

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
CURSO DE GRADUAÇÃO EM ENGENHARIA MECÂNICA**

EDERSON AUGUSTO GREIN

**AVALIAÇÃO DO DESEMPENHO DO MÉTODO MULTIGRID DE
CORREÇÕES ADITIVAS NA SIMULAÇÃO NUMÉRICA DE
RESERVATÓRIOS DE PETRÓLEO APLICANDO MALHAS POLIÉDRICAS**

Florianópolis
2012

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
CURSO DE GRADUAÇÃO EM ENGENHARIA MECÂNICA**

EDERSON AUGUSTO GREIN

**AVALIAÇÃO DO DESEMPENHO DO MÉTODO MULTIGRID DE
CORREÇÕES ADITIVAS NA SIMULAÇÃO NUMÉRICA DE
RESERVATÓRIOS DE PETRÓLEO APLICANDO MALHAS POLIÉDRICAS**

Trabalho apresentado ao Curso de Graduação em Engenharia Mecânica da Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do título de Engenheiro Mecânico

Orientador: Prof. Clovis Raimundo Maliska, Ph.D.

Coorientador: Fernando Sandro Velasco Hurtado, Dr. Eng.

Florianópolis
2012

Catálogo na fonte elaborada pela biblioteca da
Universidade Federal de Santa Catarina

Grein, Ederson Augusto, 1989 -

Avaliação do Desempenho do Método Multigrid de Correções Aditivas na Simulação Numérica de Reservatórios de Petróleo Aplicando Malhas Poliédricas / Ederson Augusto Grein - 2012.

85 f. : il. color. ; 3 cm

Orientador: Clovis Raimundo Maliska.

Coorientador: Fernando Sandro Velasco Hurtado.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Curso de Engenharia Mecânica, 2012.

1. Simulação numérica de reservatórios de petróleo. 2. Malhas poliédricas. 3. Método Multigrid de Correções Aditivas. I. Maliska, Clovis Raimundo. II. Universidade Federal de Santa Catarina. Curso de Engenharia Mecânica

EDERSON AUGUSTO GREIN

**AVALIAÇÃO DO DESEMPENHO DO MÉTODO MULTIGRID DE
CORREÇÕES ADITIVAS NA SIMULAÇÃO NUMÉRICA DE
RESERVATÓRIOS DE PETRÓLEO APLICANDO MALHAS POLIÉDRICAS**

Este Trabalho de Curso foi julgado adequado para a obtenção do título de Engenheiro Mecânico e aprovado em sua forma final pela Comissão examinadora e pelo Curso de Graduação em Engenharia Mecânica da Universidade Federal de Santa Catarina.

Prof. Lauro Cesar Nicolazzi, Dr.
Coordenador do Curso

Prof. Dylton do Vale Pereira Filho, M.Sc.
Professor da Disciplina

Comissão Examinadora:

Prof. Clovis Raimundo Maliska, Ph.D.
Orientador

Fernando Sandro Velasco Hurtado, Dr. Eng.
Coorientador

Prof. Mário César Zambaldi, Dr.
Departamento de Matemática, UFSC

Florianópolis
2012

Aos meus pais, por todo o apoio.

AGRADECIMENTOS

Agradeço primeiramente aos meus pais Aliomar e Marlene, por todo o apoio e incentivo, sobretudo nesses anos de graduação. Agradeço também aos meus irmãos Glederson e Anderson, sempre presentes.

À Universidade Federal de Santa Catarina e ao Departamento de Engenharia Mecânica, pela excelência do curso oferecido. À turma 08.1, sempre unida e de apoio mútuo. Em especial, agradeço ao meu grande amigo Victor Magri, o Vitinho, com o qual fiz praticamente todas as disciplinas da graduação.

Ao Laboratório de Simulação Numérica em Mecânica dos Fluidos e Transferência de Calor – SINMEC – e ao Programa de Recursos Humanos da ANP pela estrutura disponibilizada e pelo apoio financeiro.

Ao professor Clovis Raimundo Maliska por ter me admitido no SINMEC e por ter me orientado nesse trabalho. Agradeço também ao Fernando, ao Gustavo, ao Aymar e ao Giovani, integrantes do projeto Malhas de Transição, do qual diversos recursos foram utilizados. Em particular, agradeço ao Fernando pela revisão detalhada do trabalho e ao Aymar pelas discussões acerca do multigrid e da estruturação de sua biblioteca.

A todos aqueles do SINMEC ainda não citados: Fábio, Axel, Emílio, Tati, Terêncio, Arthur, Dalpi, Leonardo, Tada, Christiano, Fernando e Hermínio.

Ao Programa Avançado de Matemática (PAM). Aos professores Jáuber Cavalcante de Oliveira, que ministrou as quatro disciplinas de Cálculo oferecidas pelo programa, e Mário César Zambaldi, professor das disciplinas de Álgebra. Os conhecimentos adquiridos com o programa foram de fundamental importância à execução desse trabalho.

“Ideas are like rabbits. You get a couple and learn how to handle them, and pretty soon you have a dozen.”

John Steinbeck

RESUMO

Métodos multigrid têm sido frequentemente empregados na solução de sistemas de equações lineares de grande dimensão, sobretudo aqueles provenientes de discretizações de equações diferenciais. Neles o sistema linear original é decomposto num conjunto de outros sistemas, cada um associado a uma malha de grau de refino progressivamente menor. Para cada um desses sistemas um método de solução de sistemas lineares, ou *solver*, deve ser aplicado. Esse trabalho se destina à avaliação da influência do *solver* utilizado no desempenho global de um método multigrid algébrico. Serão aqui estudados os métodos do Gradiente Conjugado, de Gauss-Seidel, GMRES e MINRES. Os sistemas lineares serão provenientes de um simulador de reservatórios de petróleo que emprega malhas poligonais e poliédricas. Essas malhas apresentam grande generalidade e, dependendo do esquema numérico, podem resultar em sistemas lineares relativamente difíceis de se resolver.

Palavras-chave: Multigrid, sistemas lineares, *solvers*, simulação numérica, reservatório de petróleo, malha poliédrica.

ABSTRACT

Multigrid methods have been applied to solve large systems of linear equations, mainly those systems coming from the discretization of differential equations. These methods generate a set of new linear systems, each one of them associated to a coarser grid. The multigrid, however, does not solve those linear systems by its own. It's necessary to specify for each one a linear system solver. The objective of this work is to study the overall performance of an algebraic multigrid applying as solvers the Gauss-Seidel method, the Conjugate Gradient, the Generalized Minimal Residual (GMRES) and the Minimal Residual (MINRES). The linear systems will come from a reservoir simulator that uses polygonal and polyhedral grids. Those grids are quite general and, depending on the numerical scheme, good solvers may be required to achieve convergence.

Keywords: Multigrid, linear systems, solvers, reservoir simulation, polyhedral grids.

LISTA DE ILUSTRAÇÕES

2.1	Superfície S genérica	19
3.1	Entidades fundamentais das malhas poliédricas (adaptado de Maliska et al. (2012))	21
3.2	Entidades fundamentais das malhas poligonais	22
3.3	Volume de controle genérico p em esquema <i>cell-center</i> (adaptado de Maliska et al. (2012))	23
3.4	Novo sistema de coordenadas associado à face f (adaptado de Maliska et al. (2012))	24
3.5	Células vizinhas p e q (adaptado de Maliska et al. (2012))	26
3.6	Volume de controle p e seus vizinhos (adaptado de Maliska et al. (2012))	27
4.1	Soluções parciais da malha 11×11 após diferentes números de iterações	34
4.2	Soluções parciais da malha 51×51 após diferentes números de iterações	35
4.3	Problema de condução de calor unidimensional	36
4.4	Exemplo de diferentes níveis de malhas utilizadas no multigrid	37
4.5	Ciclos multigrid fixos: V (a), F (b) e W (c)	41
6.1	Malha regular bidimensional	56
6.2	Resultados do multigrid com método de Gauss-Seidel e esquema semi-implícito	57
6.3	Resultados do multigrid com método do Gradiente Conjugado e esquema semi-implícito	58
6.4	Resultados do multigrid com método MINRES e esquema semi-implícito	58
6.5	Resultados dos diferentes métodos com número de iterações ótimo e esquema semi-implícito	59
6.6	Resultados do multigrid com Gradiente Conjugado e método totalmente implícito	61
6.7	Resultados do multigrid com método MINRES e esquema totalmente implícito	61

6.8	Resultados dos diferentes métodos com número de iterações ótimo e esquema totalmente implícito	62
6.9	Erro em função do número de ciclos multigrid executados	63
6.10	Malha bidimensional de grande razão de aspecto	64
6.11	Resultados do multigrid com método do Gradiente Conjugado e esquema semi-implícito	65
6.12	Resultados do multigrid com método de Gauss-Seidel e esquema semi-implícito	65
6.13	Resultados do multigrid com método GMRES de dimensão 2 e esquema semi-implícito	66
6.14	Resultados do multigrid com método GMRES de dimensão 10 e esquema semi-implícito	66
6.15	Resultados do multigrid com método MINRES e esquema semi-implícito	67
6.16	Resultados dos diferentes métodos com número de iterações ótimo e esquema semi-implícito	68
6.17	Resultados do multigrid com método de Gauss-Seidel e esquema totalmente implícito	69
6.18	Resultados do multigrid com método do Gradiente Conjugado e esquema totalmente implícito	69
6.19	Resultados do multigrid com método GMRES de dimensão dois e esquema totalmente implícito	70
6.20	Resultados do multigrid com método GMRES de dimensão 10 e esquema totalmente implícito	71
6.21	Resultados do multigrid com método MINRES e esquema totalmente implícito	71
6.22	Resultados dos diferentes métodos com número de iterações ótimo e esquema totalmente implícito	72
6.23	Decaimento do erro em função do número de ciclos multigrid	72
6.24	Elemento da malha tridimensional com distorção aleatória	73
6.25	Resultados do multigrid com método do Gradiente Conjugado e esquema semi-implícito	74

6.26	Resultados do multigrid com método de Gauss-Seidel e esquema semi-implícito	75
6.27	Resultados do multigrid com método GMRES de dimensão cinco e esquema semi-implícito	76
6.28	Resultados dos diferentes métodos com número de iterações ótimo e esquema semi-implícito	76
6.29	Resultados dos diferentes métodos com número de iterações ótimo e esquema totalmente implícito	78
6.30	Decaimento do erro em função do número de ciclos multigrid executados . . .	79

LISTA DE TABELAS

6.1	Valores mantidos fixos para os parâmetros de controle do multigrid	54
6.2	Complexidade do multigrid com diversos métodos e esquema semi-implícito .	59
6.3	Complexidade do multigrid com diversos métodos e esquema totalmente implícito	63
6.4	Complexidade do multigrid com diversos métodos e esquema semi-implícito .	67
6.5	Complexidade do multigrid com diversos métodos e esquema semi-implícito .	77
6.6	Complexidade do multigrid com diversos métodos e esquema totalmente implícito	78

LISTA DE LISTAGENS

4.1	Algoritmo recursivo do ciclo V	42
4.2	Algoritmo recursivo do ciclo F	42
4.3	Algoritmo recursivo do ciclo W	43
5.1	Algoritmo do MINRES	48
5.2	Método de Arnoldi usando ortogonalização de Gram-Schmidt modificada (adaptado de Saad (2000))	49
5.3	<i>Restarted</i> GMRES, adaptado de Saad (2000)	50
5.4	Algoritmo do Gradiente Conjugado (adaptado de Strang (2006b))	52

SUMÁRIO

1	Introdução	15
1.1	Objetivos	16
1.1.1	Objetivos gerais	16
1.1.2	Objetivos específicos	17
1.2	Organização do trabalho	17
2	Modelagem Matemática	18
3	Discretização das Equações	20
3.1	Malhas Poligonais e Poliédricas	21
3.2	Integração da Equação e Aproximação dos Fluxos	23
3.3	Reconstrução do gradiente	26
3.3.1	Esquema semi-implícito	28
3.3.2	Esquema totalmente implícito	29
3.4	Observações	30
4	O Método Multigrid de Correções Aditivas	32
4.1	Introdução	32
4.2	Multigrid geométrico e multigrid algébrico	38
4.3	Operadores de interpolação e restrição	39
4.4	Ciclos Multigrid	40
5	Métodos de Solução de Sistemas Lineares Empregados no Multigrid	45
5.1	Métodos Iterativos Básicos	45
5.1.1	Método de Jacobi	45
5.1.2	Método de Gauss-Seidel	46

5.2	Métodos de Projeção	47
5.2.1	Método MINRES	48
5.2.2	Método GMRES	48
5.2.3	Método do Gradiente Conjugado	51
5.3	Método baseado na fatoração LU	52
6	Resultados	54
6.1	Malha regular bidimensional	56
6.1.1	Esquema semi-implícito	56
6.1.2	Esquema totalmente implícito	60
6.2	Malha com grande razão de aspecto	63
6.2.1	Esquema semi-implícito	64
6.2.2	Esquema totalmente implícito	68
6.3	Malha tridimensional com distorção aleatória	73
6.3.1	Esquema semi-implícito	74
6.3.2	Esquema totalmente implícito	77
7	Comentários e Conclusões	80
7.1	Sugestões para trabalhos futuros	81
	Referências	83

1 INTRODUÇÃO

A simulação numérica de reservatórios é uma ferramenta que permite prever o desempenho de certa estratégia de produção aplicada a um reservatório de petróleo. É possível, por exemplo, estimar variáveis como o fator de recuperação, o tempo de produção e a relação entre o óleo e o gás produzidos. Nos últimos 30 anos sua aceitação tem crescido progressivamente, sobretudo devido ao desenvolvimento dos computadores e dos algoritmos de solução de equações diferenciais. Para a engenharia de petróleo, em particular, a simulação numérica adquire importância ainda maior porque os fenômenos físicos estudados acontecem em múltiplas escalas, o que vem a dificultar, ou mesmo impossibilitar, a execução de atividades experimentais.

A essência da simulação numérica consiste no desenvolvimento de modelos matemáticos que descrevam os fenômenos físicos e de técnicas que venham a resolver esses modelos. A obtenção de bons resultados está condicionada à contemplação de maneira adequada de ambos os aspectos. É bastante lógico que, por melhor que seja a técnica numérica, a simulação esteja condenada ao fracasso caso o modelo matemático não corresponda a física do problema. Do mesmo modo, não é suficiente ter um bom modelo matemático se sua solução é impraticável. A fim de tornar as simulações mais realísticas deve-se, pois, buscar o desenvolvimento de ambos os aspectos.

