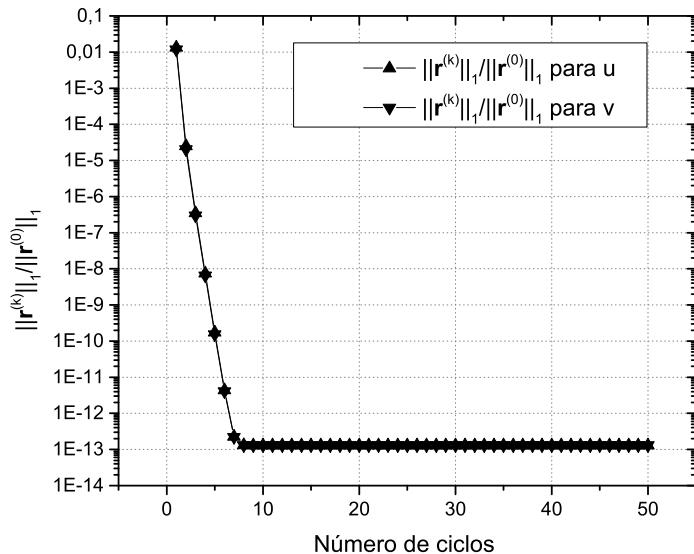
O critério de parada adotado para interromper as iterações externas (ciclos V) é a norma l_1 do resíduo na iteração atual (FERZIGER; PERIC, 2002) adimensionalizada pela norma l_1 do resíduo na estimativa inicial, isto é,

$$\frac{\|\mathbf{r}^{(k)}\|_1}{\|\mathbf{r}^{(0)}\|_1} < \epsilon,$$

sendo ϵ uma tolerância adotada.

Para determinar a tolerância ϵ a ser empregada nas simulações utiliza-se a malha mais fina possível, $N = 4096 \times 4096$, considerando as limitações de memória física computacional, sendo $N = N_\xi N_\eta$, em que N_ξ e N_η são o número de volumes nas direções ξ e η , respectivamente. O processo iterativo foi executado até eliminar o erro de iteração para os dois modelos matemáticos. A FIGURA 39 apresenta o comportamento da norma l_1 do resíduo, calculada em cada iteração, adimensionalizada pelo resíduo na estimativa inicial para a malha mais fina para as equações de Burgers e malha gerada por interpolações de Lagrange.

FIGURA 39 – COMPORTAMENTO DA NORMA l_1 DO RESÍDUO NA ITERAÇÃO ATUAL ADIMENSIONALIZADA PELO RESÍDUO NA ESTIMATIVA INICIAL EM FUNÇÃO DO NÚMERO DE CICLOS V PARA UMA MALHA $N = 4096 \times 4096$ GERADA POR INTERPOLAÇÕES DE LAGRANGE



Verifica-se que a norma decresce nos primeiros ciclos e estabiliza em 10^{-13} . O valor mínimo da tolerância a ser empregada nas simulações deve ser de pelo menos uma ordem de grandeza maior do que a norma l_1 adimensionalizada pelo resíduo na estimativa inicial, arredondando-se para cima (OLIVEIRA, 2010). Com essa verificação, obtém-se uma tolerância de 10^{-11} , o que garante que a norma do resíduo adimensionalizada não ficará oscilando em torno do erro de máquina. Para as equações de Burgers com malhas geradas empregando-se equações elípticas e para o modelo matemático de Poisson com malhas geradas empregando-se interpolações de Lagrange e equações elípticas, verificou-se que a norma l_1 do resíduo na iteração atual adimensionalizada pelo resíduo na estimativa inicial, para a malha mais fina, também estabilizou em 10^{-13} . Para todos os problemas, adota-se a estimativa inicial com valor nulo para as variáveis dependentes.

6 RESULTADOS

Neste capítulo é apresentada a análise dos parâmetros investigados para o método *multigrid* geométrico em malhas curvilíneas estruturadas não ortogonais. Foram feitas centenas de simulações computacionais, com a finalidade de investigar a influência causada pelos parâmetros sobre o tempo de CPU. Os parâmetros investigados foram o fator de convergência médio empírico (ρ_m) para todos os *solvers* empregados (GS-Lex, η -linha-GS, MSI e MILU), o número de iterações internas dos *solvers* MSI e MILU (ν), o número de níveis de malhas (L) e o número de incógnitas (N) para os problemas com malhas geradas usando interpolação de Lagrange e equações elípticas. As análises do método *multigrid* foram feitas nos problemas bidimensionais de Poisson e de Burgers com o MVF, considerando o domínio no formato de L, dado pela FIGURA 34.

As simulações computacionais foram realizadas com um número de malhas L tal que $1 \leq L \leq L_{\max}$, sendo $L = 1$ correspondente ao *singlegrid*. Nos casos em que o tempo de CPU foi inferior a 10 segundos, as simulações foram repetidas até atingir 10 segundos. O tempo de CPU foi então calculado tomando-se a média aritmética dos tempos obtidos em cada repetição.

Para a verificação dos códigos computacionais, consideram-se as ordens efetiva e aparente do erro de discretização e o comportamento do erro de discretização mediante o refino de malha computacional. Essas informações são apresentadas no Apêndice A deste texto.

Os problemas foram resolvidos com o *solver* MSI e o método *multigrid* com o FAS e ciclo V. Para o problema de Burgers também emprega-se o *solver* MILU. Os valores ótimos para os parâmetros foram investigados. No estudo dos valores ótimos para os parâmetros ν e L foram considerados os quatro maiores tamanhos de malhas, 512×512 , 1024×1024 , 2048×2048 e 4096×4096 incógnitas. Nas análises sobre o tamanho do problema (N) foram consideradas malhas de 16×16 a 4096×4096 incógnitas, que é a malha mais fina possível, dadas as limitações de memória computacional. Não foram encontrados na literatura trabalhos que resolvam problemas em malhas como aqui refinadas, para malhas geradas usando interpolação de Lagrange e equações de Laplace. Os resultados do tempo de CPU empregado na resolução dos problemas com o algoritmo FAS foram comparados com os respectivos resultados obtidos com o algoritmo SG e o desempenho do *multigrid* foi avaliado para os dois problemas resolvidos. Com os parâmetros ótimos definidos para o problema de Burgers resolvido com os *solvers* MSI e MILU e o método *multigrid*, para os diferentes geradores de malha empregados, foi feita a comparação do desempenho dos algoritmos compostos com os *solvers* MSI e MILU e com os parâmetros ótimos. O desempenho do método *multigrid* em relação ao *singlegrid* e também do *multigrid* com o *solver* MSI em relação ao *multigrid* com o *solver* MILU, com seus diferentes geradores de

malhas, foi medido pelo fator de aceleração ou *speedup* (s_p).

6.1 ANÁLISE DO FATOR DE CONVERGÊNCIA MÉDIO EMPÍRICO

Com o intuito de avaliar o efeito causado pelo descarte dos elementos durante o processo da decomposição *LU* incompleta, realizou-se um estudo do ρ_m para os problemas de Poisson e de Burgers, para malhas geradas com interpolação de Lagrange e equações elípticas. Para esta análise, consideraram-se diferentes valores de σ , com $0 \leq \sigma \leq 0,95$ e $-0,25 \leq \sigma \leq 1$, para o *multigrid* com o MSI e o MILU, respectivamente. Além disso, considera-se o número de iterações internas dos *solvers* igual a três, ou seja, $\nu = \nu_1 = \nu_2 = 3$ e $L = L_{\max}$.

As FIGURAS 40 e 41 apresentam os resultados do ρ_m na resolução dos problemas de tamanhos 512×512 , 1024×1024 e 2048×2048 , para ambos os geradores de malhas.

FIGURA 40 – FATOR DE CONVERGÊNCIA MÉDIO EMPÍRICO EM MALHAS GERADAS PELA INTERPOLAÇÃO DE LAGRANGE

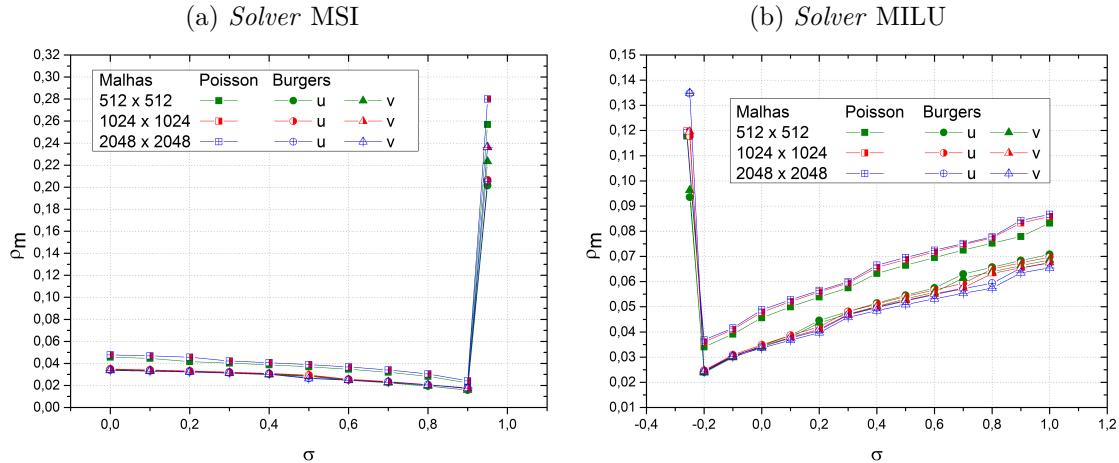
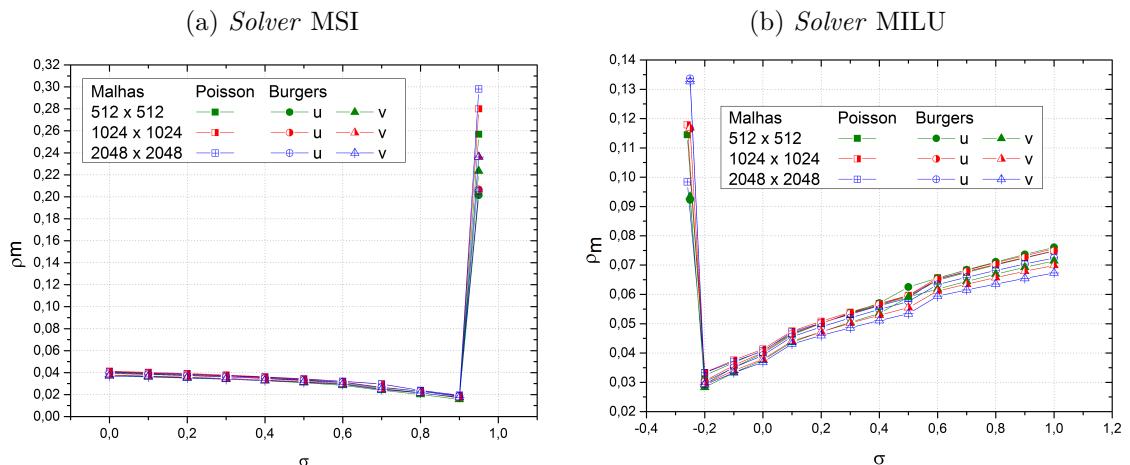


FIGURA 41 – FATOR DE CONVERGÊNCIA MÉDIO EMPÍRICO EM MALHAS GERADAS POR EQUAÇÕES ELÍPTICAS

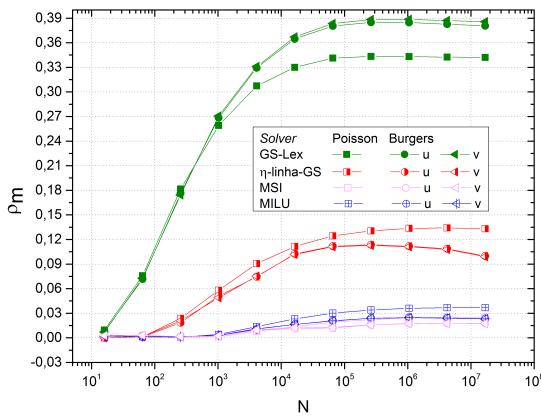


Como pode-se observar nas FIGURAS 40 e 41, os menores valores para ρ_m , independentemente do problema e do tamanho do problema, ocorrem quando $\sigma \approx 0,9$ e $\sigma \approx -0,2$ para os *solvers* MSI e MILU, respectivamente. Desta forma, daqui para frente serão empregados $\sigma = 0,9$ e $\sigma = -0,2$ nas formulações do MSI e MILU, respectivamente.

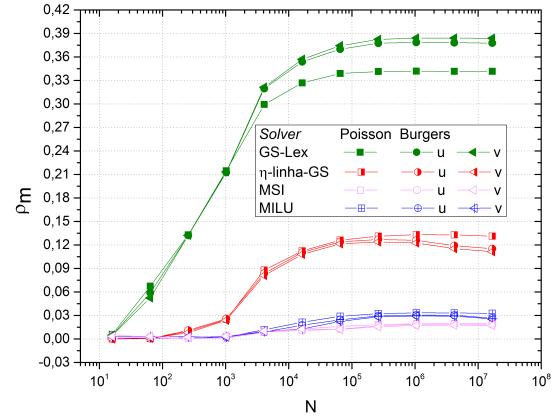
A FIGURA 42 apresenta o fator de convergência médio empírico para os *solvers* empregados neste estudo e problemas de tamanhos 2×2 a 4096×4096 . Os resultados apresentados são para malhas geradas usando interpolação de Lagrange e equações elípticas.

FIGURA 42 – FATOR DE CONVERGÊNCIA MÉDIO EMPÍRICO VERSUS N

(a) Interpolação de Lagrange



(b) Equações elípticas



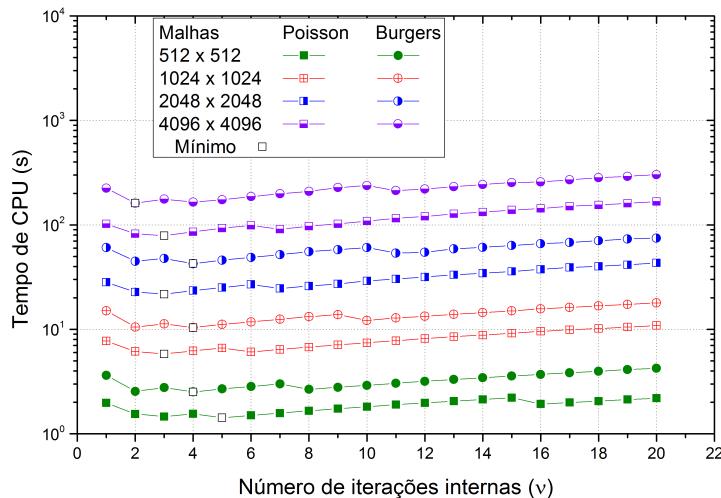
Como pode-se observar na FIGURA 42, o *solver* linha, η -linha-GS, tem o melhor fator de convergência médio empírico se comparado com o *solver* ponto-a-ponto, GS-Lex. Isso ocorreu porque as equações envolvem forte acoplamento de incógnitas na direção η . Os trabalhos de Trottenberg, Oosterlee e Schuller (2001) e Wesseling e Oosterlee (2001) apontam que os *solvers* linha, para problemas com malhas curvilíneas, apresentam melhores fatores de convergência se comparados com o *solver* GS-Lex. Entretanto, comparando todos os *solvers* empregados neste trabalho, os *solvers* MSI e MILU apresentam os melhores fatores de convergência médio empírico, ou seja, apresentam fatores ρ_m muito próximos de zero e distantes da unidade ($\rho_m \ll 1$), que é uma propriedade desejável. Por exemplo, na malha mais refinada ($N = 4096 \times 4096$), tem-se bons fatores de convergência ($\rho_m \approx 0,03$) para os problemas de Poisson e de Burgers com malhas geradas usando interpolação de Lagrange e equações elípticas. Segundo Trottenberg, Oosterlee e Schuller (2001), Wesseling e Oosterlee (2001) e Stüben (2001), os *solvers* do tipo ILU são bons suavizadores para problemas que usam malhas curvilíneas. Pinto e Marchi (2006) usaram o método MSI com o FAS para resolver a equação de Laplace bidimensional anisotrópica, usando sistema de coordenadas cartesianas e o MDF. Concluíram que o MSI é mais rápido que o ADI e GS-Lex. Dessa forma, com o intuito de desenvolver algoritmos eficientes para a resolução do problema de Burgers, primeiramente será feito um estudo com o *solver* MSI e, posteriormente, será feita a comparação dos *solvers* MSI e MILU.

6.2 NÚMERO DE ITERAÇÕES INTERNAS DO *SOLVER*

Nesta seção é apresentado o estudo do número ótimo de iterações internas do *solver* MSI sobre o tempo de CPU. Para todas as malhas consideradas são realizadas simulações com um número fixo de iterações internas do *solver*, variando de $\nu = 1$ até 20, para confirmar a tendência do parâmetro. O valor de ν que resulta no menor tempo de CPU é considerado o número ótimo de iterações internas do *solver*, denotado por $\nu_{óptimo}$. Em todas as simulações realizadas para este parâmetro empregou-se $L = L_{máx}$, isto é, o método *multigrid* percorre todos os níveis de malha do ciclo V.

As FIGURAS 43 e 44 ilustram a influência do número de iterações internas do *solver* para as equações de Poisson e de Burgers para malhas geradas usando interpolação de Lagrange e equações elípticas, respectivamente. Em cada curva indica-se o valor de ν que resulta no menor tempo de CPU usando o símbolo \square .

FIGURA 43 – EFEITO DO NÚMERO DE ITERAÇÕES INTERNAS SOBRE O TEMPO DE CPU (SEGUNDOS) PARA MALHAS GERADAS USANDO INTERPOLAÇÃO DE LAGRANGE



Como pode ser observado na FIGURA 43 para o problema de Poisson, o menor tempo de CPU é obtido com três iterações internas, ou seja, $\nu_{óptimo} = 3$ para as três malhas mais finas e $\nu_{óptimo} = 5$ para a malha mais grossa. Para o problema de Burgers, o menor tempo de CPU é obtido com duas ou quatro iterações internas do *solver*. Portanto, para o problema de Burgers, $\nu_{óptimo} = 2$ ou 4.

A FIGURA 44 ilustra a influência do número de iterações internas do *solver* para as equações de Poisson e de Burgers, para malhas geradas usando equações elípticas. Como pode ser observado, para o problema de Poisson, o menor tempo de CPU é obtido com duas iterações internas, ou seja, $\nu_{óptimo} = 2$, para as malhas mais finas, e $\nu_{óptimo} = 4$, para a malha mais grossa. Para o problema de Burgers, o menor tempo de CPU é obtido com duas ou quatro iterações internas. Portanto, para o problema de Burgers, $\nu_{óptimo} = 2$ ou 4. Os valores de $\nu_{óptimo}$ obtidos para malhas geradas usando interpolação de Lagrange e equações

elípticas, estão resumidos na TABELA 1.

FIGURA 44 – EFEITO DO NÚMERO DE ITERAÇÕES INTERNAS SOBRE O TEMPO DE CPU (SEGUNDOS) PARA MALHAS GERADAS USANDO EQUAÇÕES ELÍPTICAS

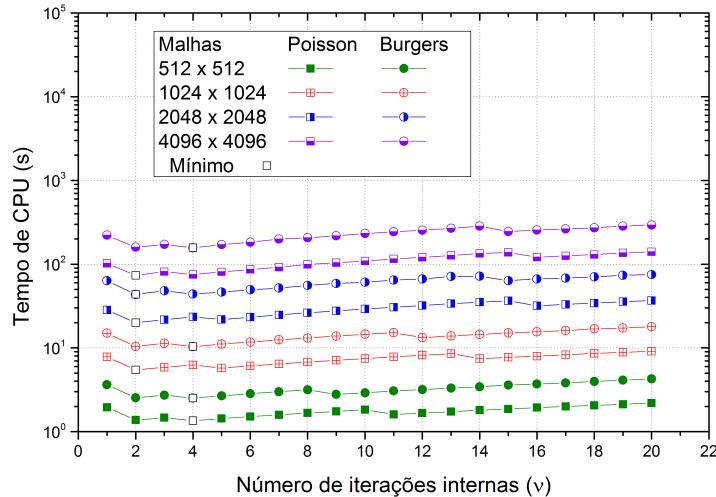


TABELA 1 – NÚMERO ÓTIMO DE ITERAÇÕES INTERNAS DO *SOLVER* PARA CADA PROBLEMA

Problema	Gerador de malha	$\nu_{óptimo}$
Poisson		3 ou 5
Burgers	Interpolação de Lagrange	2 ou 4
Poisson		2 ou 4
Burgers	Equações elípticas	2 ou 4

A análise das FIGURAS 43 e 44 não é suficiente para determinar um valor geral para uso posterior. Por este motivo, são calculadas as porcentagens nos ganhos no tempo de CPU, conforme TABELAS 2 e 3.

TABELA 2 – DIFERENÇAS ENTRE OS TEMPOS DE CPU PARA O NÚMERO DE ITERAÇÕES INTERNAS DO *SOLVER* PARA PROBLEMAS COM MALHAS GERADAS USANDO INTERPOLAÇÃO DE LAGRANGE

Problema	ν	512 x 512	1024 x 1024	2048 x 2048	4096 x 4096
Poisson	3	2,8%	0%	0%	0%
	5	0%	13%	15%	18%
Burgers	2	1%	1%	5,4 %	0%
	4	0%	0%	0%	2,1%

Da análise da TABELA 2, conclui-se que o número de iterações internas do *solver* que obteve o melhor desempenho médio é $\nu = 3$, para o problema de Poisson, e $\nu = 4$ para o problema de Burgers. O melhor desempenho médio leva em conta os problemas com mais incógnitas, que são de interesse prático. Por exemplo, a média ponderada usada para obter o desempenho médio para a equação de Poisson com $\nu = 3$ foi a seguinte

$$\frac{2,8 \times 512 + 0 \times 1024 + 0 \times 2048 + 0 \times 4096}{512 + 1024 + 2048 + 4096} = 0,182391858.$$

Essa média ponderada foi usada na TABELA 2, para ambas equações e para ambos valores de ν de cada equação. Essa média será usada sempre que for necessário. Desta forma, os valores a serem empregados nas próximas simulações são $\nu = 3$ e $\nu = 4$ para as equações de Poisson e de Burgers, respectivamente, com malhas geradas usando interpolação de Lagrange.

TABELA 3 – DIFERENÇAS ENTRE OS TEMPOS DE CPU PARA O NÚMERO DE ITERAÇÕES INTERNAS DO *SOLVER* PARA PROBLEMAS COM MALHAS GERADAS USANDO EQUAÇÕES ELÍPTICAS

Problema	ν	512×512	1024×1024	2048×2048	4096×4096
Poisson	2	1,7 %	0%	0%	0%
	4	0%	14%	17%	23%
Burgers	2	0,3%	0,3%	0%	1,2%
	4	0%	0%	1%	0%

Da análise da TABELA 3, conclui-se que o número de iterações internas do *solver* que obteve o melhor desempenho médio é $\nu = 2$, para o problema de Poisson, e $\nu = 4$ para o problema de Burgers. Desta forma, os valores a serem empregados nas próximas simulações são $\nu = 2$ e $\nu = 4$ para as equações de Poisson e de Burgers, respectivamente, com malhas geradas usando equações elípticas.

Para a resolução de problemas empregando malhas curvilíneas não ortogonais Joshi e Vanka (1991) usaram $\nu = 1$, para escoamento turbulento, e $\nu = 2$ para escoamento laminar. Oosterlee et al. (1993) e Smith, Cope e Vanka (1993) usaram $\nu = 1$ e Li, Yu e Wang (2015) usaram $\nu = 4$ para calcular as componentes do vetor velocidade. Porém, conforme já escrito, não foram encontrados trabalhos que apresentam um estudo sistemático desse parâmetro para problemas que usam sistemas de coordenadas curvilíneas não ortogonais. Para a resolução de problemas usando sistemas de coordenadas cartesianas, os trabalhos citados a seguir otimizaram alguns parâmetros do método *multigrid*. Para a equação de Laplace bidimensional, em um domínio quadrado, Pinto e Marchi (2006), com o MDF, encontraram $\nu_{óptimo} = 1$ ou 2 com o CS e $\nu_{óptimo} = 4$ com o FAS. Para a equação de Laplace discretizada em malhas quadrangulares, Suero et al. (2012) obtiveram $\nu_{óptimo} = 2$ e discretizada em malhas triangulares obtiveram $\nu_{óptimo} = 1$. Para a equação de Poisson bidimensional com anisotropia geométrica, em um domínio quadrado, Oliveira et al. (2012), com o MDF e o CS, obtiveram $\nu_{óptimo} = 1$, para o algoritmo ótimo obtido pelos autores. Para a equação de Advecção-Difusão bidimensional, Rabi e Lemos (2001) obtiveram, com o MVF e o CS, $\nu_{óptimo} = 1$ com ciclo V e $\nu_{óptimo} = 2$ com ciclo W. Para as equações de Burgers bidimensional, com domínio quadrado, Gonçalves (2013) encontrou $\nu_{óptimo} = 2$ ou 3 , empregando o MVF e o FAS-FMG. Santiago, Marchi e Souza (2015), obtiveram $\nu_{óptimo} = 5$ para o problema de Burgers com o MDF e o FAS.

Considerando-se os trabalhos citados, observa-se a sensibilidade deste parâmetro

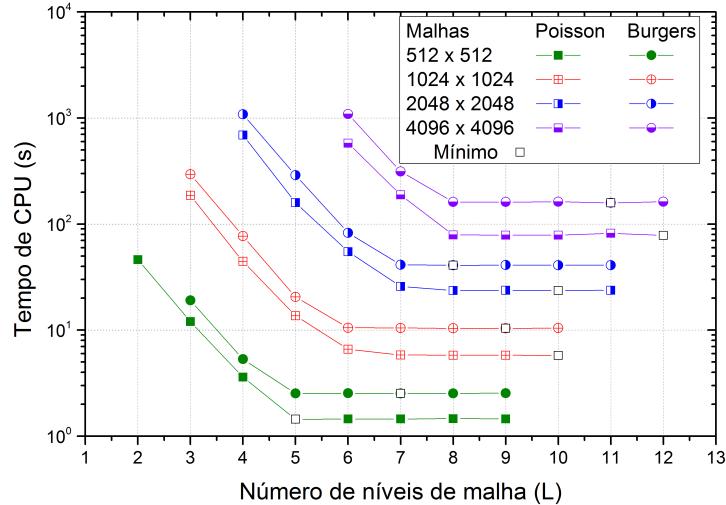
com relação aos tipos de malhas, ao método de discretização, ao método *multigrid* com CS ou FAS, com ou sem FMG e aos tipos de ciclos. Dessa forma, observa-se a importância da definição deste parâmetro. Como pode ser observado nas FIGURAS 43 e 44, o tempo de CPU aumenta para valores de ν menores ou maiores que $\nu_{\text{ótimo}}$, que pode ser significativo de acordo com o valor empregado. Por exemplo, para o problema de Burgers, com $N = 4096 \times 4096$, o emprego de $\nu = 1$ ao invés do ν sugerido, $\nu = 4$, implica em um aumento de aproximadamente 36% no tempo de CPU, para malhas geradas usando interpolação de Lagrange, e 40% para malhas geradas usando equações elípticas.

6.3 NÚMERO DE NÍVEIS DE MALHA

Outro parâmetro importante do método *multigrid* é o número de níveis de malha (L). Para o estudo da influência do número de níveis de malha sobre o tempo de CPU, considera-se o número de iterações internas do *solver* sugerido na seção 6.2, visto que se pretende encontrar parâmetros que reduzam o tempo de CPU.

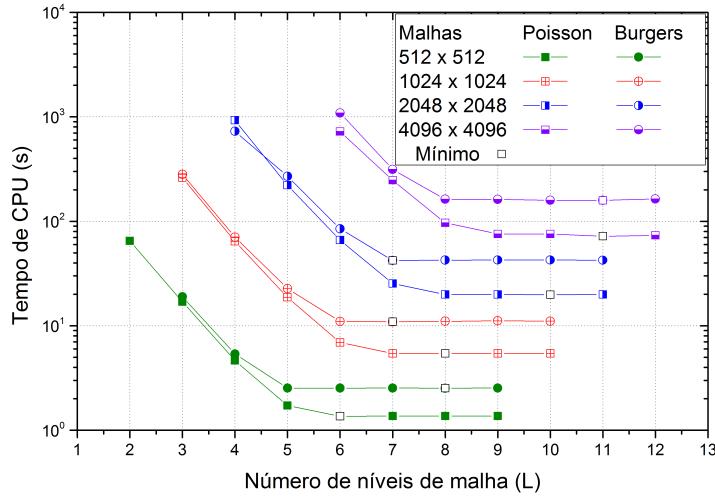
As FIGURAS 45 e 46 ilustram o efeito do número de níveis de malhas sobre o tempo de CPU, para as equações de Poisson e de Burgers para malhas geradas usando interpolação de Lagrange e equações elípticas, respectivamente, e o *solver* MSI. O símbolo \square indica o valor L que resultou no menor tempo de CPU ($L_{\text{ótimo}}$) em cada curva.