Esse trabalho se enquadra na área de desenvolvimento de técnicas numéricas para a solução eficiente dos modelos. O enfoque aqui é dado a uma etapa fundamental a praticamente todos os simuladores cujos modelos matemáticos se baseiam em equações diferenciais: a solução de sistemas de equações lineares. Segundo Maliska (2004), normalmente cerca de 60 a 70% do tempo total de simulação é dispendido computando as soluções desses sistemas, de modo que qualquer melhora, mesmo que pequena, resulta em significativa economia de tempo. Há naturalmente interesse prático na redução do tempo de computação. Simulações realísticas, que exigem malhas de elevado grau de refino, podem requerer dias, ou até mesmo semanas, para a conclusão. Reduzir o tempo viabiliza, pois, a execução de mais simulações, permitindo uma melhor compreensão das variáveis envolvidas bem como o emprego de técnicas de otimização.

Os métodos multigrid despontam como um dos mais eficientes conhecidos atualmente para a solução de sistemas de equações lineares. Eles foram concebidos justamente para a solução de sistemas provenientes da discretização de equações diferenciais parciais elípticas (STÜBEN, 2001). De forma simplificada, os métodos multigrid decompõem o sistema linear

original num conjunto de outros sistemas menores, cada um associado a uma dada malha numa hierarquia de malhas cada vez mais grosseiras. Esses sistemas são resolvidos conjuntamente, havendo nesse processo transmissão de informação entre eles. Porém, o multigrid não os resolve por si só. Faz-se necessário para cada nível de malha definir um método de solução de sistema linear (*solver*). O objetivo principal desse trabalho é investigar a influência desses métodos sobre o desempenho global do multigrid.

Para a avaliação do multigrid se fará uso do simulador que vem sendo desenvolvido no projeto Malhas de Transição (MaTra), realizado no laboratório SINMEC/UFSC com apoio da Rede Temática de Simulação e Gerenciamento de Reservatórios (SIGER). O principal diferencial desse simulador é a capacidade de utilizar malhas poligonais e poliédricas, cuja característica principal é a de seus elementos admitirem um número arbitrário de faces e de vértices. Dada a generalidade dessas malhas, os métodos numéricos nelas empregados podem conduzir a sistemas lineares relativamente difíceis de se resolver, favorecendo uma boa avaliação do multigrid. Além disso, no projeto MaTra está disponível um vasto conjunto de malhas, de diversas características e graus de refino, facilitando o processo de avaliação do multigrid.

1.1 OBJETIVOS

1.1.1 OBJETIVOS GERAIS

O objetivo principal deste trabalho é a avaliação do método Multigrid de Correções Aditivas (*Additive Correction Multigrid* (HUTCHINSON; RATHBY, 1986)) na simulação de reservatórios de petróleo utilizando-se malhas poliédricas. Mais precisamente, o objetivo é:

- Investigar a influência do *solver* interno ao multigrid sobre o desempenho global do processo de solução de sistemas de equações lineares provenientes da simulação numérica de reservatórios de petróleo utilizando-se malhas poliédricas.

Espera-se determinar quais são as configurações que resultem em menor tempo computacional e em maior robustez.

1.1.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos desse trabalho são:

- (a) Desenvolver uma biblioteca em C++ com uma implementação do multigrid apta a aceitar diferentes *solvers* em cada nível;
- (b) Implementar os métodos: de Jacobi; de Gauss-Seidel; *Minimal Residual* (MINRES); *Generalized Minimal Residual* (GMRES); do Gradiente Conjugado; e do baseado na fatoração LU.
- (c) Adaptar o simulador Orion, desenvolvido no projeto MaTra pelo SINMEC, para a utilização da nova biblioteca multigrid;
- (d) Executar diversos experimentos numéricos envolvendo as malhas utilizadas no projeto MaTra;

1.2 ORGANIZAÇÃO DO TRABALHO

O capítulo 2 é dedicado à dedução do modelo matemático utilizado nesse trabalho, ao passo que o capítulo 3 envolve o processo de discretização desse modelo na malha do domínio. Serão descritas as características básicas das malhas poliédricas e a integração das equações diferenciais. Boa parte do capítulo 3 é dedicada aos métodos de aproximação dos fluxos de massa nas faces dos volumes de controle. No capítulo 4 serão introduzidos os métodos multigrid. Dois problemas-exemplo serão apresentados para melhor compreender seus fundamentos. Serão também descritos os operadores de restrição e interpolação, a partir dos quais é feita a transmissão de informação entre os níveis do multigrid, e os ciclos (fixos) F , V e W . O capítulo 5 é destinado aos métodos de solução de sistemas lineares empregados dentro do multigrid. Nos capítulos 6 e 7 estão, respectivamente, os resultados e as conclusões.

2 MODELAGEM MATEMÁTICA

Este capítulo se destina basicamente ao estabelecimento do modelo matemático que traduz a física do problema. Uma vez que não é objetivo desse trabalho a avaliação da acurácia desse modelo, serão feitas diversas hipóteses a fim de torná-lo tão simples quanto possível.

Considere um escoamento monofásico e incompressível num reservatório de petróleo consolidado. Esse reservatório é um meio poroso e, portanto, o fluido se move através dos canais delimitados pelos grãos que formam a estrutura porosa. Obviamente a determinação da geometria desses canais é inviável, sobretudo observando-se que os reservatórios de petróleo possuem extensão da ordem de quilômetros e os canais, dimensões que podem ser inferiores a um milímetro. É usual, então, analisar o fenômeno numa escala suficientemente grande tal que os grãos se tornem indistintos. Embora não seja possível assim determinar as variáveis de interesse em cada ponto do domínio, pode-se conhecer seus valores médios, o que em geral é suficiente do ponto de vista da engenharia.

A equação da conservação da massa para meio poroso consolidado e fluido incompressível, deduzida em Bejan (2004), é dada por

$$\nabla \cdot \mathbf{v} = 0. \quad (2.1)$$

A grandeza \mathbf{v} , de componentes u , v e w , é a velocidade média na área, também conhecida como velocidade de Darcy (KAVIANY, 1995). Considerando-se a superfície genérica S ilustrada na figura 2.1, de dimensões suficientemente grandes em relação aos poros, a velocidade de Darcy do fluido que a atravessa é definida por

$$\mathbf{v} = \left(\frac{1}{S_x} \int_S u_p \, dS_x, \quad \frac{1}{S_y} \int_S v_p \, dS_y, \quad \frac{1}{S_z} \int_S w_p \, dS_z \right). \quad (2.2)$$

Observa-se que se está integrando na área total da superfície, incluindo, portanto, as regiões em que o meio é sólido.

Apenas a equação (2.1) não é suficiente para a descrição matemática do problema. Há três variáveis – as três componentes da velocidade – e apenas uma equação. Nos problemas tradicionais de mecânica de fluidos as equações adicionais são as da conservação da quantidade de movimento. Porém, elas não podem ser aplicadas neste caso devido à escala de análise utilizada, que não considera o formato dos poros. Em contrapartida, dispõe-se da lei

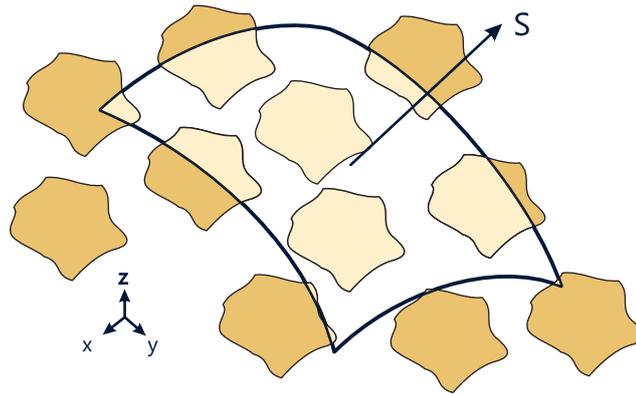


Figura 2.1: Superfície S genérica

de Darcy. Em sua forma generalizada ela é expressa por (PEACEMAN, 1977)

$$\mathbf{v} = -\frac{\mathbf{K}}{\mu} (\nabla P - \rho \vec{g}), \quad (2.3)$$

em que \mathbf{K} é o tensor permeabilidade absoluta do meio, propriedade análoga à condutividade térmica. Definindo-se a função potencial Φ como

$$\Phi = P - \rho(\vec{g} \cdot \vec{r}_p), \quad (2.4)$$

em que r_p é o vetor posição em relação à origem do sistema de coordenadas, pode-se chegar a uma expressão mais conveniente para a lei de Darcy:

$$\mathbf{v} = -\frac{\mathbf{K}}{\mu} \nabla \Phi. \quad (2.5)$$

Substituindo a equação (2.5) na equação (2.1) obtém-se uma equação diferencial de segunda ordem cuja única variável é a função potencial:

$$\nabla \cdot \left(\frac{\mathbf{K}}{\mu} \nabla \Phi \right) = 0. \quad (2.6)$$

Essa equação, juntamente com as condições de contorno, compõe o modelo matemático do problema.

3 DISCRETIZAÇÃO DAS EQUAÇÕES

Há basicamente duas formas de resolver a equação deduzida anteriormente: por meio de métodos analíticos ou de métodos numéricos. A abordagem analítica apresenta a vantagem de que a solução obtida se apresenta em forma fechada, válida para cada ponto do domínio. Entretanto, os métodos analíticos são muito restritivos. Segundo Maliska (2004), a necessidade de se adotar um grande número de hipóteses simplificativas a fim de tornar os métodos analíticos viáveis faz com que suas soluções acabem se desviando demasiadamente da realidade. Já os métodos numéricos apresentam poucas restrições, podendo ser aplicados nos mais complexos problemas, por mais complicadas que sejam a geometria e as condições de contorno (MALISKA, 2004). Embora a solução seja aproximada, ela em geral é suficiente no contexto da engenharia. Esse trabalho se enquadra dentro do contexto dos métodos numéricos de solução de equações diferenciais, os quais geralmente requerem a solução de um conjunto de sistemas de equações lineares.

Para resolver numericamente a equação diferencial (2.6) será empregado o Método dos Volumes Finitos (FVM, na sigla em inglês). Esse método apresenta a importante característica de ser conservativo em nível discreto. Isso significa que as aproximações aplicadas no processo de discretização das equações são tais que as leis de conservação são respeitadas localmente, para cada volume de controle. A conservação das propriedades acaba por tornar o FVM bastante robusto, sendo o mais utilizado nos *softwares* comerciais destinados a simulações de problemas envolvendo mecânica dos fluidos, transferência de calor e também na simulação de reservatórios.

Para aplicar o FVM faz-se necessário efetuar a discretização do domínio, que consiste em dividi-lo numa série de subdomínios, denominados elementos. À totalidade desses elementos dá-se o nome de malha. Com base nesses elementos são definidos os volumes de controle, havendo para isso duas formulações básicas: a *cell center* e a *cell vertex* (MALISKA, 2004). Na abordagem *cell center* os volumes de controle e os elementos são coincidentes, associando-se os valores das variáveis discretas aos seus centros. Já na formulação *cell vertex* os volumes de controle são construídos ao redor dos vértices dos elementos, nos quais as variáveis discretas estão localizadas. Uma vez que as malhas empregadas nesse trabalho são poliédricas, apresentando grande generalidade, é preferível a utilização da formulação *cell center*. A formulação *cell vertex* acarretaria na criação de volumes de controle desnecessariamente complexos.

3.1 MALHAS POLIGONAIS E POLIÉDRICAS

As malhas poligonais e poliédricas estão entre as mais gerais empregadas em simulação numérica. Malhas cartesianas, *corner-point*, tetraédricas, de triângulos, de Voronoi, além de outras, podem ser consideradas casos particulares de malhas poligonais ou poliédricas (MALISKA, 2004). A generalidade dessas malhas favorece sua aplicação nas mais variadas geometrias. No contexto da simulação de reservatórios de petróleo, em Maliska et al. (2012) se destaca o interesse na utilização dessas malhas para efetuar a transição entre malhas cilíndricas ao redor de poços e a malha hexaédrica usualmente utilizada para discretizar o reservatório.

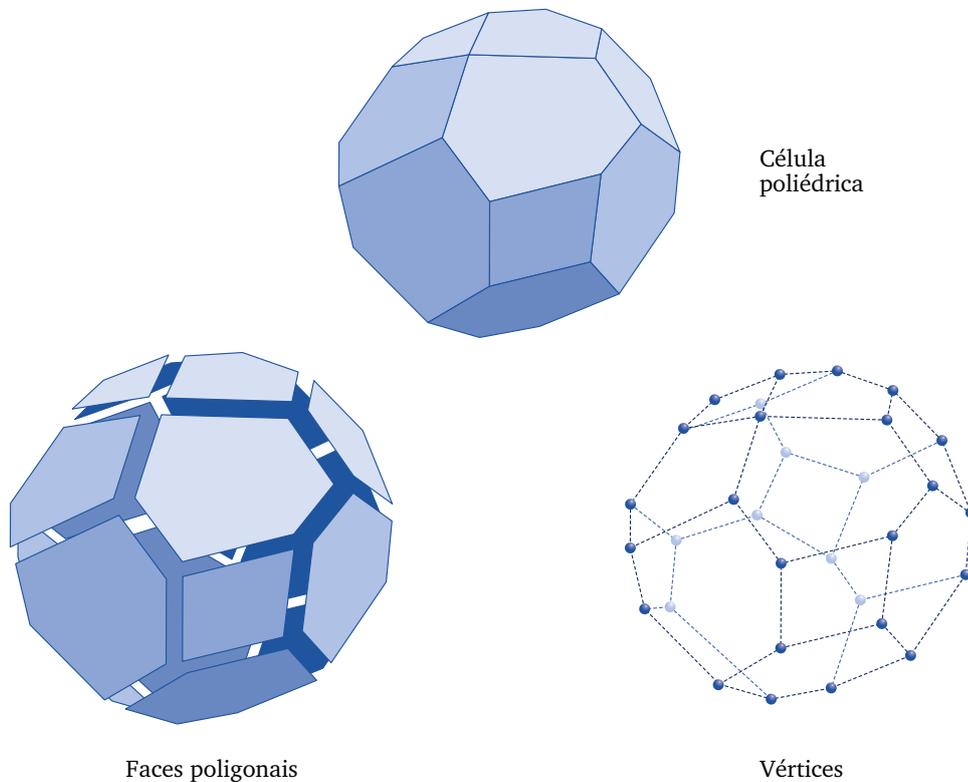


Figura 3.1: Entidades fundamentais das malhas poliédricas (adaptado de Maliska et al. (2012))

Uma malha poliédrica é uma partição finita de uma região no espaço tridimensional. Seus elementos são células poliédricas, que se caracterizam por apresentar número arbitrário de faces, as quais, por sua vez, possuem número arbitrário de vértices. Exige-se que a malha preencha completamente a região e que cada face interna – que, por definição, não está situada nas fronteiras do domínio – separe duas células.