FIGURA 45 – EFEITO DO NÚMERO DE NÍVEIS DE MALHA SOBRE O TEMPO DE CPU PARA MALHAS GERADAS USANDO INTERPOLAÇÃO DE LAGRANGE



Como pode ser observado na FIGURA 45, no problema de Poisson o valor ótimo para o número de níveis de malha é $L_{\text{ótimo}} = L_{\text{máx}} - (0, 1 \text{ ou } 4)$ e para o problema de Burgers é $L_{\text{ótimo}} = L_{\text{máx}} - (1, 2 \text{ ou } 3)$. Na FIGURA 46, pode-se observar que $L_{\text{ótimo}} = L_{\text{máx}} - (1, 2 \text{ ou } 3)$, para o problema de Poisson, e $L_{\text{ótimo}} = L_{\text{máx}} - (1, 3 \text{ ou } 4)$ para o problema de Burgers. Além disso, pode-se observar que independente do tipo e do tamanho do problema, o tempo de CPU pode aumentar significativamente dependendo do número

FIGURA 46 – EFEITO DO NÚMERO DE NÍVEIS DE MALHAS SOBRE O TEMPO DE CPU PARA MALHAS GERADAS USANDO EQUAÇÕES ELÍPTICAS



de níveis de malha empregado, conforme mostram as FIGURAS 45 e 46. Entretanto, para $L \approx L_{\max}$, o tempo de CPU não se altera significativamente.

A análise das FIGURAS 45 e 46 não é suficiente para determinar um valor geral para uso posterior. Por este motivo, são calculadas as porcentagens nos ganhos no tempo de CPU, conforme apresentam as TABELAS 4 e 5.

TABELA 4 – DIFERENÇAS ENTRE OS TEMPOS DE CPU PARA L PARA PROBLEMAS COM MALHAS GERADAS USANDO INTERPOLAÇÃO DE LAGRANGE

Problema	L	512×512	1024×1024	2048×2048	4096×4096
Poisson	L_{\max}	0,77 %	0 %	0,86 %	0 %
	$L_{\max} - 1$	1,2 %	0,67 %	0 %	4,2 %
	$L_{\max} - 4$	0 %	14 %	9,4 %	0,83 %
Burgers	$L_{\max} - 1$	0,3 %	0 %	0,14 %	0 %
	$L_{\max} - 2$	0 %	0,0003 %	0,49 %	2,2 %
	$L_{\max} - 3$	0,46 %	0,6 %	0 %	1,7 %

Da análise da TABELA 4, conclui-se que o número de níveis que obteve o melhor desempenho médio é $L = L_{\max}$, para o problema de Poisson, e $L = L_{\max} - 1$ para o problema de Burgers. Isso porque as porcentagens com relação aos valores ótimos são menores para os problemas com tamanhos maiores. Desta forma, os valores a serem empregados nas próximas simulações são L_{\max} e $L_{\max} - 1$ para as equações de Poisson e de Burgers com malhas geradas usando interpolação de Lagrange.

Da análise da TABELA 5, conclui-se que o número de níveis que obteve o melhor desempenho médio é $L = L_{\max} - 1$ para o problema de Poisson e de Burgers. Desta forma, o valor a ser empregado nas próximas simulações é $L_{\max} - 1$ para essas equações com malhas geradas usando equações elípticas.

TABELA 5 – DIFERENÇAS ENTRE OS TEMPOS DE CPU PARA L PARA PROBLEMAS COM MALHAS GERADAS USANDO EQUAÇÕES ELÍPTICAS

Problema	L	512×512	1024×1024	2048×2048	4096×4096
Poisson	$L_{máx} - 1$	0,28 %	0,14 %	0 %	0 %
	$L_{máx} - 2$	0,14 %	0 %	0,15 %	4,9 %
	$L_{máx} - 3$	0 %	0,0002 %	0,15 %	5,14 %
Burgers	$L_{máx} - 1$	0 %	2,2 %	1,10 %	0 %
	$L_{máx} - 3$	0,15 %	0 %	0,77 %	2%
	$L_{máx} - 4$	0,15 %	1%	0 %	2,54 %

Da mesma forma que para o número de iterações internas do *solver*, não foram encontrados estudos sistemáticos a respeito do efeito do número de malhas no tempo de CPU para problemas que usam sistemas de coordenadas curvilíneas não ortogonais. Rayner (1991) usou $L_{máx} - 2$ para resolver um problema de escoamento de fluidos e transferência de calor no compressor de motores usando malhas geradas com equações de Laplace transformadas. Joshi e Vanka (1991) usaram $L_{máx} - 1$, para escoamento laminar, e $L_{máx}$ para escoamento turbulento. Oosterlee et al. (1993) e Smith, Cope e Vanka (1993) usaram $L_{máx}$. Li, Yu e Wang (2015) usaram $L_{máx} - 3$, para convecção mista, e $L_{máx} - 4$ para o problema com tampa móvel e para convecção natural. Para a equação de Laplace bidimensional, usando malhas cartesianas uniformes e o MDF, Tannehill, Anderson e Pletcher (1997), verificaram que não há redução significativa no tempo de CPU com o emprego do número máximo de malhas. Sugerem que sejam empregadas apenas 4 ou 5 para o problema com 129×129 incógnitas. Para a equação de Laplace bidimensional com anisotropia geométrica, Oliveira et al. (2012), com o MDF, obtiveram $L_{ótimo} = L_{máx}$. Kumar, Kumar e Das (2009) afirmaram que não há ganho com mais do que 4 malhas para o problema da cavidade com tampa móvel e MVF com 513×513 incógnitas. Para as equações de Burgers bidimensional, usando sistema de coordenadas cartesianas e o MVF, Gonçalves (2013) obteve $L_{ótimo} = L_{máx} - 1$ com FAS-FMG. Verificou que na malha 512×512 , com o emprego de $L_{ótimo} = L_{máx} = 8$, o método converge aproximadamente 190 vezes mais rápido do que com apenas 4 malhas. Ainda, para o problema de Burgers, usando sistema de coordenadas cartesianas mas, com o MDF, Santiago, Marchi e Souza (2015) obtiveram $L_{ótimo} = L_{máx} - (0 \text{ a } 3)$. Consideraram $L = L_{máx}$, porque para $L_{ótimo} < L \leq L_{máx}$ o tempo de CPU foi praticamente o mesmo.

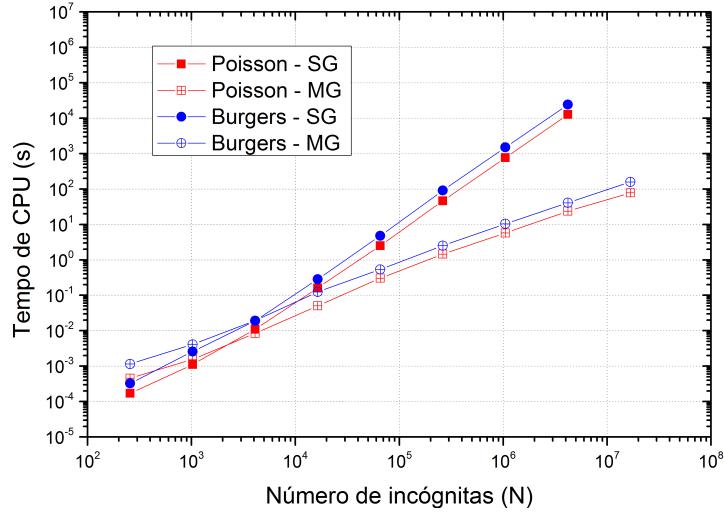
6.4 ESFORÇO COMPUTACIONAL

Com o intuito de avaliar o desempenho do algoritmo FAS para malhas curvilíneas não ortogonais perante o SG, inicialmente é analisada a influência do número de incógnitas no sistema de equações sobre o tempo de CPU. Para esta análise são considerados o número de iterações internas recomendado na seção 6.2 e o número de níveis de malhas recomendado na seção 6.3.

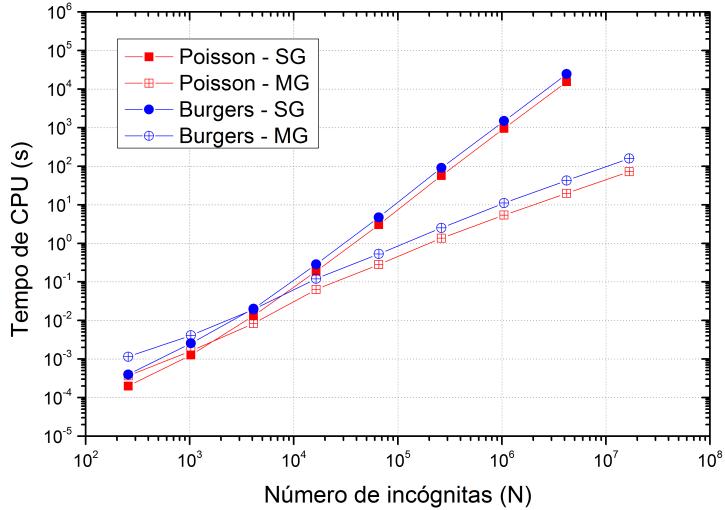
A FIGURA 47 apresenta os tempos de CPU empregados na resolução dos problemas de tamanhos $N = 16 \times 16$ até o maior tamanho de problema suportado pela memória física da máquina usada para as simulações, ou seja, $N = 4096 \times 4096$, usando o algoritmo *multigrid* com esquema FAS. Para comparação, mostram-se também os resultados obtidos com o método *singlegrid* para problemas de tamanhos $N = 16 \times 16$ até $N = 2048 \times 2048$.

FIGURA 47 – EFEITO DO NÚMERO DE INCÓGNITAS SOBRE O TEMPO DE CPU

(a) Malhas geradas usando interpolação de Lagrange



(b) Malhas geradas usando equações elípticas



Pode-se observar na FIGURA 47 que as inclinações das curvas MG para cada problema são menores com o aumento do número de incógnitas do que as respectivas curvas do SG. De acordo com Ferziger e Peric (2002), quanto maior o número de incógnitas N , maior é a vantagem do método *multigrid* em relação ao método *singlegrid*.

Na malha $N = 2048 \times 2048$, mesmo fazendo a restrição das métricas da transformação de coordenadas, o método *multigrid* é cerca de 534 e 778 vezes mais rápido que o *singlegrid* para o problema de Poisson com malhas geradas usando interpolação de

Lagrange e equações elípticas, respectivamente. Para o problema de Burgers, o *multigrid* é cerca de 585 vezes mais rápido que o *singlegrid* para malhas geradas usando interpolação de Lagrange e cerca de 573 vezes mais rápido que o *singlegrid* para o problema com malhas geradas usando equações elípticas. Como pode-se observar o *speedup* para o problema de Poisson é maior para malhas geradas usando equações elípticas se comparado com o problema com malhas geradas usando interpolação de Lagrange, o que não acontece para o problema de Burgers. A TABELA 6 resume os tempos de CPU obtidos nas malhas 512×512 , 1024×1024 e 2048×2048 volumes para os dois problemas com malhas geradas usando interpolação de Lagrange e equações elípticas.

TABELA 6 – TEMPO DE CPU (SEGUNDOS) DOS MÉTODOS MG E SG PARA OS PROBLEMAS DE POISSON E DE BURGERS COM AMBOS OS GERADORES DE MALHAS

Geradores	Problema	Malha	MG	SG
Interpolação de Lagrange	Poisson	512×512	1,457252E+00	4,625482E+01
		1024×1024	5,764274E+00	7,693196E+02
		2048×2048	2,377564E+01	1,271781E+04
	Burgers	512×512	2,534566E+00	9,144731E+01
		1024×1024	1,040380E+01	1,500109E+03
		2048×2048	4,100611E+01	2,401032E+04
Equações elípticas	Poisson	512×512	1,366867E+00	5,689292E+01
		1024×1024	5,428434E+00	9,574659E+02
		2048×2048	1,983911E+01	1,544010E+04
	Burgers	512×512	2,526756E+00	9,102551E+01
		1024×1024	1,115365E+01	1,491337E+03
		2048×2048	4,266137E+01	2,448450E+04

Smith, Cope e Vanka (1993) compararam os tempos de CPU dos métodos *multigrid* e *singlegrid*. Verificaram que, apesar do custo computacional exigido na restrição das métricas de transformação de coordenadas, os tempos de CPU do *multigrid* foram menores que os tempos de CPU do *singlegrid*.

Para determinar a ordem de complexidade (custo computacional) dos algoritmos relativos a cada método e o comportamento da curva tempo de CPU $\times N$ é feito o ajuste geométrico de cada curva (BURDEN; FAIRES, 2016), sendo a função considerada dada por

$$\text{tempo de } CPU(N) = cN^p,$$

em que c é o coeficiente que depende do método e do *solver*, p representa a ordem do algoritmo ou a inclinação de cada curva da FIGURA 47 e N o número de variáveis do problema. No caso ideal, o método *multigrid* apresenta $p = 1$, o que significa que o tempo de CPU cresce linearmente com o aumento do número de incógnitas (N) (TROTTERBERG; OOSTERLEE; SCHULLER, 2001). Assim, quanto mais próximo p for da unidade, mais eficiente é o algoritmo e quanto menor o valor de c , mais rápido ele é.

A TABELA 7 mostra os valores de c e p para os problemas de Poisson e de Burgers, para malhas geradas usando interpolação de Lagrange e equações elípticas. Para o cálculo de c e p para o *multigrid* são consideradas as malhas de $N = 64 \times 64$ até $N = 4096 \times 4096$ e para o SG são consideradas as malhas de $N = 64 \times 64$ até $N = 2048 \times 2048$.