Em Maliska et al. (2012) as malhas poliédricas são representadas com base em quatro entidades fundamentais: os vértices, as faces, as células e as fronteiras. Os vértices são simplesmente pontos no espaço, definidos a partir de suas coordenadas em relação a um

sistema de referência. Uma coleção de vértices, preferencialmente coplanares, dá origem a uma face. Por sua vez, as faces que delimitam regiões no espaço originam as células, ao passo que as situadas nas fronteiras do domínio originam representações discretas das fronteiras. Na figura 3.1 estão ilustradas três dessas entidades fundamentais.

As malhas poligonais são definidas convenientemente como uma redução das malhas poliédricas a duas dimensões. De acordo com a figura 3.2, as células poliédricas se degeneram em polígonos, ao passo que as faces, em arestas. Essa definição é utilizada na biblioteca de gerenciamento de malhas denominada *Lib-Grid*, descrita em detalhes em Maliska et al. (2012) e empregada nesse trabalho. A generalidade introduzida por essa definição permite que os códigos que fazem uso dessa biblioteca sejam genéricos, praticamente não havendo necessidade de diferenciar os casos bidimensionais dos tridimensionais.

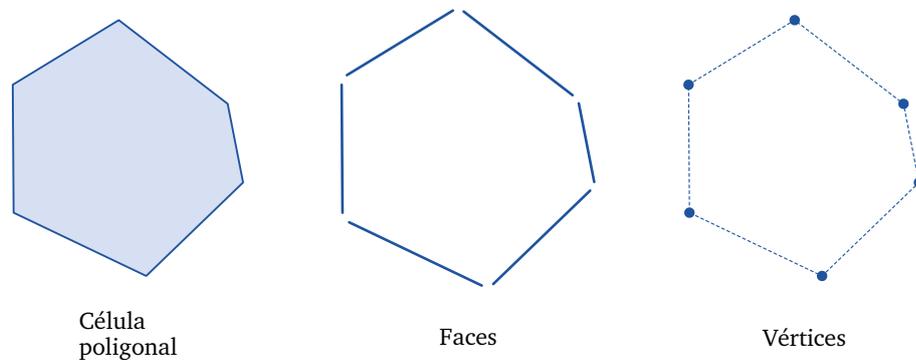


Figura 3.2: Entidades fundamentais das malhas poligonais

Reitera-se aqui que os esquemas numéricos adotados são do tipo *cell center*. Os volumes de controle coincidem exatamente com os elementos da malha, ou células, e as variáveis de interesse estão associadas aos seus centroides. Para se obter o valor dessas variáveis em algum outro local, como nas faces ou nos vértices, faz-se necessário aplicar um esquema de interpolação. Na figura 3.3 está representado um volume de controle genérico p em duas dimensões, destacando-se seus vizinhos e o local de armazenamento das variáveis.

Tanto as malhas poligonais quanto as poliédricas são não-estruturadas, pois seus elementos não seguem uma lei de construção, nem apresentam, *a priori*, o mesmo número de vizinhos. Logo, não há a princípio uma regra para a indexação dos volumes de controle e, como consequência, a estrutura das matrizes provenientes da discretização tende a ser aleatória, fato que pode prejudicar a obtenção das soluções dos sistemas lineares (MALISKA, 2004). Isso, porém, vem a ser benéfico para o trabalho, pois permite uma avaliação do multigrid e dos *solvers* nele empregados em um cenário mais exigente.

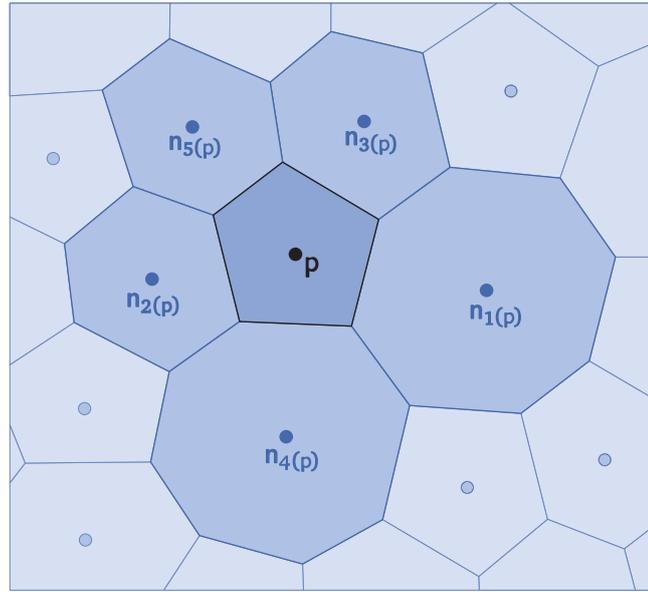


Figura 3.3: Volume de controle genérico p em esquema *cell-center* (adaptado de Maliska et al. (2012))

3.2 INTEGRAÇÃO DA EQUAÇÃO E APROXIMAÇÃO DOS FLUXOS

No método de volumes finitos há basicamente duas formas de se obter as equações discretizadas: realizando-se balanços nos volumes de controle ou neles integrando as equações diferenciais. Na realidade, ambas as formas são equivalentes, pois para se deduzir as equações diferenciais de conservação são aplicados balanços num volume de controle finito e, em seguida, faz-se suas dimensões tenderem a zero. Integrar as equações diferenciais significa, então, retroceder um passo nessa dedução, de modo a obter justamente as expressões do balanço aplicado ao volume. Como já se dispõe das equações diferenciais, é mais conveniente empregar o método da integração.

A discretização descrita abaixo da equação (2.6) é idêntica a realizada em Maliska et al. (2012). Por conveniência, essa discretização será efetuada apenas para as malhas poligonais. A generalização para as poliédricas é trivial e está disponível em Maliska et al. (2012). Considere então um volume poligonal p de N faces (as quais, na realidade, são as arestas do polígono). Integrando-se o termo difusivo e aplicando o teorema da divergência,

$$\int_{V_p} \nabla \cdot \left(\frac{\mathbf{K}}{\mu} \nabla \Phi \right) dV = \int_{A_p} \frac{\mathbf{K}}{\mu} \nabla \Phi \cdot \hat{n} dA. \quad (3.1)$$

Assumindo-se que a viscosidade é contante ao longo do domínio e aproximando-se tanto a permeabilidade absoluta \mathbf{K} quanto o gradiente médios nas faces pelos seus valores no cen-

troide do volume, segue que

$$\int_{A_p} \frac{\mathbf{K}}{\mu} \nabla \Phi \cdot \hat{n} \, dA \approx \sum_{f=1}^N \frac{\mathbf{K}}{\mu} \nabla \Phi_p \cdot \vec{A}_f. \quad (3.2)$$

Considere uma face f de p e um novo sistema de coordenadas, não ortogonal, dado pelos eixos ξ e η , conforme ilustrado na figura 3.4. O eixo ξ é colinear ao vetor que liga o centroide da célula ao da face, ao passo que η é paralelo ao eixo y . Como o fluxo de massa é inversamente proporcional à viscosidade, a qual, por sua vez, é por hipótese constante, assume-se para simplificar a notação que $\mu = 1$. Se $\mu \neq 1$, basta ao final multiplicar a expressão obtida para o fluxo por $1/\mu$. Assim, o fluxo de q_f na face f é

$$q_f \approx -(K \nabla \Phi)_p \cdot \vec{A}_f = - \begin{pmatrix} A_x & A_y \end{pmatrix} \begin{pmatrix} K_{xx} & K_{xy} \\ K_{xy} & K_{yy} \end{pmatrix}_p \begin{pmatrix} \partial_x \Phi \\ \partial_y \Phi \end{pmatrix}_p. \quad (3.3)$$

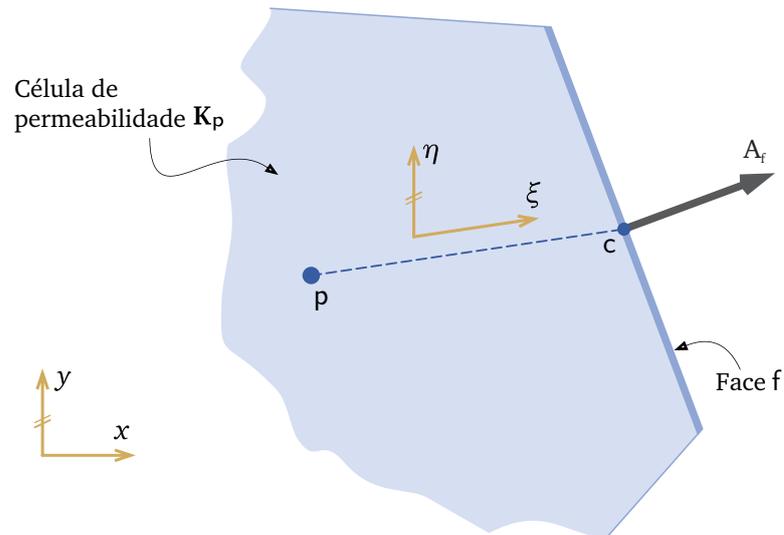


Figura 3.4: Novo sistema de coordenadas associado à face f (adaptado de Maliska et al. (2012))

Alterando o sistema de coordenadas para o dado por ξ e η , tem-se que

$$\begin{pmatrix} \partial_x \Phi \\ \partial_y \Phi \end{pmatrix}_p = \begin{pmatrix} \partial_x \xi & \partial_x \eta \\ \partial_y \xi & \partial_y \eta \end{pmatrix}_p \begin{pmatrix} \partial_\xi \Phi \\ \partial_\eta \Phi \end{pmatrix}_p. \quad (3.4)$$

De acordo com Maliska et al. (2012), os termos do jacobiano dessa transformação são dados

por

$$\begin{cases} (\partial_x \xi)_p = \frac{\Delta \xi_{p,c}}{x_c - x_p}; & (\partial_y \xi)_p = 0; \\ (\partial_x \eta)_p = -\frac{y_c - y_p}{x_c - x_p}; & (\partial_y \eta)_p = 1. \end{cases} \quad (3.5)$$

Definindo-se $\sigma := \vec{A}_f \cdot \mathbf{K}$, a expressão do fluxo resulta em

$$q_f \approx - \left[(\sigma_x)_p \frac{\Delta \xi_{p,c}}{x_c - x_p} \right] (\partial_\xi \Phi)_p - \left[-(\sigma_x)_p \frac{y_c - y_p}{x_c - x_p} + (\sigma_y)_p \right] (\partial_\eta \Phi)_p \quad (3.6)$$

Para se determinar o fluxo de massa na face f deve-se então conhecer os termos $(\partial_\xi \Phi)_p$ e $(\partial_\eta \Phi)_p$. Como ξ é colinear ao vetor que liga o centroide do volume ao da face, o primeiro termo pode ser aproximado segundo a expressão

$$(\partial_\xi \Phi)_p \approx \frac{\Phi_c - \Phi_p}{\Delta \xi_{p,c}}, \quad (3.7)$$

em que Φ_c é a função potencial avaliada no centroide da face f . Pode-se assim aproximar o primeiro termo da equação (3.6) como

$$\left[(\sigma_x)_p \frac{\Delta \xi_{p,c}}{x_c - x_p} \right] (\partial_\xi \Phi)_p \approx \left[\frac{(\sigma_x)_p}{x_c - x_p} \right] (\Phi_c - \Phi_p). \quad (3.8)$$

A derivada $(\partial_\eta \Phi)_p$, por sua vez, é igual a $(\partial_y \Phi)_p$ e provém da reconstrução do gradiente, assunto da próxima seção.

Para simplificar a notação, são definidas aqui, da mesma maneira que em Maliska et al. (2012), as variáveis

$$\begin{aligned} v_{f,p} &:= \frac{x_c - x_p}{(\sigma_x)_p} \\ \Gamma_{f,p} &:= - \left[-(\sigma_x)_p \frac{y_c - y_p}{x_c - x_p} + (\sigma_y)_p \right] (\partial_y \Phi)_p. \end{aligned} \quad (3.9)$$

Pode-se assim reescrever a equação do fluxo como

$$q_f \approx \frac{\Phi_p - \Phi_c}{v_{f,p}} + \Gamma_{f,p}. \quad (3.10)$$

Considere agora o volume q , vizinho de p , ilustrado na figura 3.5. Pode-se naturalmente aplicar para ele o mesmo procedimento de aproximação do fluxo. Isso resulta numa expressão do tipo

$$q_f \approx \frac{\Phi_q - \Phi_c}{v_{f,q}} + \Gamma_{f,q}. \quad (3.11)$$

A fim de garantir a conservação, os fluxos avaliados nas equações (3.10) e (3.11) devem ser

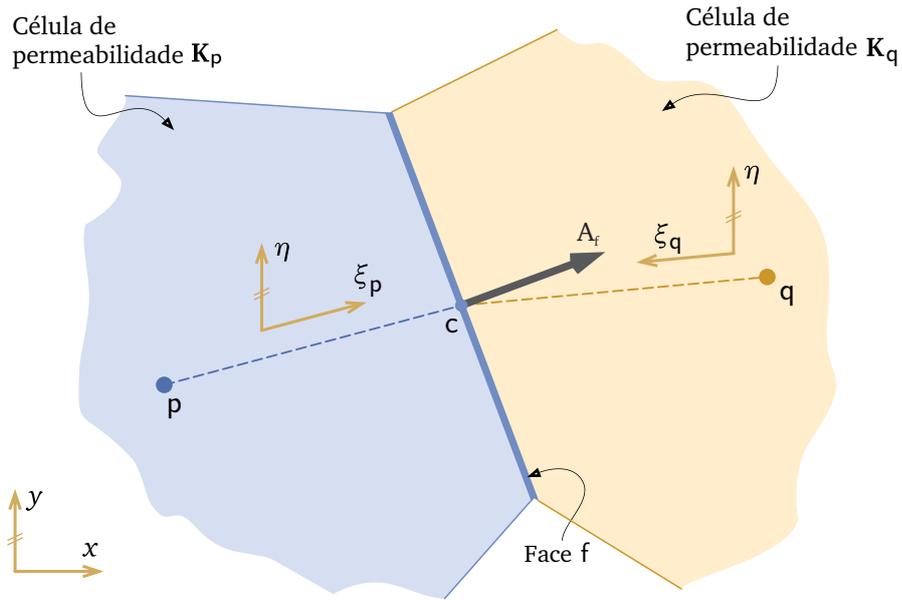


Figura 3.5: Células vizinhas p e q (adaptado de Maliska et al. (2012))

os mesmos. Pode-se isolar a variável Φ_c e eliminá-la das equações. Isso resulta na expressão

$$q_f \approx \frac{\Phi_p - \Phi_q}{\nu_{f,p} - \nu_{f,q}} + \frac{\nu_{f,p}\Gamma_{f,p} - \nu_{f,q}\Gamma_{f,q}}{\nu_{f,p} - \nu_{f,q}}. \quad (3.12)$$

Resta apenas determinar $\Gamma_{f,p}$ e $\Gamma_{f,q}$, ou, mais especificamente, $(\partial_y \Phi)_p$ e $(\partial_y \Phi)_q$.