TABELA 7 – VALORES DE p E c PARA OS PROBLEMAS PARA MALHAS GERADAS USANDO INTERPOLAÇÃO DE LAGRANGE E EQUAÇÕES ELÍPTICAS

Geradores	Problema	MG		SG	
		c	p	c	p
Interpolação de Lagrange	Poisson	1,2281E-06	1,09893	5,05591E-10	2,02201
	Burgers	3,3415E-06	1,07238	7,70927E-10	2,03899
Equações elípticas	Poisson	1,6595E-06	1,07209	5,74795E-10	2,02734
	Burgers	3,1550E-06	1,07776	8,32828E-10	2,03347

Os resultados confirmam que os tempos de CPU do método *multigrid* com o *solver* MSI para as equações de Poisson e de Burgers, para malhas geradas usando interpolação de Lagrange e equações elípticas, crescem quase que linearmente com o aumento de N . O valor de p é muito pouco afetado pelo número de equações e sua complexidade. Os valores próximos de dois para o método *singlegrid* com *solver* MSI também concordam com a literatura (BURDEN; FAIRES, 2016).

6.5 COMPARATIVO ENTRE OS *SOLVERS* MSI E MILU

Tendo por motivação os fatores de convergência médio empírico próximos aos obtidos com o *solver* MSI e os bons resultados obtidos por Rutz (2018) com o *solver* MILU, optou-se por investigar o efeito desse *solver* no algoritmo *multigrid* com FAS. Nesta seção, concentra-se na análise dos parâmetros do *multigrid* com o *solver* MILU aplicado somente ao problema de Burgers, para malhas geradas usando interpolação de Lagrange e equações elípticas. Rutz (2018), resolveu a equação de difusão bidimensional anisotrópica usando o MG e verificou que os fatores de convergência assintótica são melhores usando o MILU com $\sigma = 0,2$ do que usando o ILU.

Inicialmente, com procedimentos semelhantes aos adotados nas seções 6.2 e 6.3, são determinados os parâmetros $\nu_{\text{óptimo}}$ e $L_{\text{óptimo}}$ para o método *multigrid* com o *solver* MILU. Usa-se o *solver* MILU com $\sigma = -0,2$, porque conforme foi visto na seção 6.1, para esse valor de σ tem-se o melhor fator de convergência médio empírico.

As FIGURAS 48 e 49 ilustram a influência do número de iterações internas do *solver* MILU para o problema com malhas geradas usando interpolação de Lagrange e equações elípticas, respectivamente.

FIGURA 48 – EFEITO DO NÚMERO DE ITERAÇÕES INTERNAS SOBRE O TEMPO DE CPU PARA O PROBLEMA DE BURGERS UTILIZANDO O *SOLVER* MILU PARA MALHAS GERADAS USANDO INTERPOLAÇÃO DE LAGRANGE

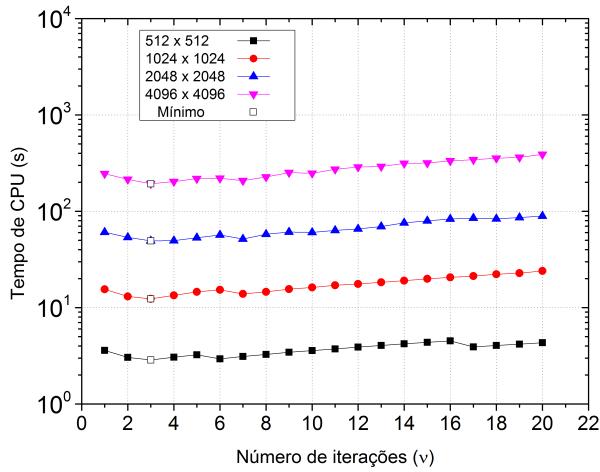
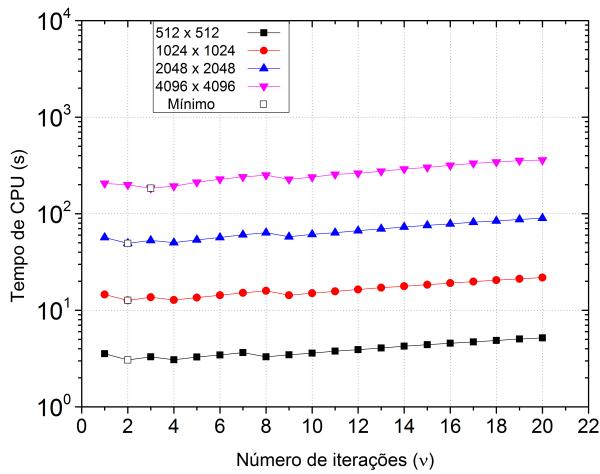


FIGURA 49 – EFEITO DO NÚMERO DE ITERAÇÕES INTERNAS SOBRE O TEMPO DE CPU PARA O PROBLEMA DE BURGERS UTILIZANDO O *SOLVER* MILU PARA MALHAS GERADAS USANDO EQUAÇÕES ELÍPTICAS



Conforme pode ser observado na FIGURA 48, $\nu = 3$ é que apresenta o menor tempo de CPU para o *solver* para o problema com malhas geradas usando interpolação de Lagrange. A análise da FIGURA 49 não é suficiente para determinar um valor geral para ν para uso posterior. Desta forma, são calculadas as porcentagens nos ganhos no tempo de CPU, conforme TABELA 8.

TABELA 8 – DIFERENÇAS ENTRE OS TEMPOS DE CPU PARA O NÚMERO DE ITERAÇÕES INTERNAS PARA O PROBLEMA DE BURGERS UTILIZANDO O *SOLVER* MILU E MALHAS GERADAS USANDO EQUAÇÕES ELÍPTICAS

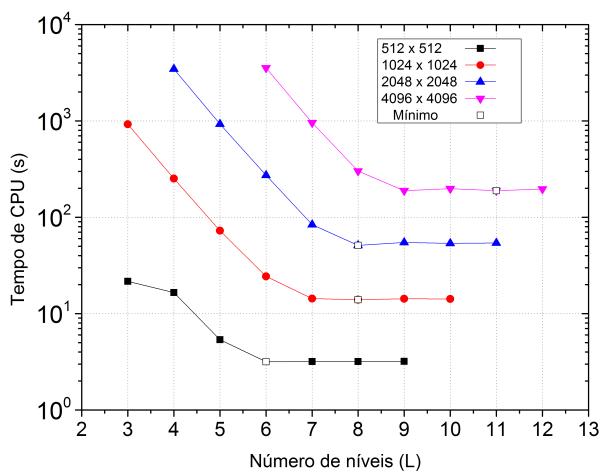
Problema	ν	512 × 512	1024 × 1024	2048 × 2048	4096 × 4096
Burgers	2	0%	0%	0%	8,4%
	3	7,6%	7,3%	7,5%	0%

Da análise da TABELA 8, conclui-se que o número de iterações do *solver* que

obteve o melhor desempenho médio é $\nu = 3$ (de acordo com a média ponderada adotada neste texto). Desta forma, o valor a ser empregado nas próximas simulações é $\nu = 3$, para o problema com malhas geradas usando equações elípticas. Novamente, observa-se a sensibilidade desse parâmetro com relação aos parâmetros algorítmicos.

As FIGURAS 50 e 51 ilustram o efeito do número de níveis de malhas sobre o tempo de CPU, utilizando o *solver* MILU e malhas geradas usando interpolação de Lagrange e equações elípticas, respectivamente.

FIGURA 50 – EFEITO DO NÚMERO DE NÍVEIS DE MALHA SOBRE O TEMPO DE CPU PARA O PROBLEMA DE BURGERS UTILIZANDO O *SOLVER* MILU E MALHAS GERADAS USANDO INTERPOLAÇÃO DE LAGRANGE



A análise das FIGURAS 50 e 51 não é suficiente para determinar um valor geral para uso posterior. As TABELAS 9 e 10 apresentam as porcentagens de ganhos no tempo de CPU com relação ao valor ótimo, para o problema utilizando o *solver* MILU e malhas geradas usando interpolação de Lagrange e equações elípticas, respectivamente.

FIGURA 51 – EFEITO DO NÚMERO DE NÍVEIS DE MALHA SOBRE O TEMPO DE CPU PARA O PROBLEMA DE BURGERS UTILIZANDO O *SOLVER* MILU E MALHAS GERADAS USANDO EQUAÇÕES ELÍPTICAS

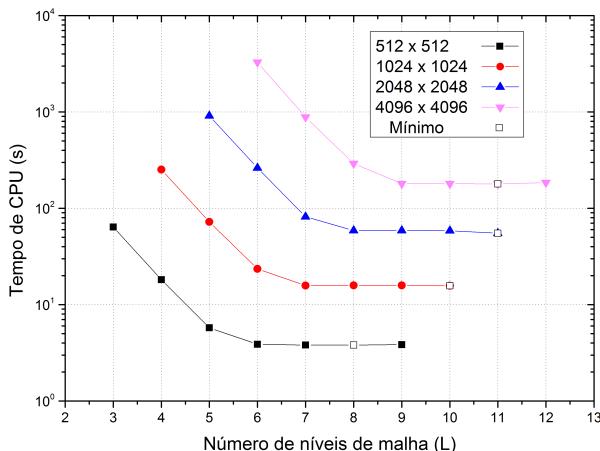


TABELA 9 – DIFERENÇAS ENTRE OS TEMPOS DE CPU PARA L PARA O PROBLEMA DE BURGERS, UTILIZANDO O *SOLVER* MILU E MALHAS GERADAS USANDO INTERPOLAÇÃO DE LAGRANGE

Problema	L	512×512	1024×1024	2048×2048	4096×4096
Burgers	$L_{\max} - 1$	0,2%	2%	4,7 %	0%
	$L_{\max} - 2$	0,3%	0%	6,9 %	5%
	$L_{\max} - 3$	0%	2,4 %	0%	0,1%

Da análise da TABELA 9, conclui-se que o número de níveis que obteve o melhor desempenho médio é $L = L_{\max} - 3$. Desta forma, o valor a ser empregado nas próximas simulações é $L_{\max} - 3$, para as equações de Burgers com malhas geradas usando interpolação de Lagrange.

TABELA 10 – DIFERENÇAS ENTRE OS TEMPOS DE CPU PARA L PARA O PROBLEMA DE BURGERS, UTILIZANDO O *SOLVER* MILU E MALHAS GERADAS USANDO EQUAÇÕES ELÍPTICAS

Problema	L	512×512	1024×1024	2048×2048	4096×4096
Burgers	L_{\max}	0,9%	0%	0%	2,9%
	$L_{\max} - 1$	0%	0,7%	6,5%	0%

Da análise da TABELA 10, conclui-se que o número de níveis que obteve o melhor desempenho médio é $L = L_{\max}$. Desta forma, o valor a ser empregado nas próximas simulações é L_{\max} , para as equações de Burgers com malhas geradas usando equações elípticas.

A seguir é avaliado o efeito do MILU sobre o tempo de CPU. As FIGURAS 52 e 53 mostram um comparativo do tempo de CPU empregado na solução do problema com os *solvers* MSI e MILU, para os métodos SG e MG e para malhas geradas usando interpolação de Lagrange e equações elípticas, respectivamente.

FIGURA 52 – COMPARATIVO DO TEMPO DE CPU (SEGUNDOS) VERSUS N PARA O PROBLEMA DE BURGERS COM OS *SOLVERS* MSI E MILU PARA MALHAS GERADAS USANDO INTERPOLAÇÃO DE LAGRANGE

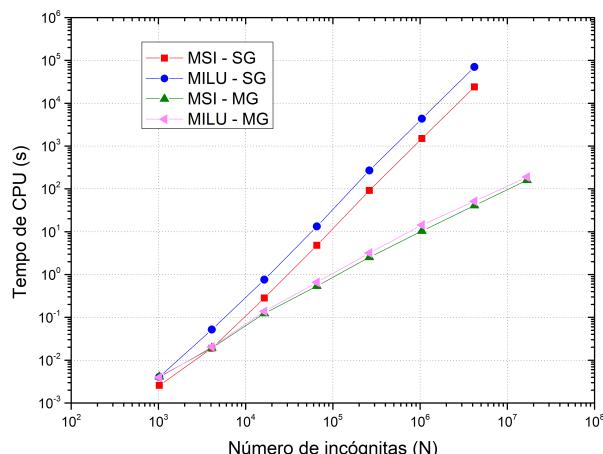
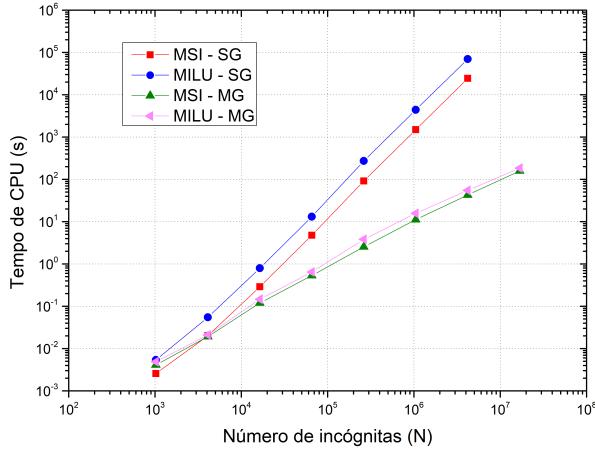


FIGURA 53 – COMPARATIVO DO TEMPO DE CPU (SEGUNDOS) VERSUS N PARA O PROBLEMA DE BURGERS COM OS *SOLVERS* MSI E MILU PARA MALHAS GERADAS USANDO EQUAÇÕES ELÍPTICAS



Observando as FIGURAS 52 e 53, os tempos de CPU empregados para resolver o problema usando o *multigrid*, para ambos os geradores de malhas, são próximos, não permitindo uma análise detalhada dos tempos de CPU. As TABELAS 11 e 12 apresentam os tempos de CPU para o problema com os *solvers* MSI e MILU para ambos os geradores de malha.

Como pode ser observado na TABELA 11, considerando o método *multigrid* com o FAS, os tempos de CPU para o problema com o *solver* MSI e malhas geradas usando interpolação de Lagrange, são menores do que os obtidos com o MILU, sendo em torno de 19% mais rápido na malha mais refinada. Na TABELA 12, observa-se que considerando o método *multigrid* com o FAS, os tempos de CPU para o problema com o *solver* MSI e malhas geradas usando equações elípticas também são menores do que os obtidos com o MILU, sendo 16% mais rápido na malha mais refinada. Comparando-se os tempos de CPU do SG, nas TABELAS 11 e 12, o MSI também foi mais rápido em todas as malhas. O comportamento do MSI em relação ao MILU, obtido neste trabalho, foi semelhante ao obtido por Norris (2000). No trabalho citado, o *multigrid* com o MSI convergiu mais rápido que o *multigrid* com o ILU (MILU com $\sigma = 0$), para a equação de Laplace bidimensional com condições de contorno do tipo Dirichlet, usando sistema de coordenadas cartesianas, na malha $N = 259 \times 259$. Pinto (2006), investigando o efeito de alguns parâmetros do *multigrid* com o FAS para a equação de Laplace bidimensional, usando sistema de coordenadas cartesianas, verificou que o *solver* MSI também foi mais rápido que o GS-Lex e o ADI.