3.3 RECONSTRUÇÃO DO GRADIENTE

A reconstrução do gradiente consiste na obtenção numérica do vetor gradiente em cada célula a partir dos valores discretos de um campo escalar. Há diversos métodos disponíveis na literatura, mas, como seu estudo não faz parte dos objetivos principais desse trabalho, será aqui empregado apenas um: a reconstrução baseada em mínimos quadrados.

Considere um volume de controle p , como o ilustrado na figura 3.6. A função potencial num vizinho n_k pode ser aproximado segundo a série de Taylor truncada, como

$$\Phi_{n_k} \approx \Phi_p + (\nabla \Phi)_p \cdot \Delta \mathbf{r}_{p,n_k}. \quad (3.13)$$

Rearranjando a expressão acima e aplicando-a a todos os vizinhos obtém-se um sistema linear sobre-determinado cujas incógnitas são as componentes do gradiente em p . Em duas

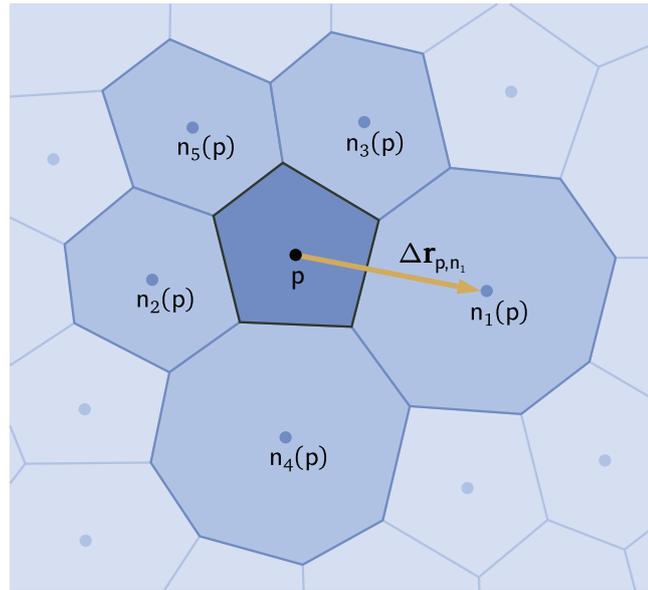


Figura 3.6: Volume de controle p e seus vizinhos (adaptado de Maliska et al. (2012))

dimensões esse sistema linear é dado por

$$\begin{pmatrix} \Delta x_{n_1} & \Delta y_{n_1} \\ \Delta x_{n_2} & \Delta y_{n_2} \\ \vdots & \vdots \\ \Delta x_{n_N} & \Delta y_{n_N} \end{pmatrix}_p \begin{pmatrix} \partial_x \Phi \\ \partial_y \Phi \end{pmatrix}_p = \begin{pmatrix} \Phi_{n_1} - \Phi_p \\ \Phi_{n_2} - \Phi_p \\ \vdots \\ \Phi_{n_N} - \Phi_p \end{pmatrix}. \quad (3.14)$$

Difícilmente o sistema linear da equação (3.14) terá solução, pois há mais equações do que incógnitas. Porém, definindo-se G_p como a matriz da equação (3.14) e b , o termo independente, pode-se tomar a solução que minimiza o resíduo, ou seja, tomar a solução tal que

$$x = \min_{\tilde{x} \in \mathbb{R}^2} \|b - G_p \tilde{x}\|_2. \quad (3.15)$$

De acordo com Strang (1988), a solução da equação acima é justamente a solução das equações normais, dadas por

$$G_p^T G_p x = G_p^T b, \quad (3.16)$$

Supondo-se que a matriz A tenha posto completo, então a solução é

$$x = G_p^\dagger b, \quad (3.17)$$

em que $G_p^\dagger = (G_p^T G_p)^{-1} G_p^T$ é a pseudo inversa de G_p (ZAMBALDI, 2009).

No sistema linear da equação (3.16) a importância relativa de cada linha de G_p , asso-

ciada a um vizinho de p , é a mesma. Porém, é razoável considerar que os vizinhos mais próximos de p tenham maior influência sobre o resultado. Para incluir esse aspecto pode-se multiplicar ambos os membros do sistema linear original por uma matriz de ponderação (STRANG, 1988), de sorte que o novo sistema linear fica

$$WG_p x = Wb, \quad (3.18)$$

sendo W uma matriz diagonal na forma

$$W = \begin{pmatrix} w_1 & & & \\ & w_2 & & \\ & & \ddots & \\ & & & w_N \end{pmatrix}. \quad (3.19)$$

No projeto MaTra, no qual o simulador aqui utilizado foi desenvolvido, obtiveram-se bons resultados utilizando como ponderação o quadrado do inverso da distância entre os centroides dos volumes, ou seja,

$$w_k = \frac{1}{\|\Delta \mathbf{r}_{p,n_k}\|_2^2}. \quad (3.20)$$

Conclui-se, assim, que de fato é possível reconstruir o campo de gradientes aproximado a partir do campo escalar da variável. Todavia, o campo de potencial é justamente a solução do problema, sendo, portanto, ainda desconhecido. Serão aqui avaliadas duas estratégias de solução desse aspecto: uma dita semi-implícita e outra, totalmente implícita.

3.3.1 ESQUEMA SEMI-IMPLÍCITO

No esquema semi-implícito, os gradientes necessários para a avaliação dos fluxos são calculado de maneira explícita. Nele é inicialmente estimado o campo da variável Φ , permitindo assim, por meio do método de reconstrução descrito anteriormente, o cálculo do campo de gradientes. Embora esse campo não satisfaça a equação diferencial, pode-se utilizá-lo para calcular uma nova solução aproximada para a variável Φ . Com base nessa nova solução calcula-se novamente o campo de gradientes. Isso acaba, pois, por definir um ciclo iterativo que, se convergir, seu resultado deve ser a solução do problema.

No esquema semi-implícito, o termo $\Gamma_{f,p}$, definido na equação (3.9), é um termo conhe-

cido em cada iteração. Dessa forma, a equação (3.12) pode ser escrita como

$$q_f \approx \frac{\Phi_p - \Phi_q}{v_p - v_q} + \theta^*, \quad (3.21)$$

em que

$$\theta^* := \frac{v_{f,p}\Gamma_{f,p} - v_{f,q}\Gamma_{f,q}}{v_{f,p} - v_{f,q}} \quad (3.22)$$

é um termo com valor numérico conhecido e, portanto, pode ser adicionado ao termo independente do sistema linear.

Da equação (3.21), caso o termo $1/(v_p - v_q)$ seja de fato positivo, tem-se que a matriz do sistema linear resultante é simétrica e diagonal dominante. Em particular, da condição de diagonal dominância segue que ela é também positiva definida (ZAMBALDI, 2009), condição que, junto a de simetria, viabiliza o emprego do método do Gradiente Conjugado, descrito na subseção 5.2.3. Além disso, a condição de diagonal dominância garante teoricamente a convergência dos métodos de Jacobi e Gauss-Seidel, descritos nas subseções 5.1.1 e 5.1.2, respectivamente. Portanto, espera-se que os sistemas lineares do esquema semi-implícito sejam mais fáceis de se resolver em relação aos do esquema totalmente implícito, descrito a seguir.

3.3.2 ESQUEMA TOTALMENTE IMPLÍCITO

Seja

$$G_p^\dagger = \begin{pmatrix} \mathbf{g}_{p,n_1}^x & \mathbf{g}_{p,n_2}^x & \cdots & \mathbf{g}_{p,n_N}^x \\ \mathbf{g}_{p,n_1}^y & \mathbf{g}_{p,n_2}^y & \cdots & \mathbf{g}_{p,n_N}^y \end{pmatrix} \quad (3.23)$$

a matriz pseudo inversa do método de mínimos quadrados relativa ao volume p . O gradiente nesse volume é calculado por meio da expressão

$$(\nabla\Phi)_p = G_p^\dagger \Delta\Phi_p, \quad (3.24)$$

em que $\Delta\Phi$ denota o vetor de diferenças

$$\Delta\Phi = \begin{pmatrix} \Phi_{n_1} - \Phi_p \\ \Phi_{n_2} - \Phi_p \\ \vdots \\ \Phi_{n_N} - \Phi_p \end{pmatrix}. \quad (3.25)$$

O esquema totalmente implícito surge da aplicação da equação (3.24) na equação (3.12), originando uma expressão da forma

$$q_f \approx \left(\frac{1 - \nu_p \gamma_p^y \sum_{k=1}^{N_p} g_{p,n_k}^y}{\nu_p - \nu_q} \right) \Phi_p - \left(\frac{1 - \nu_q \gamma_q^y \sum_{k=1}^{N_q} g_{q,n_k}^y}{\nu_p - \nu_q} \right) \Phi_q + \frac{\nu_p \sum_{n=1}^{N_p} g_{p,n_k}^y \Phi_{p,n_k} - \nu_q \sum_{n=1}^{N_q} g_{q,n_k}^y \Phi_{q,n_k}}{\nu_p - \nu_q}. \quad (3.26)$$

A equação (3.26), além de envolver as células p e q , também abrange os vizinhos desses volumes. Como consequência, o método totalmente implícito possui estêncil com dois níveis de vizinhança, resultando num aumento considerável da quantidade de termos não-nulos na matriz do sistema linear. Em malhas bidimensionais com elementos retangulares, por exemplo, o número de termos não nulos numa mesma linha passa de cinco, no esquema semi-implícito, para 13. Já em malhas de elementos hexaédricos esse número salta de sete para 25.

É interessante notar que, embora da equação (3.26) resulte uma matriz simétrica, não há garantia de que ela seja diagonal dominante, o que pode trazer dificuldades para a convergência dos métodos de Gauss-Seidel e do Gradiente Conjugado. De fato, há de se observar que o termo que multiplica Φ_p é distinto daquele que multiplica Φ_q . Além disso, os coeficientes que multiplicam os termos Φ_{p,n_k} e Φ_{q,n_k} não contribuem para aumentar o coeficiente da diagonal da matriz.

3.4 OBSERVAÇÕES

Na seção 3.2 deduziu-se um método numérico de cálculo dos fluxos baseado na transformação do sistema de coordenadas. Escolheu-se um novo sistema de coordenadas em que ξ é colinear com o vetor que liga o centroide do volume ao centroide da face e η é paralelo a y . Isso é, pois, equivalente a transformar apenas um dos eixos do sistema de coordenadas: nesse caso, o x . Porém, o mesmo procedimento poderia ter sido feito com y . Definir-se-ia o eixo ξ paralelo ao x e o η , colinear ao referido vetor que liga os dois centroides. Há situações em que a escolha de um dos eixos para a transformação é preferível à do outro, como na-

que em que um deles induz a um coeficiente ν_p nulo. Atualmente se estudam no projeto MaTra, do qual o simulador desse trabalho provém, diversos algoritmos para a escolha do eixo. O que tem se mostrado mais promissor, e que será utilizado nesse trabalho, é aquele que busca maximizar o coeficiente $1/(\nu_p - \nu_q)$.

Também é interessante fazer alguns comentários do porquê avaliar dois esquemas de aproximação de fluxo: semi implícito e totalmente implícito. O esquema semi-implícito resulta numa matriz diagonal dominante, muito comum em problemas puramente difusivos, facilitando a obtenção da solução do sistema linear. Isso, somado ao fato de que o laço iterativo adicional deve tornar o esquema semi-implícito mais lento que o totalmente implícito, faz com que esse esquema seja interessante para a avaliação do desempenho computacional do multigrid. Já o esquema totalmente implícito, por apresentar sistema linear teoricamente mais difícil de resolver (a matriz é menos esparsa e talvez não diagonal dominante), pode ser empregado para se verificar a robustez do método.

4 O MÉTODO MULTIGRID DE CORREÇÕES ADITIVAS

4.1 INTRODUÇÃO

O capítulo anterior terminou com a caracterização do sistema linear que deve ser resolvido para a obtenção da solução numérica do problema. A solução de sistemas lineares consiste numa das etapas fundamentais da simulação numérica. Segundo Maliska (2004), dispense-se cerca de 60% a 70% do tempo total de simulação apenas resolvendo-se sistemas lineares. Considerando-se que a dimensão dos sistemas lineares dos problemas reais de engenharia chegam a ultrapassar a ordem de um milhão, o tempo consumido pode ser consideravelmente alto. De fato, há simulações que demandam semanas, ou mesmo meses, para serem concluídas. Assim, qualquer melhora no processo de solução de sistemas lineares, mesmo que pequena, pode resultar numa economia de tempo apreciavelmente grande.

Atualmente, os métodos multigrid estão entre os métodos mais eficientes disponíveis. Sua ideia básica é decompor o sistema linear original num conjunto de novos sistemas, cada um associado a uma malha progressivamente mais grosseira. Todos esses sistemas são então resolvidos simultaneamente, havendo troca de informação entre eles. Segundo Demmel (1996), há dois aspectos fundamentais, ambos dependentes da malha, que justificam a melhora no processo de solução: a velocidade com que a informação se difunde no domínio e a taxa com que cada componente do erro, decomposto numa série de Fourier discreta, é eliminada. A seguir são apresentados dois exemplos que exploram cada um desses aspectos. O primeiro deles é adaptado de Demmel (1996) e o segundo, de Briggs et al. (2000).

Considere uma placa plana, retangular, que possui temperatura prescrita no contorno e, inicialmente, todo o seu domínio se encontra a T_0 . Subitamente a temperatura em seu ponto central passa para T_1 , temperatura superior a T_0 . Deseja-se aqui calcular qual é o novo campo de temperatura da placa. Suponha que a solução é calculada numericamente utilizando-se uma malha cartesiana uniforme $(n+1) \times (n+1)$, sendo n um número par. Sendo constantes as propriedades do meio, a equação discreta para um volume certo volume P é, utilizando a notação de Maliska (2004),

$$T_P = \frac{T_E + T_W + T_N + T_S}{4}, \quad (4.1)$$

Suponha-se que o método de Jacobi, descrito no próximo capítulo, seja empregado para o cálculo da solução do sistema linear. Com base na equação (4.1), verifica-se que o valor de

temperatura em P na iteração corrente é simplesmente a média das temperaturas de seus vizinhos na iteração anterior. Como consequência, na primeira iteração apenas a temperatura nos volumes vizinhos ao central, onde a temperatura é T_1 , será alterada. Para que todos os volumes sofram variação de temperatura serão necessárias da ordem de $n/2$ iterações. Isso, pois, indica que a velocidade de difusão da informação na malha depende do número de volumes presentes. Em malhas refinadas, além de serem necessárias mais iterações para a solução do problema, o custo de cada iteração é maior, pois devem ser varridos todos os seus elementos.

Nas figuras 4.1 e 4.2 estão ilustradas soluções parciais de duas malhas empregadas na solução do problema descrito acima: uma delas é 11×11 e a outra, 51×51 . Na malha 11×11 , em cada iteração são percorridos 121 volumes e, após 50 iterações, já praticamente se atingiu a convergência. Na malha 51×51 , entretanto, em cada iteração se faz necessário varrer 2601 volumes e, após 50 iterações, a solução parcial ainda está muito distante da solução do problema. Fica, pois, bastante claro o crescimento acentuado do custo computacional resultante do refinamento da malha.

Embora no exemplo anterior se tenha utilizado o método de Jacobi, pode-se provar que, para qualquer método de solução baseado em médias, o número de iterações mínimo necessário para a propagação da informação do centro para as fronteiras é da ordem de $\log(n/2)$ (DEMMEL, 1996). Nisso se incluem métodos como o de Gauss-Seidel, SOR e aqueles baseados no espaço de Krylov, como o GMRES e o Gradiente Conjugado. Já o multigrid, por resolver simultaneamente o problema em malhas de diversos graus de refino, permite propagação muito mais eficiente. As malhas grosseiras se prestam ao aumento da velocidade de propagação da informação ao longo do domínio, ao passo que nas mais refinadas são obtidas soluções de melhor qualidade.

O outro aspecto fundamental do multigrid é a relação entre a malha e a taxa de eliminação das componentes de diferentes frequências do erro. Considerando-se novamente um problema de condução de calor, mas agora unidimensional. Na fronteira esquerda se prescreve temperatura $T_L = 1$ e na direita, $T_R = 0$. As propriedades são supostas constantes e homogêneas. A malha é uniforme, possui N elementos e, nas fronteiras, há meios-volumes de controle, conforme figura 4.3. Há interesse apenas na solução de regime permanente.

A discretização da equação diferencial da energia dá origem a um sistema linear $Ax = b$

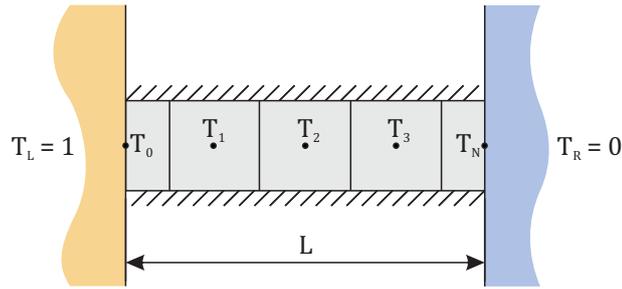


Figura 4.3: Problema de condução de calor unidimensional

forma

$$R_J = \Omega \Lambda \Omega^T, \quad (4.6)$$

em que Ω é a matriz dos autovetores, normalizados, e Λ , matriz diagonal que contém os autovalores. Aplicando essa fatoração na equação (4.4) resulta

$$e^{(k)} = (\Omega \Lambda \Omega^T)^k e^{(0)} = \Omega \Lambda^k \Omega^T e^{(0)} = \sum_{i=1}^N [\lambda_i^k \text{proj}_{\omega_i} (e^{(0)})]. \quad (4.7)$$

Isso significa que o erro na iteração k é igual ao somatório da projeção de $e^{(0)}$ em cada autovetor multiplicado por seu correspondente autovalor elevado a k .

A equação (4.7) permite inferir que a convergência do método de Jacobi é garantida se todos os autovalores da matriz de iteração são inferiores à unidade. Além disso, a convergência será tão mais rápida quanto mais próximo estiver o erro inicial do autovetor associado ao menor autovalor.

Segundo Briggs et al. (2000), os autovalores e autovetores da matriz de iteração do presente problema são

$$\lambda_k = 1 - 2 \sin^2 \left[\frac{k\pi}{2(n+1)} \right], \quad 1 \leq k \leq N, \quad (4.8)$$

$$\omega_k(j) = \sin \left[\left(\frac{k\pi}{n+1} \right) j \right], \quad \begin{array}{l} 1 \leq j \leq n \\ 1 \leq k \leq N. \end{array} \quad (4.9)$$

Observa-se que os autovetores são basicamente termos discretos da série de Fourier, baseada em senos, definida no período de 0 a $\pi/(n+1)$. É interessante notar que há um erro inicial ótimo para a convergência do método. Supondo sem perda de generalidade que N é ímpar, se $e^{(0)}$ for colinear a $\omega_{(n+1)/2}$, então sua projeção nos outros autovetores será nula e $\lambda_{(k+1)/2} = 0$. De acordo com equação (4.7), o método deve convergir na primeira iteração. Todavia, caso $e^{(0)}$ esteja próximo de ω_1 ou de ω_N , $\lambda \approx 1$ e a convergência do método será consideravelmente lenta.

As observações do parágrafo anterior são de suma importância para os métodos multigrid. Basicamente elas indicam que, dado um erro de certa frequência, há uma malha ótima para eliminá-lo. Além disso, cada malha acaba por definir uma faixa de frequências considerada alta – se $k > n/2$ – e outra, baixa: se $k < n/2$. O refinamento de uma malha faz com que as frequências do erro, suposto constante, sejam cada vez mais baixos em relação à malha, levando a convergência ficar cada vez mais lenta. O uso de diferentes graus de refino de malha no multigrid acaba, pois, por favorecer que cada frequência do erro seja eliminado numa malha mais adequada. As malhas mais refinadas eliminam de maneira mais eficiente as componentes de maior frequência, enquanto que as mais grosseiras, as de menor.

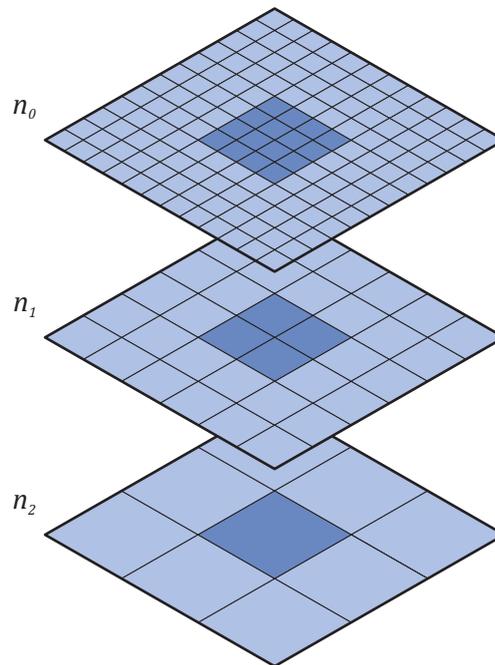


Figura 4.4: Exemplo de diferentes níveis de malhas utilizadas no multigrid

Na figura 4.4 há um exemplo de três malhas utilizadas no multigrid. Cada volume da malha do nível n_1 resultou da aglomeração de quatro volumes do nível n_0 . Do mesmo modo, cada volume do nível n_2 provém de quatro volumes do nível n_1 . Um único volume da malha n_2 está, portanto, associado a 16 volumes da malha mais refinada. Assim, se por exemplo o primeiro problema-exemplo dessa seção fosse resolvido usando o multigrid, já na primeira iteração todos os elementos da malha do nível n_2 sofreriam influência da temperatura no centro. Essa influência seria, por sua vez, repassada aos níveis mais finos, de modo que todos os volumes da malha fina também acabariam sofrendo essa influência.

4.2 MULTIGRID GEOMÉTRICO E MULTIGRID ALGÉBRICO

A definição de um método multigrid requer tanto a especificação do modo como a informação será transmitida entre as malhas, ou, mais precisamente, a estrutura dos operadores de interpolação e restrição, quanto do algoritmo de obtenção de malhas mais grosseiras. Usualmente os métodos multigrid são classificados em duas categorias: geométricos e algébricos.

Os multigrids geométricos foram os primeiros a serem desenvolvidos, há cerca de 40 anos (STÜBEN, 2001). Neles a técnica de obtenção das malhas mais grosseiras é fixa e depende fundamentalmente de aspectos geométricos e topológicos da malha, sem considerar a física do problema. Em malhas estruturadas, ela pode ser relativamente simples, pois cada elemento já possui *a priori* um número definido de vizinhos. Isso, entretanto, não ocorre em malhas não-estruturadas. Nesses casos pode ser realmente desafiador conceber um algoritmo suficientemente genérico para obter malhas grossas adequadas. Essa dificuldade de obter generalidade também se verifica nos operadores de interpolação, que, nos multigrids geométricos, são definidos normalmente em função das posições relativas entre centroides dos volumes da malha fina e os da grossa.

Os multigrids algébricos foram desenvolvidos visando-se justamente a prover flexibilidade às técnicas multigrid. Eles se basearam na constatação de que é possível estabelecer técnicas eficazes de obtenção de malhas grossas e operadores de interpolação e restrição apenas analisando-se a matriz do sistema linear, independentemente da malha empregada (STÜBEN, 2001). Por operar apenas na matriz do problema, a obtenção das malhas mais grosseiras fica muito mais simples, não havendo, pois, as dificuldades oriundas da natureza da malha. Na realidade, as malhas ficam praticamente desacopladas do problema, razão pela qual o multigrid algébrico pode ser utilizado para a solução de qualquer sistema linear, mesmo que ele não seja proveniente de algum processo de discretização. Por esse motivo, o multigrid algébrico também tem sido denominado de *multilevel* (STÜBEN, 2001). Como esse trabalho está sendo publicado no contexto da engenharia, em que o termo *multilevel* praticamente não é conhecido, o termo multigrid será mantido.

Uma vez que atualmente o interesse no método multigrid algébrico é muito maior que o geométrico, o multigrid a ser estudado se enquadra na modalidade algébrica. A metodologia aqui empregada é a ACM: *Additive Correction Multigrid* (HUTCHINSON; RAITHBY, 1986). Caso os resultados do presente trabalho se mostrem positivos, pode-se aperfeiçoar a biblioteca ACMLib, biblioteca desenvolvida pelo SINMEC.

4.3 OPERADORES DE INTERPOLAÇÃO E RESTRIÇÃO

Suponha-se, sem perda de generalidade, que dois níveis de malha sejam utilizados. O mais fino é denotado por h , enquanto que o mais grosseiro, por H . Caso mais de dois níveis estejam envolvidos, basta aplicar o mesmo raciocínio recursivamente. Considera-se ainda que a malha do nível H foi obtida aglomerando-se os volumes da malha h . O sistema linear que se quer resolver é denotado por

$$A_h x^h = b^h. \quad (4.10)$$

O operador de interpolação I_h^H é um operador linear, de posto completo, que mapeia h em H (STÜBEN, 2001). Já o operador de restrição, denotado por I_H^h , é basicamente o operador de interpolação transposto. A matriz do nível H é definida como

$$A_H = I_h^H A_h I_H^h, \quad \text{sendo } I_H^h = (I_h^H)^T. \quad (4.11)$$

Seja x_k^h uma solução aproximada, obtida na iteração k , da equação (4.10) e x_e^h a solução exata. De acordo com a equação (4.3), o erro e^h é dado por

$$e^h = x_k^h - x_e^h. \quad (4.12)$$

Aplicando a equação anterior na equação (4.10) obtém-se um sistema linear equivalente:

$$A_h e^h = r^h, \quad (4.13)$$

em que r^h é o resíduo do nível h , definido por

$$r^h = b^h - A_h x_a^h. \quad (4.14)$$

Pré-multiplicando a equação (4.13) pelo operador de interpolação,

$$I_h^H A_h e^h = I_h^H r^h. \quad (4.15)$$

Sendo $r^H = I_h^H r^h$ e $e^H = (I_H^h)^T e^h$ o resíduo e o erro, respectivamente, associados ao nível H , segue que

$$I_h^H A_h I_H^h e^H = r^H, \quad (4.16)$$

ou ainda, aplicando-se a equação (4.11),

$$A_H e^H = r^H. \quad (4.17)$$

Uma vez calculado e^H , corrige-se x^h , relativo à equação (4.10), utilizando-se a expressão

$$x_{k+1}^h = x_k^h + I_H^h e^H. \quad (4.18)$$

A dedução anterior é geral, válida tanto para os métodos multigrid geométricos quanto para os algébricos (STÜBEN, 2001). O que acaba por definir o método em si é a estrutura do operador de interpolação. Esse operador depende tanto da forma com que as malhas mais grosseiras são obtidas quanto do método que relaciona os volumes da malha fina com os da grosseira. Nos métodos geométricos esse operador é fixo, dependendo fundamentalmente da geometria do problema, ao passo que nos algébricos ele é flexível e é construído em função da matriz.

No método multigrid desenvolvido nesse trabalho cada malha é construída aglomerando-se volumes da malha imediatamente mais refinada. Busca-se nessa aglomeração que a matriz da malha resultante tenha coeficientes balanceados, tão próximos quanto possível. Isso acaba por facilitar a difusão de informação na malha, melhorando o desempenho do método de solução do sistema linear (KELLER, 2007). Não serão aqui abordados os detalhes do algoritmo que realiza a aglomeração. Caso haja interesse deve-se consultar Keller (2007).

Sendo Ω_i^h um elemento de uma das malhas do multigrid e Ω_j^H o elemento da malha imediatamente mais grosseira, o operador de interpolação é aqui definido como

$$I_h^H = \{I_{ij}\} = \begin{cases} i_{ij} = 1, & \text{se } \Omega_i^h \in \Omega_j^H \\ i_{ij} = 0, & \text{se } \Omega_i^h \notin \Omega_j^H \end{cases}, \quad (4.19)$$

sendo equivalente ao empregado em Pescador Jr (2010) e ao em Keller (2007).

4.4 CICLOS MULTIGRID

O ciclo multigrid é definido pela forma com que são percorridas as malhas do multigrid. Existem os ciclos adaptativos e os fixos. Nos adaptativos faz-se necessário o cálculo de algum parâmetro, como a taxa de decaimento do resíduo, para se decidir se a malha deve ser alterada e, se for esse o caso, se se deve passar para uma mais grosseira ou para uma mais refinada. Nos fixos, por sua vez, o ciclo é definido de antemão, sendo os mais comuns o V, o F e o W. Além da forma como são percorridas as malhas, também já é definido *a priori* o número de iterações que cada *solver* realiza antes de mudar de malha. Dado os bons resulta-

dos obtidos por Pescador Jr (2010) para o ciclo F em comparação a ciclos adaptativos, serão utilizados nesse trabalho apenas ciclos fixos.

É importante salientar que o multigrid, por si só, não é suficiente para resolver um sistema linear. Nele basicamente são criados diversos sistemas lineares, cada um correspondendo a malhas progressivamente mais grosseiras. Para resolver cada um desses sistemas faz-se necessário um *solver* específico, que pode ser diferente para cada um deles.