Outra análise está relacionada ao estudo do erro (diferença entre a solução analítica e numérica). Calculou-se a norma infinito do erro para o problema de Burgers resolvido com o MSI e malhas geradas usando interpolação de Lagrange e equações elípticas, uma vez que os tempos de CPU para esse *solver* foram menores do que os obtidos com o MILU,

para ambos os geradores de malha. A TABELA 13 apresenta os resultados.

TABELA 11 – TEMPO DE CPU (SEGUNDOS) PARA O PROBLEMA DE BURGERS RESOLVIDO COM OS *SOLVERS* MSI e MILU E MALHAS GERADAS USANDO INTERPOLAÇÃO DE LAGRANGE

Malha	SG		MG	
	MSI	MILU	MSI	MILU
32×32	0,003	0,004	0,004	0,004
64×64	0,019	0,052	0,019	0,020
128×128	0,284	0,758	0,124	0,138
256×256	4,806	13,278	0,539	0,651
512×512	91,447	268,773	2,535	3,171
1024×1024	1500,109	4370,962	10,404	14,325
2048×2048	24010,317	70533,482	41,006	51,175
4096×4096	—	—	158,838	189,018

TABELA 12 – TEMPO DE CPU (SEGUNDOS) PARA O PROBLEMA DE BURGERS RESOLVIDO COM OS *SOLVERS* MSI e MILU E MALHAS GERADAS USANDO EQUAÇÕES ELÍPTICAS

Malha	SG		MG	
	MSI	MILU	MSI	MILU
32×32	0,003	0,005	0,004	0,005
64×64	0,020	0,055	0,019	0,021
128×128	0,288	0,793	0,120	0,146
256×256	4,744	13,059	0,533	0,645
512×512	91,026	272,577	2,527	3,843
1024×1024	1491,337	4378,430	11,154	15,731
2048×2048	24484,499	69576,038	42,661	55,018
4096×4096	—	—	158,729	184,816

TABELA 13 – NORMA INFINITO DO ERRO PARA O PROBLEMA DE BURGERS RESOLVIDO COM O MSI E AMBOS OS GERADORES DE MALHAS

Malha	MSI	
	Interpolação de Lagrange	Equações elípticas
16×16	4,9562166219E-05	2,7453241013E-05
32×32	2,4758930310E-05	2,2246971829E-05
64×64	1,5475177469E-05	1,0139419193E-05
128×128	9,9001163248E-06	5,9667487559E-06
256×256	5,4838644067E-06	3,1500430985E-06
512×512	2,8784941319E-06	1,6042873321E-06
1024×1024	1,4741079725E-06	8,0633172263E-07
2048×2048	7,4539890321E-07	4,0348652942E-07
4096×4096	3,7419632012E-07	2,0167536730E-07

Considerando-se os resultados da TABELA 13, as soluções obtidas para o problema com malhas geradas usando equações elípticas são levemente mais acuradas do que as

soluções obtidas para o problema com malhas geradas usando interpolação de Lagrange. Isso porque os geradores de malhas que empregam equações diferenciais elípticas geram malhas mais suaves e melhoram a não ortogonalidade da malha (FLETCHER, 1991; OOSTERLEE et al., 1993; HOFFMANN; CHIANG, 2000; MALISKA, 2010). Oosterlee et al. (1993) usaram malhas geradas usando métodos algébrico e diferenciais no domínio no formato de L. Verificaram que para obter soluções mais acuradas, para números de Reynolds iguais a 100 e 1000, foram necessárias usar malhas geradas por equações biharmônicas para discretizar o domínio.

7 CONCLUSÃO

Este trabalho teve como principal objetivo a investigação de parâmetros do método *multigrid* geométrico para as equações bidimensionais de Poisson (problema linear) e de Burgers (problema não linear com duas equações) usando malhas curvilíneas estruturadas não ortogonais. Para alcançar este objetivo foi desenvolvido um código computacional para a resolução de cada problema, sendo que cada um deles foi resolvido com o algoritmo SG e MG com FAS para malhas geradas usando interpolação de Lagrange e equações elípticas.

Primeiramente, foi realizado um estudo do fator de convergência médio empírico (ρ_m) para todos os *solvers* empregados (GS-Lex, η -linha-GS, MSI e MILU). Além disso, foi realizado um estudo em que analisou-se a influência de vários parâmetros do método *multigrid* sobre o tempo de CPU empregado para resolver cada problema. Os parâmetros analisados foram: o número de iterações internas (ν), o número de níveis de malha (L) e o número de incógnitas (N). Foi analizado o desempenho e a eficiência do *multigrid* com FAS para o problema de Poisson e de Burgers para malhas geradas tanto usando interpolação de Lagrange quanto equações elípticas. Posteriormente, definidos os parâmetros ótimos para as equações de Burgers para os dois geradores de malhas, foram realizados estudos procurando identificar a melhor escolha do *solver*, entre o MSI e o MILU, objetivando melhorar o desempenho do método. Como resultados deste estudo, verificou-se que:

- Os valores de σ que resultaram em menores valores para ρ_m para os problemas abordados foram 0,9 e -0,2 para os *solvers* MSI e MILU, respectivamente;
- Entre os *solvers* avaliados (GS-Lex, η -linha-GS, MSI e MILU), o MSI ($\sigma = 0,9$) e MILU ($\sigma = -0,2$) apresentaram os melhores fatores de convergência médio empírico;
- Os números de iterações internas para o MSI sugeridos para a equação de Poisson são $\nu = 3$ e $\nu = 2$ para malhas geradas usando interpolação de Lagrange e equações elípticas, respectivamente. Para as equações de Burgers os números de iterações internas sugeridos são $\nu = 4$ e $\nu = 3$, para os *solvers* MSI e MILU, respectivamente, para ambos geradores de malhas;
- O tempo de CPU pode aumentar consideravelmente, com o emprego de valores que não o sugerido para o número de iterações internas do *solver*;
- O valor recomendado para o número de níveis de malha para a equação de Poisson, usando MSI, é $L = L_{\max}$, para malhas geradas usando interpolação de Lagrange, e $L = L_{\max} - 1$ para malhas geradas usando equações de elípticas. Para as equações de Burgers, com o *solver* MSI, o valor recomendado para o número de níveis de malha é $L = L_{\max} - 1$ para ambos geradores de malhas, e com o *solver* MILU é

$L = L_{\max} - 3$ para malhas geradas usando interpolação de Lagrange e $L = L_{\max}$ para equações de elípticas.

- Independente do problema, do valor de N , do *solver* empregado e do gerador de malha, o tempo de CPU não se altera significativamente quando $L \approx L_{\max}$, mas aumenta significativamente dependendo do número de níveis de malha empregado;
- O desempenho teórico do método *multigrid* para as equações de Poisson e de Burgers, resolvidas usando malhas geradas empregando interpolação de Lagrange e equações elípticas, foi confirmado, ou seja, $p \approx 1$, independentemente do gerador de malha empregado;
- O número de equações envolvidas em cada problema, assim como a complexidade presente (formato do domínio físico e equações não lineares), não afeta significativamente a eficiência do método *multigrid* com FAS;
- Na malha $N = 4096 \times 4096$, tem-se que o método *multigrid* com o *solver* MSI, para o problema de Burgers, é aproximadamente 19% e 16% mais rápido que o *solver* MILU, para as malhas geradas usando interpolação de Lagrange e equações elípticas, respectivamente;
- A norma infinito do erro para o problema de Burgers, resolvido com o método *multigrid* e o *solver* MSI, é sensivelmente maior para malhas geradas usando interpolação de Lagrange se comparada com os valores obtidos para malhas geradas usando equações elípticas.

7.1 CONTRIBUIÇÕES

Nesta tese foram desenvolvidos códigos computacionais com base no método *multigrid* geométrico para as equações de Poisson e de Burgers usando malhas curvilíneas estruturadas não ortogonais e realizaram-se estudos sistemáticos de parâmetros para importantes modelos matemáticos da CFD. Assim, contribui-se com a literatura existente no sentido de que:

- As equações de Poisson e de Burgers foram resolvidas usando malhas curvilíneas estruturadas não ortogonais de até 4096×4096 volumes de controle;
- Apresentou-se um estudo sistemático de alguns parâmetros (*solvers*, fator de convergência médio empírico, número de iterações internas do *solver* e número de níveis de malha) para as equações de Poisson e de Burgers usando malhas curvilíneas estruturadas não ortogonais e o MVF, que não é encontrado na literatura;

- O esforço computacional proporcional ao número de incógnitas, para o método *multigrid* com o FAS, foi confirmado para os dois problemas investigados;
- Apresentou-se um código eficiente para o problema de Burgers. A eficiência foi obtida pelo estudo dos parâmetros ótimos do MG e pela seleção dos melhores *solvers* para esse problema; além de se ter um código com fator de convergência médio empírico próximo de 0,03.

7.2 PROPOSTA DE TRABALHOS FUTUROS

Com a finalidade de complementar e expandir os estudos deste trabalho, são sugeridas as seguintes propostas de trabalhos futuros:

- Investigar o efeito de diferentes métodos de discretização para as métricas da transformação de coordenadas, a fim de obter discretizações mais acuradas;
- Mapear o domínio usando de multiblocos;
- Estudar o efeito de outros tipos de ciclos como os ciclos W e F, sobre o tempo de CPU;
- Investigar o efeito do solver ILU por blocos, BILU, sobre o tempo de CPU;
- Fazer um estudo comparativo entre os algoritmos FAS e FAS-FMG;
- Fazer um estudo comparativo de operadores de restrição e de prolongação;
- Aplicar o método *multigrid* para as equações de Navier-Stokes bidimensionais, usando malhas curvilíneas não ortogonais e investigar os parâmetros ótimos.

REFERÊNCIAS

- AKSAN, E. N.; ÖZDEŞ, A. A numerical solution of burgers equation. *Applied mathematics and computation*, Elsevier, v. 156, n. 2, p. 395–402, 2004. Citado na página 29.
- ANDERSON, D.; TANNEHILL, J. C.; PLETCHER, R. H. *Computational fluid mechanics and heat transfer*. 3rd ed. ed. Boca Ratón, USA: CRC Press, 2013. Citado 6 vezes nas páginas 21, 22, 50, 59, 61 e 85.
- ARAKI, L. K. *Verificação de soluções numéricas de escoamentos reativos em motores foguete*. Tese (Doutorado) — Curitiba: Universidade Federal do Paraná, 2007. Citado na página 127.
- BAKHVALOV, N. S. On the convergence of a relaxation method with natural constraints on the elliptic operator. *USSR Computational Mathematics and Mathematical Physics*, Elsevier, v. 6, n. 5, p. 101–135, 1966. Citado na página 23.
- BONKILE, M. P. et al. A systematic literature review of burgers equation with recent advances. *Pramana*, Springer, v. 90, n. 6, p. 69, 2018. Citado 2 vezes nas páginas 28 e 29.
- BRANDT, A. Multi-level adaptive technique (mlat) for fast numerical solution to boundary value problems. In: *Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics*. Berlin, Heidelberg: Springer, 1973. v. 18, p. 82–89. Citado na página 23.
- BRANDT, A. Multi-level adaptive techniques (mlat): I. the multi-grid method. IBM Thomas J. Watson Research Report RC6026, 1976. Citado na página 23.
- BRANDT, A. Multi-level adaptive solutions to boundary-value problems, mathematics of computation. *USSR Comput. Math. and Math. Phys.*, n. 31, p. 333–390, 1977. Citado 3 vezes nas páginas 23, 67 e 70.
- BRANDT, A. Barriers to achieving textbook multigrid efficiency (tme) in cfd. In: *NASA/CR, ICASE Interim Report*. Rehovot, Israel: Weizmann Institute for Science, 1998. v. 32. Citado na página 24.
- BRIGGS, W. L.; HENSON, V. E.; McCORMICK, S. F. *A Multigrid Tutorial*. 2nd. ed. Philadelphia: SIAM, 2000. Citado 12 vezes nas páginas 22, 23, 32, 40, 67, 69, 70, 71, 73, 75, 76 e 96.
- BURDEN, R.; FAIRES, J. *Numerical Analysis*. 10nd. ed. Boston: Cengage Learning, 2016. Citado 8 vezes nas páginas 21, 22, 40, 42, 65, 68, 109 e 110.
- DEMIRDŽIĆ, I.; LILEK, Ž.; PERIĆ, M. Fluid flow and heat transfer test problems for non-orthogonal grids: bench-mark solutions. *International Journal for Numerical Methods in Fluids*, Wiley Online Library, v. 15, n. 3, p. 329–354, 1992. Citado na página 28.
- DHAWAN, S. et al. Contemporary review of techniques for the solution of nonlinear burgers equation. *Journal of Computational Science*, Elsevier, v. 3, n. 5, p. 405–419, 2012. Citado na página 29.
- DOUGLAS, C. Multigrid methods in science and engineering. *IEEE Computational Science and Engineering*, IEEE, v. 3, n. 4, p. 55–68, 1996. Citado na página 24.

- EFE, M. Ö. Observer-based boundary control for 2d burgers equation. *Transactions of the Institute of Measurement and Control*, SAGE Publications, v. 28, n. 2, p. 177–185, 2006. Citado na página 33.
- FEDORENKO, R. P. On the speed of convergence of an iteration process. *USSR Comput. Math. and Math. Phys.*, n. 4, p. 227, 1964. Citado na página 23.
- FERM, L.; LÖTSTEDT, P. Two-grid solution of shock problems. *SIAM Journal on Scientific Computing*, Society for Industrial and Applied Mathematics, v. 18, n. 6, p. 1533–1552, 1997. Citado 2 vezes nas páginas 30 e 33.
- FERZIGER, J. H.; PERIC, M. *Computational Methods for Fluid Dynamics*. 3nd. ed. New York, USA: Springer, 2002. Citado 10 vezes nas páginas 23, 32, 38, 51, 54, 75, 97, 108, 127 e 132.
- FLETCHER, C. A. *Computational techniques for fluid dynamics 1: Fundamental and General Techniques*. 2nd. ed. Berlim, Alemanha: Springer - Verlag, 1991. v. 1. Citado 4 vezes nas páginas 78, 81, 116 e 132.
- FORTUNA, A. O. *Técnicas Computacionais para Dinâmica dos Fluidos*. São Paulo: Edusp, 2000. Citado 4 vezes nas páginas 20, 22, 40 e 82.
- FRANCO, S. R. et al. Multigrid method based on a space-time approach with standard coarsening for parabolic problems. *Applied Mathematics and Computation*, Elsevier, v. 317, p. 25–34, 2018. Citado na página 32.
- GONÇALVES, S. F. T. *Estudo de parâmetros do método multigrid geométrico para equações 2D em CFD e volumes finitos*. Tese (Doutorado) — Universidade Federal do Paraná, Curitiba, 2013. Citado 5 vezes nas páginas 31, 32, 33, 104 e 107.
- HE, P. et al. Multigrid calculation of fluid flows in complex 3d geometries using curvilinear grids. *Computers & fluids*, Elsevier, v. 25, n. 4, p. 395–419, 1996. Citado na página 28.
- HOFFMAN, K. A.; CHIANG, S. T. *Computational Fluid Dynamics Volume II*. 4nd. ed. [S.l.]: Wichita, Kan USA, 2000. Citado 2 vezes nas páginas 57 e 59.
- HOFFMANN, K. A.; CHIANG, S. T. *Computational fluid dynamics volume I*. [S.l.: s.n.], 2000. Citado 2 vezes nas páginas 57 e 116.
- HOWSE, A. J. et al. Parallel-in-time multigrid with adaptive spatial coarsening for the linear advection and inviscid burgers equations. *SIAM Journal on Scientific Computing*, SIAM, v. 41, n. 1, p. A538–A565, 2019. Citado na página 31.
- HUGHES, T. J. *The finite element method: linear static and dynamic finite element analysis*. Mineola, NY: Courier Corporation, 2000. Citado na página 21.
- IBRAHEEM, S.; DEMUREN, A. On bi-grid local mode analysis of solution techniques for 3-d euler and navier–stokes equations. *Journal of Computational Physics*, Elsevier, v. 125, n. 2, p. 354–377, 1996. Citado na página 29.
- INCROPERA, F. P. et al. *Fundamentos de transferencia de calor e de massa*. 6. ed. Rio de Janeiro: LTC - Livros Técnicos e Científicos Editora S. A., 2008. Citado na página 79.

- ISAACSON, E.; KELLER, H. B. *Analysis of numerical methods*. 2nd. ed. Mineola, NY: Courier Corporation, 1994. Citado na página 96.
- JOSHI, D. S.; VANKA, S. P. Multigrid calculation procedure for internal flows in complex geometries. *Numerical Heat Transfer, Part B Fundamentals*, Taylor & Francis, v. 20, n. 1, p. 61–80, 1991. Citado 4 vezes nas páginas 26, 74, 104 e 107.
- KANNAN, R.; WANG, Z. A high order spectral volume solution to the burgers equation using the hopf-cole transformation. *International Journal for Numerical Methods in Fluids*, Wiley Online Library, v. 69, n. 4, p. 781–801, 2012. Citado na página 29.
- KREYSZIG, E. *Matemática superior para engenharia*. 9. ed. Rio de Janeiro: Livros Técnicos e Científicos - LTC, 2009. Citado 2 vezes nas páginas 36 e 52.
- KUMAR, D. S.; KUMAR, K. S.; DAS, M. K. A fine grid solution for a lid-driven cavity flow using multigrid method. *Engineering Applications of Computational Fluid Mechanics*, Taylor & Francis, v. 3, n. 3, p. 336–354, 2009. Citado 3 vezes nas páginas 25, 74 e 107.
- LI, J.; YU, B.; WANG, M. Benchmark solutions for two-dimensional fluid flow and heat transfer problems in irregular regions using multigrid method. *Advances in Mechanical Engineering*, SAGE Publications Sage UK: London, England, v. 7, n. 11, 2015. Citado 4 vezes nas páginas 28, 75, 104 e 107.
- LILEK, Z.; MUZAFERIJA, S.; PERIĆ, M. Efficiency and accuracy aspects of a full-multigrid simple algorithm for three-dimensional flows. *Numerical Heat Transfer*, Taylor & Francis, v. 31, n. 1, p. 23–42, 1997. Citado na página 38.
- LIU, F.; WEIPING, S. Numerical solutions of two-dimensional burgers equations by lattice boltzmann method. *Communications in Nonlinear Science and Numerical Simulation*, Elsevier, v. 16, n. 1, p. 150–157, 2011. Citado na página 29.
- MALISKA, C. R. *Transferência de Calor e Mecânica dos Fluidos Computacionais*. 2nd. ed. Rio de Janeiro: LTC, 2010. Citado 18 vezes nas páginas 20, 21, 26, 35, 37, 38, 39, 51, 52, 54, 55, 56, 57, 58, 61, 78, 92 e 116.
- MARCHI, C.; MALISKA, C. A nonorthogonal finite-volume method for the solution of all speed flows using co-located variables. *Numerical Heat Transfer*, Taylor & Francis, v. 26, n. 3, p. 293–311, 1994. Citado na página 92.
- MARCHI, C. H. *Verificação de soluções numéricas unidimensionais em dinâmica dos fluidos*. Tese (Doutorado) — Universidade Federal de Santa Catarina, Florianópolis, 2001. Citado 2 vezes nas páginas 127 e 128.
- MESQUITA, M. S.; LEMOS, M. J. S. de. Optimal multigrid solutions of two-dimensional convection-conduction problems. *Applied Mathematics and Computation*, Elsevier, v. 152, n. 3, p. 725–742, 2004. Citado 2 vezes nas páginas 25 e 32.
- MULDER, W. A. Multigrid for the one-dimensional inviscid burgers equation. *SIAM Journal on Scientific and Statistical Computing*, SIAM, v. 11, n. 1, p. 33–50, 1990. Citado na página 29.
- NERNEY, S.; SCHMAHL, E. J.; MUSIELAK, Z. Analytic solutions of the vector burgers equation. *Quarterly of applied mathematics*, v. 54, n. 1, p. 63–71, 1996. Citado na página 29.

- NEVEU, E.; DEBREU, L.; DIMET, F. X. L. Multigrid methods and data assimilation - convergence study and first experiments on non-linear equations. *ARIMA*, v. 14, p. 63–80, 2011. Citado na página 30.
- NORRIS, S. *A Parallel Navier-Stokes Solver for Natural Convection and Free Surface Flow*. Tese (Doutorado) — University of Sydney, 2000. Citado na página 114.
- OLIVEIRA, F. *Efeito de malhas anisotrópicas bidimensionais sobre o desempenho do método multigrid geométrico*. Tese (Doutorado) — Curitiba: Universidade Federal do Paraná, 2010. Citado na página 98.
- OLIVEIRA, F. et al. Optimized partial semicoarsening multigrid algorithm for heat diffusion problems and anisotropic grids. *Applied Mathematical Modelling*, Elsevier, v. 36, n. 10, p. 4665–4676, 2012. Citado 3 vezes nas páginas 25, 104 e 107.
- OLIVEIRA, M. L. et al. On the robustness of the xy-zebra-gauss-seidel smoother on an anisotropic diffusion problem. *CMES-COMPUTER MODELING IN ENGINEERING & SCIENCES*, TECH SCIENCE PRESS 6825 JIMMY CARTER BLVD, STE 1850, NORCROSS, GA 30071 USA, v. 117, n. 2, p. 251–270, 2018. Citado 2 vezes nas páginas 32 e 42.
- OOSTERLEE, C. et al. Benchmark solutions for the incompressible navier–stokes equations in general co-ordinates on staggered grids. *International Journal for Numerical Methods in Fluids*, Wiley Online Library, v. 17, n. 4, p. 301–321, 1993. Citado 6 vezes nas páginas 27, 42, 78, 104, 107 e 116.
- OOSTERLEE, C. W.; WESSELING, P. A robust multigrid method for a discretization of the incompressible navier-stokes equations in general coordinates. *IMPACT of Computing in Science and Engineering*, Elsevier Science, v. 5, n. 2, p. 128–151, 1993. Citado 2 vezes nas páginas 46 e 132.
- PATANKAR, S. *Numerical heat transfer and fluid flow*. Washington: Hemisphere Publishing Corporation, 1980. Citado 3 vezes nas páginas 21, 35 e 38.
- PINTO, M. A. V. *Comportamento do Multigrid geométrico em problemas de transferencia de calor*. Tese (Doutorado) — Curitiba: Universidade Federal do Paraná, 2006. Citado na página 114.
- PINTO, M. A. V.; MARCHI, C. H. Efeito de parâmetros do método multigrid cs e fas sobre o tempo de cpu para a equação de laplace bidimensional. *Proceedings of ENCIT*, 2006. Citado 4 vezes nas páginas 25, 70, 101 e 104.
- RABI, J. A.; LEMOS, M. J. Optimization of convergence acceleration in multigrid numerical solutions of conductive–convective problems. *Applied Mathematics and Computation*, Elsevier, v. 124, n. 2, p. 215–226, 2001. Citado 2 vezes nas páginas 25 e 104.
- RAYNER, D. Multigrid flow solutions in complex two-dimensional geometries. *International journal for numerical methods in fluids*, Wiley Online Library, v. 13, n. 4, p. 507–518, 1991. Citado 4 vezes nas páginas 26, 75, 85 e 107.
- ROACHE, P. J. Perspective: a method for uniform reporting of grid refinement studies. *Transactions-American Society of Mechanical Engineers Journal of Fluids Engineering*,

- AMERICAN SOCIETY MECHANICAL ENGINEERS, v. 116, p. 405–405, 1994. Citado na página 128.
- ROACHE, P. J. Code verification by the method of manufactured solutions. *Journal of Fluids Engineering*, American Society of Mechanical Engineers, v. 124, n. 1, p. 4–10, 2002. Citado na página 79.
- RUTZ, G. V. *Efeitos da anisotropia física sobre o método multigrid aplicado à equação de difusão*. Tese (Doutorado) — Universidade Federal do Paraná, Curitiba, 2018. Citado 2 vezes nas páginas 46 e 110.
- RUTZ, G. V.; PINTO, M. A. V.; GONÇALVES, S. F. T. On the robustness of the multigrid method combining ilu and partial weight applied in an orthotropic diffusion. *REVISTA INTERNACIONAL DE METODOS NUMERICOS PARA CALCULO Y DISENO EN INGENIERIA*, SCIPEDIA SL SCIPEDIA SL, BARCELONA, 00000, SPAIN, v. 35, n. 1, 2019. Citado na página 32.
- SAAD, Y. *Iterative methods for sparse linear systems*. 2nd. ed. Philadelphia: Society for Industrial and Applied Mathematics - SIAM, 2003. v. 82. Citado 3 vezes nas páginas 40, 43 e 46.
- SANTIAGO, C.; MARCHI, C.; SOUZA, L. F. d. Performance of geometric multigrid method for coupled two-dimensional systems in cfd. *Applied Mathematical Modelling*, Elsevier, v. 39, n. 9, p. 2602–2616, 2015. Citado 6 vezes nas páginas 25, 31, 32, 33, 104 e 107.
- SANTIAGO, C. D. *Estudo de parâmetros do método Multigrid para sistemas de equações 2D em CFD*. Tese (Doutorado) — Universidade Federal do Paraná, Curitiba, 2010. Citado 5 vezes nas páginas 70, 71, 72, 73 e 77.
- SCHNEIDER, G.; ZEDAN, M. A modified strongly implicit procedure for the numerical solution of field problems. *Numerical Heat Transfer*, Taylor & Francis, v. 4, n. 1, p. 1–19, 1981. Citado 4 vezes nas páginas 22, 47, 49 e 50.
- SHIH, T.; TAN, C.; HWANG, B. Effects of grid staggering on numerical schemes. *International Journal for numerical methods in fluids*, Wiley Online Library, v. 9, n. 2, p. 193–212, 1989. Citado 2 vezes nas páginas 81 e 82.
- SMITH, K. M.; COPE, W. K.; VANKA, S. P. A multigrid procedure for three-dimensional flows on non-orthogonal collocated grids. *International journal for numerical methods in fluids*, Wiley Online Library, v. 17, n. 10, p. 887–904, 1993. Citado 8 vezes nas páginas 27, 32, 74, 75, 94, 104, 107 e 109.
- STEINBRUCH, A.; WINTERLE, P. *Geometria analítica*. 2. ed. São Paulo: [s.n.], 1987. Citado na página 54.
- STÜBEN, K. A review of algebraic multigrid. *Journal of Computational and Applied Mathematics*, Elsevier, v. 128, n. 1, p. 281–309, 2001. Citado 2 vezes nas páginas 24 e 101.
- SUERO, R. et al. Analysis of algebraic multigrid parameters for two-dimensional steady-state heat diffusion equations. *Applied Mathematical Modelling*, Elsevier, v. 36, n. 7, p. 2996–3006, 2012. Citado 2 vezes nas páginas 25 e 104.