Naturalmente não faz sentido escolher um solver direto nos níveis mais refinados, pois eles demandam memória e tempo computacional elevados. Além disso, se um *solver* direto fosse utilizado, a solução exata do sistema linear, desconsiderando-se error numéricos, já seria imediatamente obtida, tornando desnecessárias as malhas mais grosseiras. Logo, os *solvers* devem ser iterativos, excetuando-se possivelmente no nível mais grosseiro. De fato, será aqui empregado no último nível a fatoração LU. Dado que o número de elementos desse nível é usualmente pequeno, da ordem de 10, o custo computacional da fatoração também é pequeno. Adicionalmente, da seção anterior se tem que, embora o termo independente dos sistemas lineares sofra alteração, a matriz sempre permanece constante, de modo que a fatoração deve ser calculada uma vez apenas. Uma vez calculada essa fatoração, a solução do sistema linear é bastante eficiente.

Na figura 4.5 há a representação gráfica dos ciclos V, F e W, ao passo que nas listagens 4.1, 4.2 e 4.3, seus algoritmos implementados em C++ (levemente modificados para maior clareza). Esses ciclos foram implementados recursivamente via funções aqui denominadas de `executeVCycle`, `executeFCycle` e `executeWCycle`. Cada uma delas está presente numa classe denominada *Layer*, na qual estão contidos os métodos e atributos pertinentes aos níveis do multigrid.

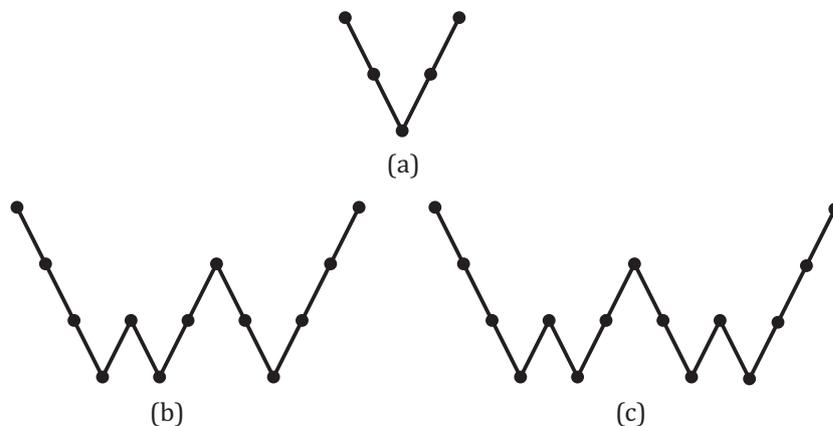


Figura 4.5: Ciclos multigrid fixos: V (a), F (b) e W (c)

Listagem 4.1: Algoritmo recursivo do ciclo V

```

1  void executeVCycle(){
2      if ( isCoarsestLayer ){
3          solver->solve();
4      }else{
5          solver->iterate();
6          restrictResidue ();
7          coarserLayer->executeVCycle();
8          interpolateError ();
9          solver->iterate();
10     }
11 }
```

Listagem 4.2: Algoritmo recursivo do ciclo F

```

1  void executeFCycle(){
2      if ( isCoarsestLayer ){
3          solver->solve();
4      }else{
5          solver->iterate();
6          restrictResidue ();
7          coarserLayer->executeFCycle();
8          interpolateError ();
9          solver->iterate();
10     if ( !isFinest ){
11         restrictResidue ();
12         coarserLayer->executeVCycle();
13         interpolateError ();
14         solver->iterate();
15     }
16 }
17 }
```

Listagem 4.3: Algoritmo recursivo do ciclo W

```

1  void executeWCycle(){
2      if( isCoarsestLayer() ){
3          solver->solve();
4      }else{
5          solver->iterate();
6          restrictResidue ();
7          coarserLayer->executeWCycle();
8          interpolateError ();
9          solver->iterate();
10         if( !isFinest ){
11             restrictResidue ();
12             coarserLayer->executeWCycle();
13             interpolateError ();
14             solver->iterate();
15         }
16     }
17 }

```

Um dos atributos da classe *Layer* é um objeto da classe *Solver*, responsável pela solução do sistema linear do nível correspondente. A declaração `solver->iterate()` indica que o objeto *solver* deve executar o número de iterações, fixo, especificado para ele. É usual definir essa operação como suavização da solução. No nível mais grosseiro, entretanto, um *solver* direto é empregado, de sorte que a solução exata já é imediatamente obtida, daí o comando `solver->solve`.

Os comandos `restrictResidue()` e `interpolateError()` fazem com que sejam aplicados os operadores de restrição e interpolação, respectivamente. Para a aplicação desses operadores são utilizadas as informações disponíveis num objeto *Layer*, de um nível imediatamente mais grosseiro, acessado via o ponteiro `coarserLayer`. Esse ponteiro também é empregado para chamar novamente as funções dos ciclos, tornando-as recursivas.

Para melhor entender os algoritmos das listagens anteriores, considere um caso em que três são os níveis de malha e que o ciclo empregado é o V. A função `executeVCycle` será chamada no nível mais fino, algumas iterações do método de solução do sistema linear serão realizadas e, então, o resíduo obtido será mapeado (restringido) para o nível seguinte.

Chama-se novamente a função `execut eVCycle`, mas agora para o segundo nível do multigrid. Como ainda não se está no nível mais grosseiro, o atributo `isCoarsestLayer` é falso, levando à execução de algumas iterações do *solver*, à restrição do resíduo desse nível e à chamada da função `execut eVCycle`. Agora se passa ao nível mais grosseiro e, por isso, seu sistema linear é resolvido exatamente, encerrando a função `execut eVCycle` para esse nível. Porém, para os níveis mais finos a função ainda não está concluída, sendo necessário executar para cada os comandos das linhas 8 e 9. A partir desses comandos é feita a interpolação do erro, definida de acordo com a equação (4.18) e, em seguida, a execução de mais algumas iterações do *solver*.

5 MÉTODOS DE SOLUÇÃO DE SISTEMAS LINEARES EMPREGADOS NO MULTIGRID

O presente capítulo é dedicado à apresentação sucinta dos métodos de solução de sistemas lineares utilizados no Multigrid. Serão aqui apresentados os métodos de Jacobi, de Gauss-Seidel, os baseados no espaço de Krylov – MinRes, Gradiente Conjugado e GMRes – e a fatoração LU, sempre utilizada no nível mais grosseiro. Embora o método de Jacobi não tenha de fato sido utilizado na obtenção de resultados, sua descrição é importante para o capítulo anterior.

5.1 MÉTODOS ITERATIVOS BÁSICOS

5.1.1 MÉTODO DE JACOBI

O método de Jacobi é um método iterativo e, portanto, da sua aplicação resulta uma sequência de soluções aproximadas que, se convergir, tem como limite a solução do sistema linear. O método de Jacobi propõe que, se $x^{(k)} \in \mathbb{R}^n$ é a solução aproximada na iteração k , a componente i , $1 \leq i \leq n$, da nova solução seja dada por

$$x_i^{(k+1)} = \left[b - \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij} x_j^{(k)} \right] a_{ii}^{-1} \quad 1 \leq i \leq n. \quad (5.1)$$

O método de Jacobi, na realidade, pertence a um conjunto de métodos baseados na decomposição da matriz A em duas outras matrizes S e T tais que (STRANG, 1988)

$$A = S - T. \quad (5.2)$$

Eles propõem que a correção da solução aproximada se dê segundo a expressão

$$x^{(k+1)} = S^{-1} T x^{(k)} + S^{-1} b. \quad (5.3)$$

Subtraindo-se a solução exata do sistema linear de ambos os membros da equação anterior e definindo-se o erro segundo a equação (4.3), a expressão anterior também é equivalente a

$$e^{(k+1)} = R e^{(k)}, \quad (5.4)$$

ou ainda

$$e^{(k+1)} = R^k e^{(0)}, \quad (5.5)$$

sendo $R = S^{-1}T$ denominado de matriz de iteração do método.

A convergência é garantida para qualquer estimativa inicial se, e somente se, o raio espectral da matriz de iteração for inferior à unidade (STRANG, 1988), ou seja,

$$\rho(R) = \max_i |\lambda_i| < 1, \quad (5.6)$$

sendo λ_i autovalor da matriz de iteração. Caso a matriz de iteração seja diagonalizável, a equação (5.4) pode ser reescrita como

$$e^{(k+1)} = \sum_{i=1}^n c_i \lambda_i^{(k)} \omega_i, \quad (5.7)$$

sendo λ_i e ω_i autovalor e autovetor de R , respectivamente. A convergência será tão mais rápida quanto maior for, em relação aos demais, o coeficiente c que multiplica o menor autovalor.

A matriz A pode ser decomposta de forma que $A = L + D + U$, em que a matriz D contém os elementos da diagonal de A ; a L , os elementos abaixo da diagonal; e a U , os acima. Propõe-se no método de Jacobi que $S = D$ e $T = -(L + D)$. Assim, a equação (5.3) fica

$$x^{(k+1)} = -D^{-1}(L + U)x^{(k)} + D^{-1}b, \quad (5.8)$$

ou ainda, em função do erro,

$$e^{(k+1)} = (-D^{-1}(L + U))^k e^{(0)}. \quad (5.9)$$

5.1.2 MÉTODO DE GAUSS-SEIDEL

O método de Gauss-Seidel é análogo ao de Jacobi. Ele diferencia-se apenas no fato de que no cálculo das componentes de $x^{(k+1)}$ são sempre empregados os valores atualizados da solução aproximada. Ou seja,

$$x_i^{(k+1)} = \left[b - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij} x_j^{(k+1)} \right] a_{ii}^{-1} \quad 1 \leq i \leq n \quad (5.10)$$

O método de Gauss-Seidel também pode ser expresso segundo a decomposição $A = S -$

T . Nesse caso, $S = L + D$ e $T = -U$. A correção da solução é, pois,

$$x^{(k+1)} = -(L+D)^{-1}Ux^{(k)} + (L+D)^{-1}b. \quad (5.11)$$

5.2 MÉTODOS DE PROJEÇÃO

Seja o sistema linear $Ax = b$, $A \in \mathbb{R}^n$, e Γ e Ξ dois subespaços de dimensão m de \mathbb{R}^n . Se x_0 é a estimativa inicial da solução, a definição geral de um método de projeção é (SAAD, 2000)

$$\text{Encontre } \bar{x} \in x_0 + \Gamma \quad \text{tal que } b - A\bar{x} \perp \Xi. \quad (5.12)$$

Os subespaços Γ e Ξ são gerais e são suficientes para definir o método de projeção. É usual, entretanto, escolher Γ e Ξ tais que $\Gamma = \Xi$ ou $\Gamma = A\Xi$. Caso $\Gamma = \Xi$, pode-se provar que o erro, definido segundo a expressão (norma A)

$$E(x_*) = [(x_* - x_e)^T A(x_* - x_e)]^{1/2}, \quad (5.13)$$

sendo x_e a solução exata do sistema linear, é minimizado em $x_0 + \Gamma$ se x_* coincidir com o vetor \bar{x} da equação (5.12) (SAAD, 2000). Em outras palavras

$$E(\bar{x}) = \min_{x \in x_0 + \Gamma} E(x). \quad (5.14)$$

Caso $\Xi = A\Gamma$ há um resultado análogo:

$$R(\bar{x}) = \min_{x_0 + \Gamma} R(x), \quad (5.15)$$

em que

$$R(x) = \|b - Ax\|_2. \quad (5.16)$$

Ou seja, a nova aproximação da solução do sistema linear é aquela que minimiza em $x_0 + \Gamma$ a norma euclideana do resíduo.

Por conveniência são aqui definidos os conjuntos $V = \{v_1, v_2, \dots, v_m\}$ e $W = \{w_1, w_2, \dots, w_m\}$ tais que

$$\Gamma = \text{span}(v_i; v_i \in V, 0 \leq i \leq m) \quad (5.17)$$

$$\Xi = \text{span}(w_i; w_i \in W, 0 \leq i \leq m) \quad (5.18)$$

5.2.1 MÉTODO MINRES

O método MINRES (Minimal Residual) pertence a categoria dos métodos de projeção unidimensionais. O único elemento da base de Γ é $v = r$, ao passo que o da de Ξ , $w = Ar$. Sendo $r_0 = b - Ax_0$ o resíduo inicial, a nova aproximação da solução é dada por

$$\bar{x} = x_0 + \alpha r_0, \quad \alpha \in \mathbb{R}. \quad (5.19)$$

O fator α provém da condição de ortogonalidade entre o novo resíduo e o subespaço Ξ . Assim, se \bar{r} é o resíduo associado a \bar{x} ,

$$\begin{aligned} \bar{r}^T Ar_0 = 0 &\longrightarrow (b - A\bar{x})^T Ar_0 = (b - A(x_0 + \alpha r_0))^T Ar_0 = (r_0 - \alpha Ar_0)^T Ar_0 = 0 \\ \longrightarrow &\alpha = \frac{r_0^T Ar_0}{r_0^T A^T Ar_0}. \end{aligned} \quad (5.20)$$

Na listagem 5.1 há o algoritmo do método MINRES, observando-se que “(,)” denota produto escalar.

Listagem 5.1: Algoritmo do MINRES

-
- 1 Até convergir faça
 - 2 $r \leftarrow b - Ax$
 - 3 $\alpha \leftarrow (Ar, r) / (Ar, Ar)$
 - 4 $x \leftarrow x + \alpha r$
 - 5 Fim – faça
-

5.2.2 MÉTODO GMRES

O método GMRES (*Generalized Minimum Residual*) é uma generalização do método MINRES. Nele tem-se novamente que $\Xi = A\Gamma$, mas agora a dimensão dos subespaços Γ e Ξ é m , sendo $1 \leq m \leq n$. O subespaço Γ é tomado como sendo o espaço de Krylov m -dimensional (SAAD, 2000), dado por

$$\kappa_m = \text{span}(r, Ar, A^2r, \dots, A^{m-1}r). \quad (5.21)$$

Infelizmente os elementos da base do subespaço de Krylov da equação (5.21) não são adequados à implementação computacional. Segundo Strang (2006a), o número de condi-

cionamento associado a essa base é alto e, além disso, dependendo da matriz A , seus elementos podem ser quase colineares, algo que pode ser extremamente danoso devido à representação aritmética finita nos computadores. Para, então, reduzir esse problema, é usual aplicar o método de Arnoldi, que consiste basicamente na ortogonalização da base do subespaço de Krylov (STRANG, 2006a). Os vetores obtidos, normalizados, irão compor a base de Γ . Na listagem 5.2 há o método de Arnoldi utilizando-se para a ortogonalização o método de Gram-Schmidt modificado (SAAD, 2000).

Listagem 5.2: Método de Arnoldi usando ortogonalização de Gram-Schmidt modificada (adaptado de Saad (2000))

-
- 1 Escolha um vetor v_1 de norma 1
 - 2 Para $j = 1, 2, \dots, m$ faça
 - 3 Calcule $w_j = Av_j$
 - 4 Para $i = 1, \dots, j$ faça
 - 5 $h_{ij} \leftarrow (w_j, v_i)$
 - 6 $w_j \leftarrow w_j - h_{ij}v_i$
 - 7 Fim – faça
 - 8 $h_{j+1,j} \leftarrow \|w_j\|_2$. Se $h_{j+1,j} = 0$, pare.
 - 9 $v_{j+1} \leftarrow w_j/h_{j+1,j}$
 - 10 Fim – faça
-

Se V_m é a matriz $n \times m$ cujas colunas são v_1, v_2, \dots, v_m e \bar{H}_m é a matriz $(m+1) \times m$ de Hessenberg cujos coeficientes não nulos h_{ij} são definidos de acordo com a listagem 5.2, é possível provar que (SAAD, 2000)

$$AV_m = V_{m+1}\bar{H}_m. \quad (5.22)$$

Com base nessa igualdade e no método de Arnoldi pode-se então deduzir o algoritmo do GMRES.

Seja x_0 a estimativa inicial da solução de $Ax = b$ e $r_0 = b - Ax_0$, o resíduo associado, e $\beta = \|r_0\|_2$. Seja m é a dimensão dos espaços Γ e Ξ , sendo a base de Γ os vetores v_1, v_2, \dots, v_m definidos a partir do método de Arnoldi. A nova aproximação da solução é dada por

$$\bar{x} = x_0 + V_m y. \quad (5.23)$$

Aplicando-se \bar{x} na função $R(x)$, definida na equação (5.16), segue que

$$\begin{aligned}
 R(\bar{x}) &= \|b - A\bar{x}\|_2 \\
 &= \|b - A(x_0 + V_m y)\|_2 \\
 &= \|r_0 - AV_m y\|_2 \\
 &= \|\beta v_1 - V_{m+1} H_m y\|_2 \\
 &= \|V_{m+1}(\beta e_1 - H_m y)\|_2 \\
 &= \|\beta e_1 - H_m y\|_2,
 \end{aligned} \tag{5.24}$$

observando-se que a última igualdade provém do fato de que V_{m+1} é ortogonal. Como y minimiza R , y deve ser a solução de mínimos quadrados do sistema linear

$$H_m y = \beta e_1, \tag{5.25}$$

cuja ordem é $(m+1) \times m$. A obtenção de y é relativamente fácil dado o formato de Hessemberg de H_m : os elementos abaixo da primeira subdiagonal são nulos. Para tanto, a matriz H_m pode ser decomposta em matrizes Q e R , segundo a fatoração QR, construídas aplicando-se rotações de Givens a fim de zerar os elementos abaixo da diagonal (processo detalhado em Saad (2000)). A variável y é obtida a partir de retro-substituição no sistema $Ry = \beta Q^T e_1$. Tendo-se y , corrige-se x segundo a equação (5.23).

É interessante observar que caso se escolha $m = n$, o método GMRES é direto, pois o subespaço $x_0 + \Gamma$ irá conter necessariamente a solução do sistema linear. Entretanto, para sistemas lineares de grande dimensão dificilmente se faz essa escolha porque as matrizes V e H , que a princípio não são esparsas, iriam requerer grande capacidade de armazenamento, além de envolver um número enorme de operações. Na realidade é usual escolher uma dimensão pequena para o subespaço de Krylov, igual ou inferior a 10 (STRANG, 2006a). A solução obtida dificilmente será a solução do sistema linear, mas ela pode ser empregada para a construção de um novo subespaço de Krylov, do qual resulta numa nova solução aproximada. Procedendo-se assim até a convergência tem-se o que é denominado na literatura de *Restarted GMRES* (SAAD, 2000), método que foi utilizado neste trabalho. Na listagem 5.3 há o seu algoritmo.

Listagem 5.3: *Restarted GMRES*, adaptado de Saad (2000)

-
- 1 Calcule $r_0 = b - Ax_0$, $\beta = \|r_0\|_2$ e $v_1 = r_0/\beta$.
 - 2 Gere a base de Arnoldi segundo o método de Gramm–Schmidt modificado.
 - 3 Calcule a solução de $H_m y = \beta e_1$ via rotações de Givens e retro–substituição.

- 4 Calcule a nova solução aproximada $x = x_0 + V_m y$.
 - 5 Se $x \approx x_0$, pare. Caso contrário, retorne ao item 1.
-

5.2.3 MÉTODO DO GRADIENTE CONJUGADO

O método do Gradiente Conjugado é especialmente concebido para sistemas lineares em que a matriz é simétrica e positiva definida. Ele é considerado um método semi-direto, pois, embora em cada iteração se obtenha uma solução aproximada, é necessário no máximo um número de iterações igual à dimensão do sistema linear para se chegar à solução exata (STRANG, 1988).

Define-se no método do Gradiente Conjugado que $\Gamma_k = \Xi_k$, sendo, pois, válido o resultado da equação (5.13). Escolhe-se para Γ_k o subespaço de Krylov k -dimensional κ_k (STRANG, 2006b), que é o subespaço gerado por $r_0, Ar_0, \dots, A^{k-1}r_0$. Em cada iteração se aumenta em uma unidade a dimensão o subespaço de Krylov. Se n é a dimensão da matriz do sistema linear, então na n -ésima iteração o subespaço de Krylov será equivalente a \mathbb{R}^n e, portanto, a solução encontrada nessa iteração deve ser a exata.

Seja $r_1 = r_0 - A\Delta x_0$ o resíduo calculado após a primeira iteração do método. O resíduo inicial r_0 por definição pertence a κ_1 , assim como também por definição deve pertencer Δx_0 . Logo, r_0 e Δx_0 são colineares e, portanto, existe $c_1 \in \mathbb{R}$ tal que

$$r_1 = r_0 - c_1 Ar_0. \quad (5.26)$$

Como $Ar_0 \in \kappa_2$, $r_1 \in \kappa_2$. O resíduo da segunda iteração, por sua vez, é $r_2 = r_1 - A\Delta x_1$. Sabe-se que $r_1 \in \kappa_2$ e, como $\Delta x_1 \in \kappa_2$, $\exists c_1, c_2 \in \mathbb{R}$ tais que $\Delta x_1 = c_1 r_0 + c_2 Ar_0$. Portanto, $r_2 = r_1 - c_1 Ar_0 - c_2 A^2 r_0$ e, conseqüentemente, $r_2 \in \kappa_3$. Aplicando-se esse raciocínio indutivamente se conclui que $r_k \in \kappa_{k+1}$.

Da definição de método de projeção dada na equação (5.12), $r_k \perp \Xi_k$ e, como $\Xi_k = \Gamma_k = \kappa_k$ e $r_k \in \kappa_{k+1}$, segue que

$$r_k^T r_i = 0, \quad 0 \leq i \leq k-1, \quad (5.27)$$

ou seja, cada novo resíduo obtido é ortogonal a todos os resíduos das iterações anteriores. Outra propriedade, que acaba por definir o nome do método, é que

$$\Delta x_i A \Delta x_k, \quad 0 \leq i \leq k-1, \quad (5.28)$$

ou seja, as correções, também denominadas de gradientes (STRANG, 2006b), são ortogonais na norma A , ou seja, são conjugadas. Das duas condições acima pode-se deduzir o algoritmo do Gradiente Conjugado, contido na listagem 5.4. Os detalhes dessa dedução estão presentes em Saad (2000).

Listagem 5.4: Algoritmo do Gradiente Conjugado (adaptado de Strang (2006b))

```

1  Calcule  $r_0 = b - Ax_0$ ,  $d_0 \leftarrow r_0$ 
2  Para  $j = 0, 1, \dots$  até a convergência faça
3       $\alpha_j \leftarrow (r_j, r_j) / (Ad_j, d_j)$ 
4       $x_{j+1} \leftarrow x_j + \alpha_j d_j$ 
5       $r_{j+1} \leftarrow r_j - \alpha_j Ad_j$ 
6       $\beta_j \leftarrow (r_{j+1}, r_{j+1}) / (r_j, r_j)$ 
7       $d_{j+1} \leftarrow r_{j+1} + \beta_j d_j$ 
8  Fim – faça

```

5.3 MÉTODO BASEADO NA FATORAÇÃO LU

A fatoração LU consiste em decompor via eliminação gaussiana a matriz original A nas matrizes L e U tais que $A = LU$. A matriz L é triangular inferior, de modo que os elementos acima da diagonal são nulos, ao passo que a U é triangular superior. Para se resolver o sistema linear $Ax = b$ resolve-se inicialmente o sistema $Lc = b$ e, então, o $Ux = c$. Os sistemas lineares envolvendo L e U são relativamente simples de se resolver, pois essas matrizes são triangulares e, assim, pode-se aplicar retro-substituição. Enquanto o sistema $Ax = b$ demanda, caso se utilize eliminação gaussiana, da ordem de $n^3/3$ operações, sendo n a dimensão de A , os sistemas $Lc = b$ e $Ux = c$ demandam da ordem de $n^2/2$ cada (STRANG, 1988).

Na realidade a fatoração LU é interessante apenas quando se deseja resolver vários sistemas lineares em que a matriz é a mesma. Embora o custo computacional de se resolver os sistemas $Lc = b$ e $Ux = c$ seja bem inferior ao do sistema $Ax = b$ via eliminação gaussiana, a obtenção das matrizes L e U envolve da ordem de n^3 operações. Em particular, dado o custo computacional das matrizes L e U e a necessidade de se armazenar uma grande quantidade de coeficientes, a fatoração LU raramente é aplicada na solução de sistemas lineares de grande dimensão. No multigrid, por outro lado, a fatoração LU acaba se mostrando in-

interessante no nível mais grosseiro, em que o sistema linear apresenta pequena dimensão. A fatoração, que nesse caso apresenta baixo custo computacional, é realizada uma única vez. Todos os sistemas lineares desse nível são então resolvidos de maneira exata, envolvendo um número fixo e relativamente pequeno de operações.

6 RESULTADOS

A fim de não tornar o trabalho demasiadamente longo, serão apresentados apenas os resultados principais. Eles foram obtidos aplicando-se a biblioteca multigrid no simulador Orion, desenvolvido no projeto MaTra pelo SINMEC. O fato desse multigrid ser do tipo algébrico, não requerendo, portanto, informações geométricas, permitiu fácil acoplamento da biblioteca ao simulador. Do projeto MaTra foram também utilizadas algumas malhas poligonais e poliédricas. Há malhas de diversas características e graus de refino, o que vem permitir um estudo adequado da relação entre o número de volumes e o tempo computacional requerido para a solução dos sistemas lineares.

Há diversos fatores que influenciam no desempenho do multigrid. Porém, nesse trabalho a análise está restrita aos métodos de solução de sistemas lineares utilizados nos níveis do multigrid. Na tabela 6.1 estão alguns parâmetros que serão mantidas constantes neste trabalho. Os dois primeiras são parâmetros que controlam a aglomeração. A primeira delas indica que cada volume de uma nova malha será construído aglomerando-se aproximadamente cinco volumes da malha anterior. Assim, se a malha original apresentar 1000 volumes, a segunda deve ter aproximadamente 200, a terceira, 40, e assim por diante. O processo de criação de malhas continua até que a malha mais grosseira apresente um número igual ou inferior ao estabelecido pelo segundo parâmetro definido na tabela 6.1. Como as malhas bidimensionais aqui empregadas apresentam em geral número inferior de volumes, admite-se um número máximo de volumes na malha mais grosseira também inferior. Finalmente, o ciclo multigrid utilizado é o F , a fim de que os resultados obtidos possam ser comparados com aqueles de Pescador Jr (2010).

Tabela 6.1: Valores mantidos fixos para os parâmetros de controle do multigrid

Parâmetro	Valor
Número médio de volumes por bloco	5
Número máximo de volumes na malha grossa	10 (2D) e 40 (3D)
Ciclo	F
Tolerância do erro	10^{-6}

Para a verificação da convergência define-se o erro na iteração k como sendo

$$e^{(k)} = \|b - Ax^{(k)}\|_{\infty}, \quad (6.1)$$

ou seja, o erro é igual a maior componente em módulo do vetor resíduo. Deve ficar claro

que essa definição é distinta daquela da equação (4.3) e que ela foi feita apenas por conveniência de terminologia. Caso $x^{(k)}$ coincida com a solução exata do sistema linear, o erro será zero tanto na definição da equação (4.3) quanto na da equação (6.1). Diz-se que houve convergência caso o erro seja menor que a tolerância especificada na tabela 6.1. Por outro lado, diz-se que houve divergência caso para algum k

$$\frac{e^{(k)}}{e^{(0)}} > 1000. \quad (6.2)$$

Foram escolhidas três classes de malhas para a execução dos testes: regular, de elevada razão de aspecto e tridimensional com distorção randômica. A malha regular é a malha de referência, da qual resulta, caso o tensor de permeabilidade seja isotrópico, uma matriz com coeficientes próximos entre si. Já na malha de elevada razão de aspecto, denominada aqui de *stretched*, a anisotropia geométrica faz com que os coeficientes da matriz sejam também anisotrópicos, de sorte que a escala de tempo de propagação da informação na direção x é sensivelmente distinta da na direção y . Quanto à malha tridimensional, seus elementos apresentam número de vizinhos bem superior às bidimensionais, levando, sobretudo se o esquema totalmente implícito for utilizado, a matrizes com grau de esparsidade bem menor. Além disso, aplicou-se nela distorção randômica em seus vértices, originando elementos distorcidos e não necessariamente convexos.

Em cada nível do multigrid foi especificado um mesmo método de solução de sistema linear, excetuando-se no nível mais grosseiro, em que sempre se aplicou a fatoração LU. Escolheram-se os métodos: do Gradiente Conjugado; de Gauss-Seidel; GMRES e MINRES. O método de Jacobi não foi selecionado porque ele é semelhante ao método de Gauss-Seidel, embora em geral seja menos eficiente. No caso do GMRES, as dimensões do subespaço de Krylov especificadas foram 2, 3, 5 e 10. Dimensões maiores, embora tendam a tornar o método mais robusto, acabam reduzindo o desempenho do multigrid. Ressalta-se que não se utilizou dimensão 1 porque nesse caso a solução de cada iteração dos métodos GMRES e MINRES é a mesma, embora o algoritmo deste seja mais simples que o daquele.