- SZABO, B.; BABUSKA, I. *Finite Element Analysis*. New York, USA: Wiley, 1991. Citado na página 127.
- TANNEHILL, J. C.; ANDERSON, D. A.; PLETCHER, R. H. *Computational Fluid Mechanics and Heat Transfer*. Philadelphia: Taylor & Francis, 1997. Citado 4 vezes nas páginas 23, 24, 65 e 107.
- THEKALE, A. et al. Optimizing the number of multigrid cycles in the full multigrid algorithm. *Numerical Linear Algebra with Applications*, Wiley Online Library, v. 17, n. 2-3, p. 199–210, 2010. Citado na página 32.
- THOMPSON, J. F.; THAMES, F. C.; MASTIN, C. W. Automatic numerical generation of body-fitted curvilinear coordinate system for field containing any number of arbitrary two-dimensional bodies. *Journal of computational physics*, Elsevier, v. 15, n. 3, p. 299–319, 1974. Citado 2 vezes nas páginas 26 e 61.
- THOMPSON, J. F.; THAMES, F. C.; MASTIN, C. W. Boundary-fitted curvilinear coordinate systems for solution of partial differential equations on fields containing any number of arbitrary two-dimensional bodies. NASA Langley Research Center CR-2729, 1976. Citado na página 26.
- THOMPSON, J. F.; THAMES, F. C.; MASTIN, C. W. Tomcat— a code for numerical generation of boundary-fitted curvilinear coordinate systems on fields containing any number of arbitrary two-dimensional bodies. *Journal of Computational Physics*, Elsevier, v. 24, n. 3, p. 274–302, 1977. Citado 3 vezes nas páginas 26, 51 e 61.
- THOMPSON, J. F.; WARSI, Z. U.; MASTIN, C. W. *Numerical grid generation: foundations and applications*. New York, USA: Elsevier North-Holland, 1985. v. 45. Citado 4 vezes nas páginas 55, 57, 58 e 59.
- TROTTERBERG, U.; OOSTERLEE, C. W.; SCHULLER, A. *Multigrid*. San Diego: Academic Press, 2001. Citado 16 vezes nas páginas 22, 23, 24, 30, 32, 43, 46, 67, 72, 73, 74, 75, 76, 96, 101 e 109.
- VERSTEEG, H. K.; MALALASEKERA, W. *An introduction to computational fluid dynamics: the finite volume method*. 2nd. ed. Harlow, England: Pearson Education, 2007. Citado 4 vezes nas páginas 21, 32, 35 e 74.
- WALLIS, J. T. Methods toward better multigrid solver convergence. *Applied Mathematics Research Report*, Daejeon, Coreia do Sul, 2008. Citado na página 32.
- WESSELING, P. *An Introduction to Multigrid Methods*. Philadelphia: R. T. Edwards, 2004. Citado 8 vezes nas páginas 32, 43, 44, 46, 67, 72, 73 e 75.
- WESSELING, P.; OOSTERLEE, C. W. Geometric multigrid with applications to computational fluid dynamics. *Computational and Applied Mathematics*, Elsevier, v. 128, n. 1, p. 311–334, 2001. Citado 2 vezes nas páginas 24 e 101.
- WIENANDS, R.; JOPPICH, W. *Practical Fourier analysis for multigrid methods*. Washington, USA: Chapman and Hall/CRC, 2004. Citado na página 32.
- WIENANDS, R.; JOPPICH, W. *Practical Fourier analysis for multigrid methods*. New York, USA: Chapman e Hall/CRC Press, 2005. Citado na página 42.

- XU, P. P.; SUN, Z. Z. A second-order accurate difference scheme for the two-dimensional burgers' system. *Numerical Methods for Partial Differential Equations: An International Journal*, Wiley Online Library, v. 25, n. 1, p. 172–194, 2009. Citado na página 29.
- YAVNEH, I. Why multigrid methods are so efficient. *Computing in science & engineering*, IEEE, v. 8, n. 6, p. 12–22, 2006. Citado na página 24.
- ZHANG, W.; ZHANG, C.; XI, G. An explicit chebyshev pseudospectral multigrid method for incompressible navier–stokes equations. *Computers & Fluids*, Elsevier, v. 39, n. 1, p. 178–188, 2010. Citado na página 30.
- ZHANLAV, T.; CHULUUNBAATAR, O.; ULZIIBAYAR, V. Higher-order accurate numerical solution of unsteady burgers' equation. *Applied Mathematics and Computation*, Elsevier, v. 250, p. 701–707, 2015. Citado na página 29.
- ZHAO, G.; YU, X.; ZHANG, R. The new numerical method for solving the system of two-dimensional burgers equations. *Computers & Mathematics with Applications*, Elsevier, v. 62, n. 8, p. 3279–3291, 2011. Citado 2 vezes nas páginas 29 e 81.
- ZHENG, Y. Stochastic stability of burgers equation. *Acta Mathematica Sinica, English Series*, Springer-Verlag Berlin Heidelberg, v. 32, n. 12, p. 1509–1514, 2016. Citado na página 33.

APÊNDICE A – VERIFICAÇÃO DOS CÓDIGOS COMPUTACIONAIS

Neste apêndice apresenta-se a verificação numérica dos códigos computacionais desenvolvidos neste trabalho. Para isso, realiza-se a análise do comportamento do erro de discretização de algumas variáveis de interesse, selecionadas para cada problema.

A.1 VERIFICAÇÃO

A verificação se divide em verificação do código computacional e da solução. A verificação do código constitui-se na asserção, tanto quanto possível, de que não existem erros ou inconsistências no algoritmo implementado. Já, a verificação da solução consiste no processo de quantificação dos erros decorrentes da simulação numérica (ARAKI, 2007).

No caso em que os erros de programação, de arredondamento e de iteração são controlados a ponto de serem desprezíveis, observa-se que o erro da solução numérica obtida é causado pelos erros de truncamento. Neste caso, o erro numérico recebe a denominação de erro de discretização (FERZIGER; PERIC, 2002), podendo ser expresso como

$$E_h(\phi) = b_1 h^{p_L} + b_2 h^{p^2} + b_3 h^{p^3} + \dots,$$

que é conhecida como equação geral do erro de discretização, sendo b_i , $i = 1, 2, 3, \dots$, os coeficientes independentes do espaçamento da malha (h) e os p_v (isto é, $p_L, p_2, p_3 \dots$) as ordens verdadeiras do erro. Por definição, as ordens verdadeiras correspondem aos expoentes de h dos termos não nulos. O menor dos expoentes de h na equação geral do erro de truncamento é denominado ordem assintótica (p_L), que deve ser um número inteiro positivo, satisfazendo a condição $p_L \geq 1$.

As estimativas do erro de discretização podem ser classificadas como estimativas *a priori* e *a posteriori* da obtenção da solução numérica (SZABO; BABUSKA, 1991). As estimativas de erro *a priori* são usadas para estimar a ordem do erro de discretização. Segundo Marchi (2001), com a estimativa de erro *a priori* é possível prever, antes da obtenção da solução numérica, o comportamento assintótico do erro de discretização com relação a redução do tamanho (h) dos elementos de malha.

As estimativas de erro *a posteriori* são usadas efetivamente para avaliar a magnitude do erro de discretização. Segundo Marchi (2001) as estimativas de erro *a posteriori* são usadas para avaliar a magnitude do erro de discretização. As estimativas de erro *a posteriori* podem ser realizadas a partir da solução numérica em uma única malha (usualmente utilizada para o método de elementos finitos) e para soluções numéricas obtidas em duas ou mais malhas diferentes (normalmente utilizada para os MDF e MVF).

As ordens efetiva e aparente do erro permitem verificar *a posteriori* se a ordem assintótica dos erros de discretização é atingida, uma vez que é um resultado teórico, obtido

a priori das soluções numéricas. A ordem efetiva (p_E) é a inclinação local da curva do erro de discretização da solução numérica da variável de interesse em função do tamanho (h) dos elementos da malha em um gráfico logarítmico (MARCIHI, 2001). Se a solução analítica, Φ , do problema é conhecida, pode-se calcular a ordem efetiva do erro utilizando-se Φ e duas soluções numéricas, ϕ_f e ϕ_g , obtidas nas malhas fina e grossa, respectivamente. A ordem efetiva é dada por

$$p_E = \frac{\log\left(\frac{\Phi - \phi_g}{\Phi - \phi_f}\right)}{\log(q)}, \quad (\text{A.1})$$

sendo q a razão de refino entre as malhas. A definição de razão de refino usada para malhas irregulares é dada por (ROACHE, 1994)

$$q = \left(\frac{N_f}{N_g}\right)^{\frac{1}{D}}, \quad (\text{A.2})$$

sendo N_f e N_g , o número de elementos da malha fina e grossa, respectivamente, e D a dimensão espacial do problema.

Se a solução analítica é desconhecida, a ordem assintótica do erro de truncamento é verificada através da ordem aparente (p_U), definida da seguinte forma

$$p_U = \frac{\log\left(\frac{\phi_g - \phi_{sg}}{\phi_f - \phi_g}\right)}{\log(q)}, \quad (\text{A.3})$$

sendo ϕ_{sg} a solução numérica obtida na malha supergrossa (MARCIHI, 2001).

A verificação numérica dos códigos computacionais desenvolvidos neste trabalho foi realizada mediante a análise do comportamento do erro de discretização de algumas variáveis de interesse, selecionadas para cada problema. Para o problema de Poisson com domínio no formato de L, usando malhas curvilíneas estruturadas não ortogonais, consideram-se como variáveis de interesse a norma infinito do erro na solução T e a temperatura média, calculada pela regra do retângulo. Para as equações de Burgers usando o mesmo domínio e as mesmas malhas, consideram-se as normas infinito das soluções u e v e a força do fluido na fronteira $y = 1$. Todos os problemas estudados têm soluções analíticas, cujas equações e respectivas condições de contorno foram especificadas no capítulo 4 deste trabalho. Nesta verificação, foram consideradas as ordens efetiva e aparente do erro de discretização e o comportamento do erro de discretização mediante o refino da malha. Para isso, foram determinados os resultados analíticos e numéricos obtidos para cada variável de interesse e calculadas as respectivas ordens efetiva e aparente.

A.2 ERRO DE DISCRETIZAÇÃO

Nesta seção, apresenta-se um estudo do erro de discretização obtido nas soluções numéricas das equações de Poisson e de Burgers. São analisadas as ordens efetiva e aparente

bem como o comportamento do erro de discretização mediante o refino da malha. Foram determinados os resultados analíticos e numéricos obtidos para cada variável de interesse e calculadas as respectivas ordens efetiva e aparente. Para minimizar o erro de iteração, cada programa foi executado até atingir o erro de máquina. Esta análise busca verificar se as ordens efetiva e aparente do erro de discretização tendem às respectivas ordens assintóticas com o refino da malha.

A ordem assintótica do erro de discretização para problemas com aproximações UDS é $p_L = 1$ e com aproximações CDS é $p_L = 2$. Para os problemas considerados, os termos difusivos foram discretizados mediante o CDS. Os termos advectivos foram discretizados mediante o CDS, com correção adiada sobre o UDS, de forma que a ordem assintótica do erro de discretização também é $p_L = 2$. As FIGURAS A.1 a A.8 ilustram o comportamento das ordens efetiva e aparente dos erros para as variáveis de interesse em questão de cada problema.

FIGURA A.1 – ORDEM EFETIVA DA NORMA INFINITO DO ERRO NA SOLUÇÃO T E DA TEMPERATURA MÉDIA EM FUNÇÃO DO ESPAÇAMENTO DA MALHA, CALCULADA PARA A EQUAÇÃO DE POISSON PARA MALHAS GERADAS USANDO INTERPOLAÇÃO DE LAGRANGE

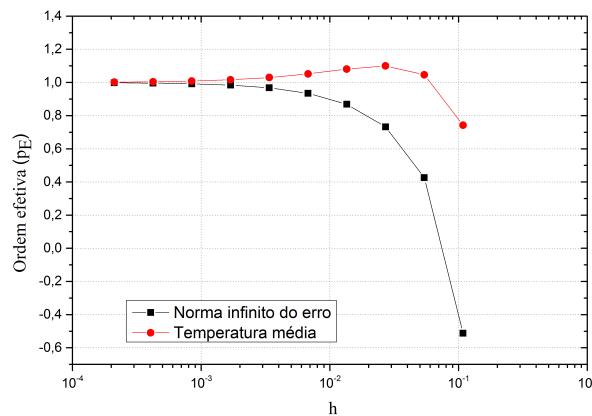


FIGURA A.2 – ORDEM APARENTE DA NORMA INFINITO DO ERRO NA SOLUÇÃO T E DA TEMPERATURA MÉDIA EM FUNÇÃO DO ESPAÇAMENTO DA MALHA, CALCULADA PARA A EQUAÇÃO DE POISSON PARA MALHAS GERADAS USANDO INTERPOLAÇÃO DE LAGRANGE

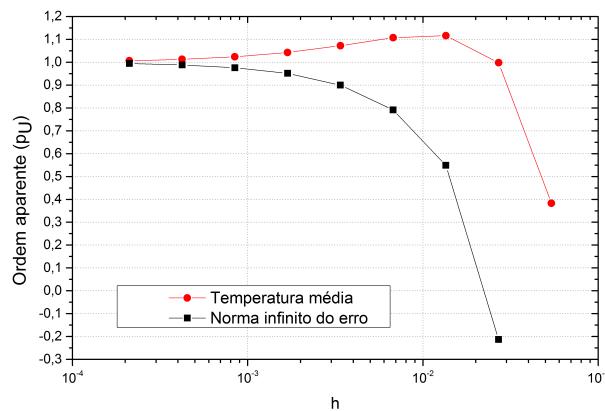


FIGURA A.3 – ORDEM EFETIVA DA NORMA INFINITO DO ERRO NA SOLUÇÃO T E DA TEMPERATURA MÉDIA EM FUNÇÃO DO ESPAÇAMENTO DA MALHA, CALCULADA PARA A EQUAÇÃO DE POISSON PARA MALHAS GERADAS USANDO EQUAÇÕES ELÍPTICAS

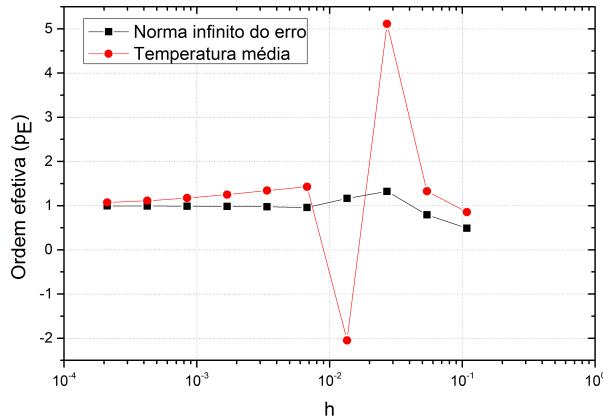
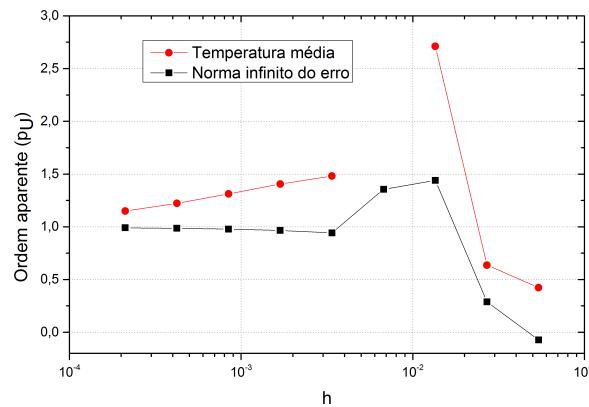


FIGURA A.4 – ORDEM APARENTE DA NORMA INFINITO DO ERRO NA SOLUÇÃO T E DA TEMPERATURA MÉDIA EM FUNÇÃO DO ESPAÇAMENTO DA MALHA, CALCULADA PARA A EQUAÇÃO DE POISSON PARA MALHAS GERADAS USANDO EQUAÇÕES ELÍPTICAS



Como pode ser observado nas FIGURAS A.1 a A.4, as ordens efetiva e aparente para a norma infinito do erro na solução T e a temperatura média, para o problema de Poisson, tendem à unidade mediante o refino da malha, enquanto que a ordem assintótica é $p_L = 2$. Para as equações de Burgers, as normas infinito das soluções u e v também apresentam esse comportamento, conforme pode ser observado nas FIGURAS A.5 a A.8, exceto para a força do fluido na fronteira $y = 1$. Para essa variável o comportamento ainda não é monotônico, sendo necessário malhas mais refinadas para observar o comportamento dessa variável.

FIGURA A.5 – ORDEM EFETIVA DA NORMA INFINITO DO ERRO NA SOLUÇÃO E DA FORÇA DO FLUIDO NA FRONTEIRA EM FUNÇÃO DO ESPAÇAMENTO DA MALHA, CALCULADA PARA AS EQUAÇÕES DE BURGERS PARA MALHAS GERADAS USANDO INTERPOLAÇÃO DE LAGRANGE

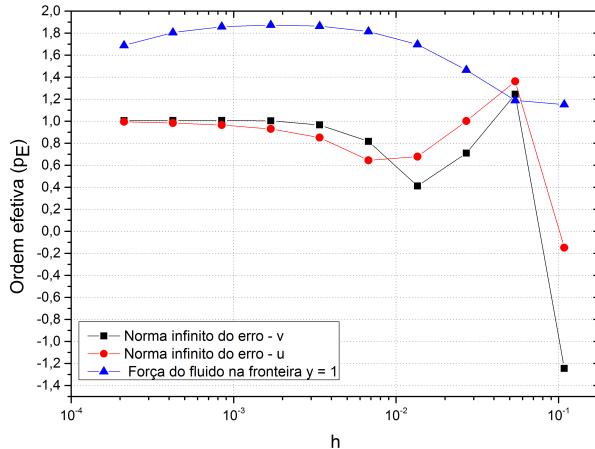


FIGURA A.6 – ORDEM APARENTE DA NORMA INFINITO DO ERRO NA SOLUÇÃO E DA FORÇA DO FLUIDO NA FRONTEIRA EM FUNÇÃO DO ESPAÇAMENTO DA MALHA, CALCULADA PARA AS EQUAÇÕES DE BURGERS PARA MALHAS GERADAS USANDO INTERPOLAÇÃO DE LAGRANGE

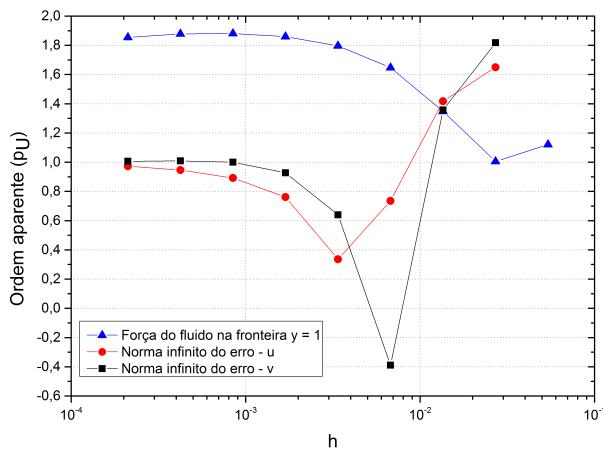


FIGURA A.7 – ORDEM EFETIVA DA NORMA INFINITO DO ERRO NA SOLUÇÃO E DA FORÇA DO FLUIDO NA FRONTEIRA EM FUNÇÃO DO ESPAÇAMENTO DA MALHA, CALCULADA PARA AS EQUAÇÕES DE BURGERS PARA MALHAS GERADAS USANDO EQUAÇÕES ELÍPTICAS

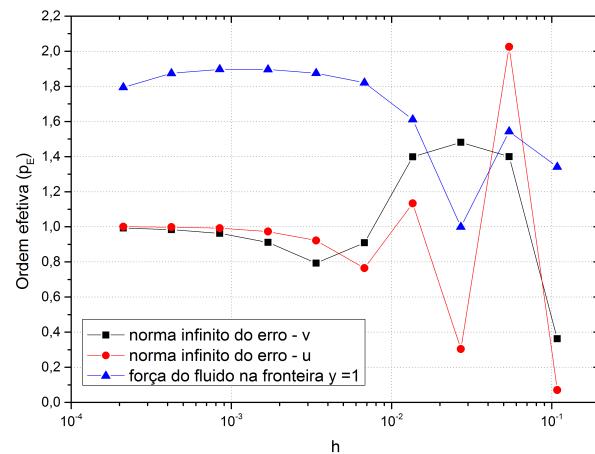
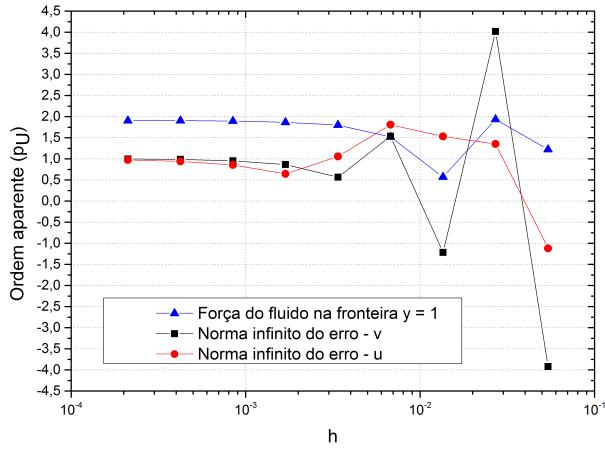


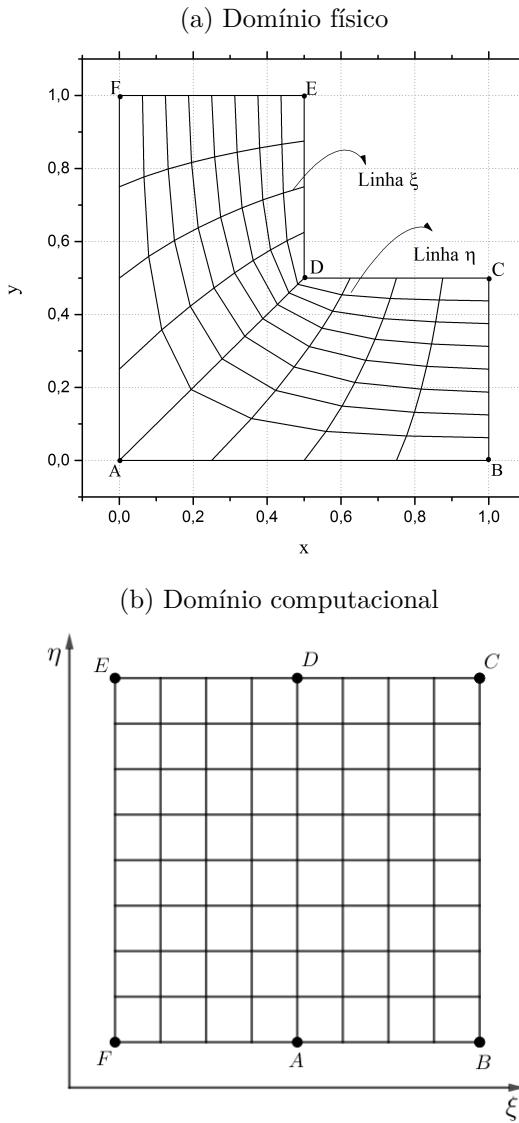
FIGURA A.8 – ORDEM APARENTE DA NORMA INFINITO DO ERRO NA SOLUÇÃO E DA FORÇA DO FLUIDO NA FRONTEIRA EM FUNÇÃO DO ESPAÇAMENTO DA MALHA, CALCULADA PARA AS EQUAÇÕES DE BURGERS PARA MALHAS GERADAS USANDO EQUAÇÕES ELÍPTICAS



Para os dois problemas estudados, os resultados mostram que ocorre uma degeneração nas ordens efetiva e aparente. Segundo Fletcher (1991) isso acontece porque discretizando uma geometria no formato de L com malhas curvilíneas, conforme mostra, por exemplo, a FIGURA A.9, ocorre uma excessiva distorção no mapeamento do domínio físico sobre o domínio computacional. Pode-se observar na FIGURA A.9 que a maior distorção ocorre próximo ao segmento \overline{AD} . Essa distorção no mapeamento do domínio físico sobre o domínio computacional afeta a acurácia da solução local (FLETCHER, 1991). De acordo com Ferziger e Peric (2002), embora a malha transformada pareça mais simples que a malha não transformada, as informações sobre a sua complexidade estão contidas nas métricas e no jacobiano da transformação. Dessa forma, enquanto a discretização da equação transformada é acurada, o cálculo das métricas e do jacobiano da transformação introduzem erros de discretização adicionais. Mas, mesmo assim, as transformações de coordenadas são frequentemente empregadas para transformar malhas não ortogonais em malhas cartesianas.

Devido a procedimentos numéricos para a geração das malhas e também por razões de acurácia, recomenda-se usar o método *multigrid* com *solvers* mais eficientes para diminuir o efeito do forte acoplamento de incógnitas em uma direção, causado por volumes distorcidos na discretização do domínio físico (OOSTERLEE; WESSELING, 1993). Ainda, de acordo com Fletcher (1991) e Oosterlee e Wesseling (1993), para obter soluções mais acuradas para o domínio L, pode-se mapear o domínio computacional usando de multiblocos, cada qual apresentando uma malha em separado que deverá ser unida corretamente aos seus vizinhos, em que as fronteiras do domínio computacional são exatamente as fronteiras do domínio físico.

FIGURA A.9 – DOMÍNIO NO FORMATO DE L E SEU RESPECTIVO DOMÍNIO COMPUTACIONAL



Para a verificação da consistência das soluções numéricas, estuda-se o comportamento do erro de discretização mediante o refino da malha. Neste estudo, espera-se que o erro tenda a zero quando o espaçamento (h) da malha tende a zero, o que significa que o erro de discretização deve diminuir com o aumento do número de incógnitas. As FIGURAS A.10 a A.13 mostram o comportamento do erro de discretização, para todas as variáveis de interesse, relativas aos problemas estudados.

Como pode ser observado nas FIGURAS A.10 a A.13, o comportamento do erro de discretização está de acordo com o esperado. Isto é, o erro de discretização diminui com o refino da malha. Isto ocorre para todas as variáveis analisadas, correspondentes aos dois problemas estudados.

FIGURA A.10 – ERRO NUMÉRICO EM FUNÇÃO DO ESPAÇAMENTO DA MALHA PARA A EQUAÇÃO DE POISSON PARA MALHAS GERADAS USANDO INTERPOLAÇÃO DE LAGRANGE

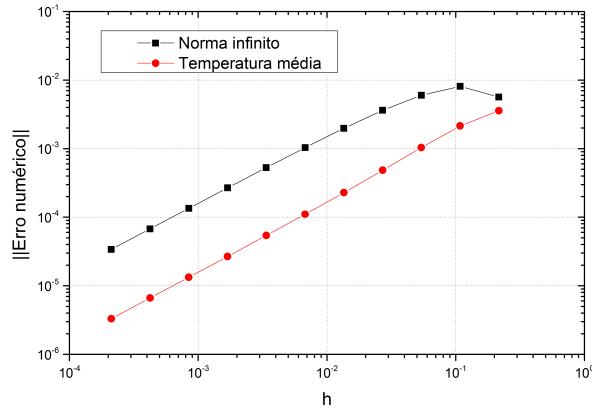


FIGURA A.11 – ERRO NUMÉRICO EM FUNÇÃO DO ESPAÇAMENTO DA MALHA PARA A EQUAÇÃO DE POISSON PARA MALHAS GERADAS USANDO EQUAÇÕES ELÍPTICAS

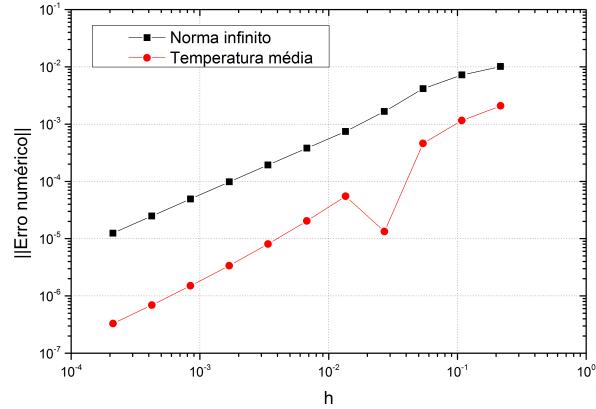


FIGURA A.12 – ERRO NUMÉRICO EM FUNÇÃO DO ESPAÇAMENTO DA MALHA PARA AS EQUAÇÕES DE BURGERS PARA MALHAS GERADAS USANDO INTERPOLAÇÃO DE LAGRANGE

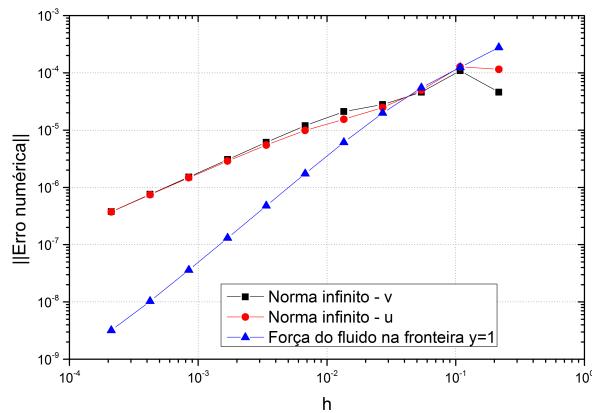


FIGURA A.13 – ERRO NUMÉRICO EM FUNÇÃO DO ESPAÇAMENTO DA MALHA PARA AS EQUAÇÕES DE BURGERS PARA MALHAS GERADAS USANDO EQUAÇÕES ELÍPTICAS