Em cada nível do multigrid são executados um número fixo de iterações do *solver* especificado. Pescador Jr (2010) observou que o desempenho do multigrid empregando o Gauss-Seidel como *solver* é melhorado caso se adote um número baixo de iterações por nível, preferivelmente apenas uma. Com base nisso foram definidos números de iterações relativamente pequenos: 1, 2, 5, 7, 10, 15.

Os resultados foram obtidos em um computador com processador Intel Core i7-2600, 3,4 GHz, 8 GB de memória RAM, sistema operacional Windows 7 Home Premium 64 bits e

compilador Visual Studio C++ 2008 Professional. Os tempos computacionais foram medidos por meio da classe `timer`, disponível na biblioteca BOOST. Nos casos com esquema semi-implícito, o tempo computacional avaliado é igual ao somatório dos tempos necessários para resolver cada sistema linear do laço iterativo descrito na subseção 3.3.1. No esquema totalmente implícito, por outro lado, por não ser necessário ciclo iterativo adicional, o tempo computacional corresponde à solução de um único sistema linear.

6.1 MALHA REGULAR BIDIMENSIONAL

A malha regular bidimensional, ilustrada na figura 6.1, é caracterizada por apresentar predominantemente hexágonos quase regulares (eles são levemente deformados na vertical para se adequarem à geometria). Especificou-se Φ unitário na fronteira norte e nulo nas demais. O meio é isotrópico e homogêneo.

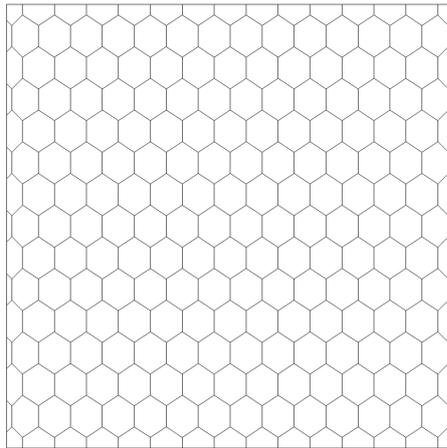


Figura 6.1: Malha regular bidimensional

6.1.1 ESQUEMA SEMI-IMPLÍCITO

De forma geral, o desempenho foi melhor para apenas uma iteração em cada nível. Os resultados do método de Gauss-Seidel, ilustrados na figura 6.2, vêm a confirmar aqueles obtidos por Pescador Jr (2010). Houve, entretanto, algumas peculiaridades. Utilizando-se o método do Gradiente Conjugado, o melhor desempenho nas malhas mais refinadas se deu para duas iterações por nível, de acordo com o gráfico da figura 6.3. Com apenas uma iteração, o método ficou instável, divergindo na malha mais refinada e demorando a convergir na malha de 4128 elementos.

Outro resultado que merece destaque é que o MINRES com apenas uma iteração por nível apenas convergiu nas duas primeiras malhas. Com duas iterações esse método obteve seu melhor desempenho, embora seus pontos no gráfico não tenham ficado bem alinhados, indicando certa instabilidade. O gráfico do multigrid com método MINRES está ilustrado na figura 6.4.

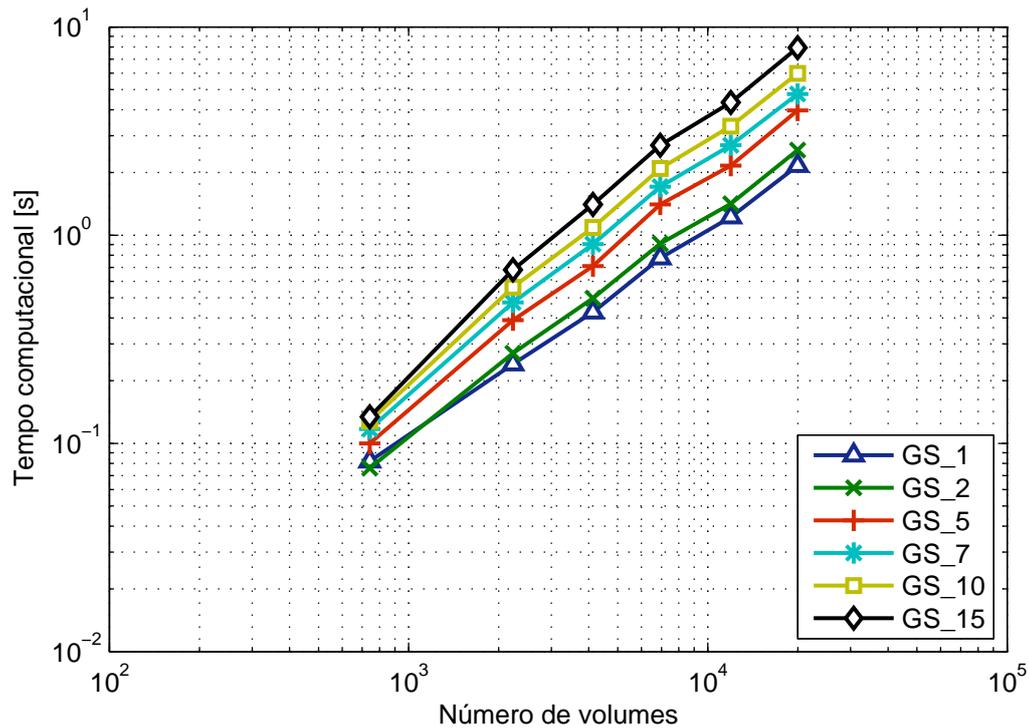


Figura 6.2: Resultados do multigrid com método de Gauss-Seidel e esquema semi-implícito

Para comparar os métodos entre si, tomou-se para cada um a configuração de melhor desempenho. Os resultados estão na figura 6.5. Observa-se que os resultados foram sensivelmente melhores para o método de Gauss-Seidel. Já o GMRES foi tão mais rápido quanto menor era a dimensão do subespaço de Krylov. Embora subespaços de maior dimensão resultem numa solução também mais próxima da exata, o custo computacional adicional não foi compensado.

A inclinação das retas que interpolam os pontos da figura 6.5 fornecem a complexidade do algoritmo. O método é tanto melhor quanto mais próximo da unidade estiver esse valor. Isso significa que o aumento do número de variáveis acarreta apenas aumento proporcional do tempo computacional. Na tabela 6.2 se tem a complexidade observada para cada caso¹. Fica, pois, claro que o multigrid com Gauss-Seidel e apenas uma iteração foi o melhor dentre

¹As complexidades foram obtidas interpolando-se os dados com uma curva do tipo potência. Caso os pontos não fiquem bem alinhados no gráfico *log log*, a curva acaba não os descrevendo bem e, assim, sua inclinação deixa de ter significado para a avaliação da complexidade

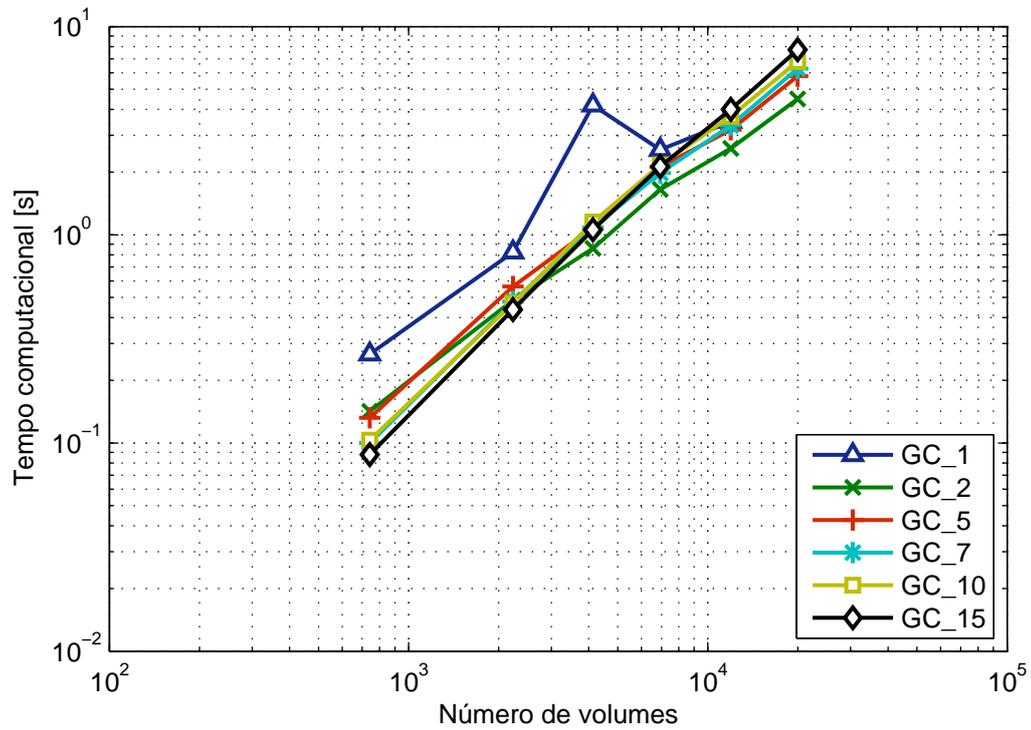


Figura 6.3: Resultados do multigrid com método do Gradiente Conjugado e esquema semi-implícito

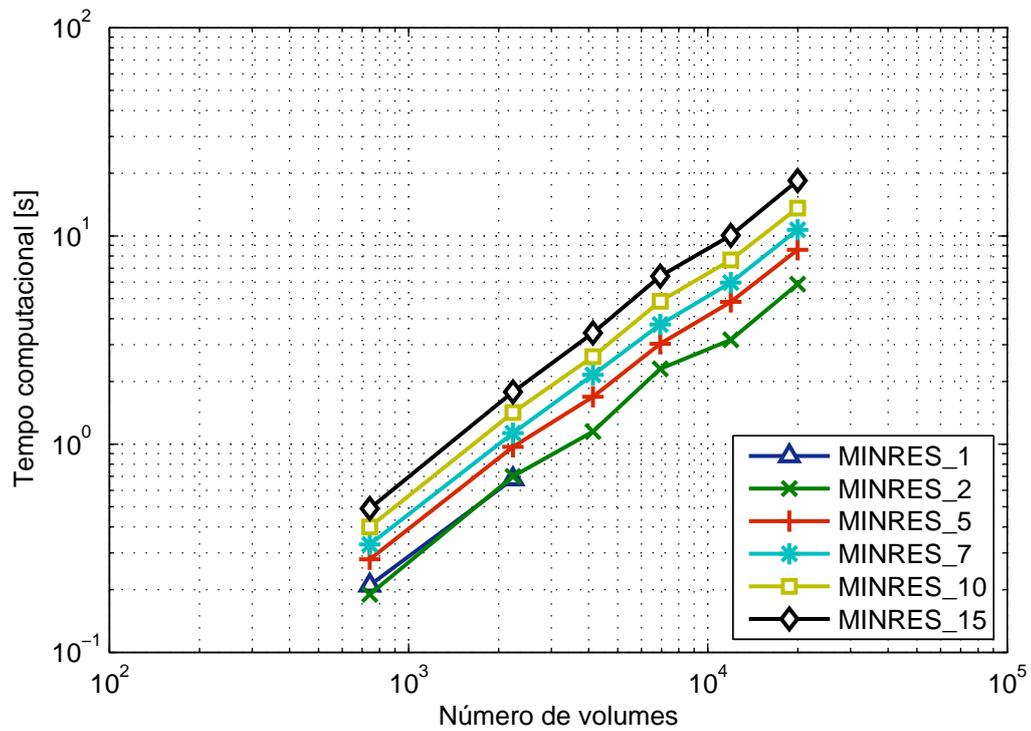


Figura 6.4: Resultados do multigrid com método MINRES e esquema semi-implícito

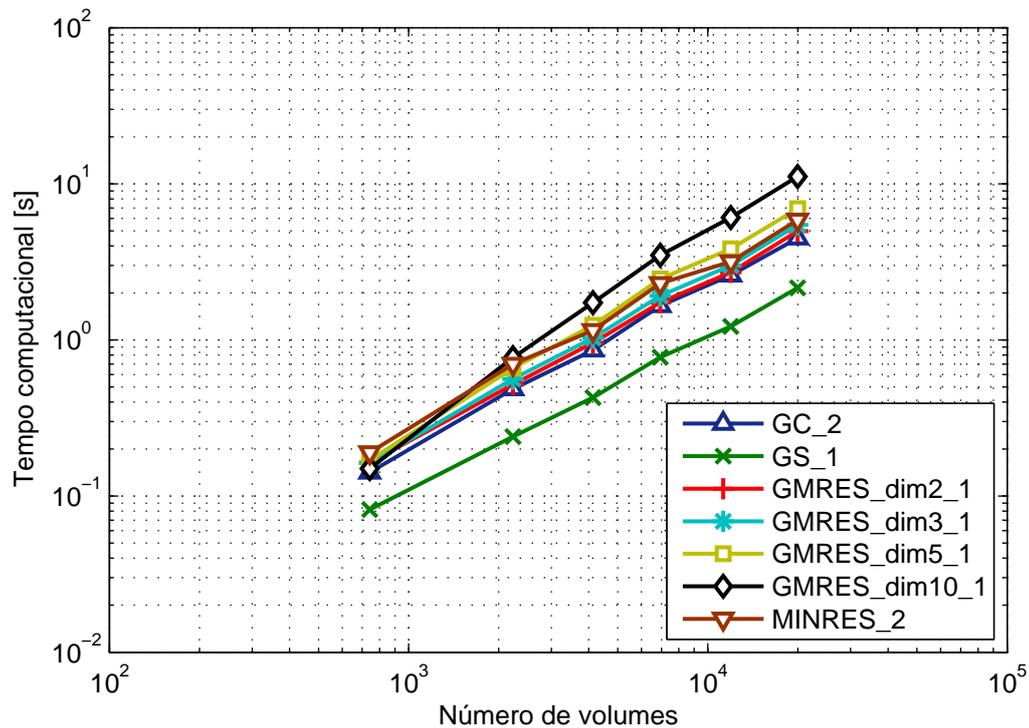


Figura 6.5: Resultados dos diferentes métodos com número de iterações ótimo e esquema semi-implícito

eles, apresentando grau de complexidade virtualmente igual à unidade. Nos casos em que se utilizou Gradiente Conjugado, GMRES com dimensão do espaço de Krylov dois e MINRES o grau de complexidade também ficou próximo da unidade, mas suas curvas no gráfico da figura 6.5 ficaram acima da do caso com Gauss-Seidel.

Tabela 6.2: Complexidade do multigrid com diversos métodos e esquema semi-implícito

Método	Complexidade
Gradiente Conjugado, 2 iter.	1,0463
Gauss-Seidel, 1 iter.	0,9904
GMRES dim. 2, 1 iter.	1,0290
GMRES dim. 3, 1 iter.	1,0572
GMRES dim. 5, 1 iter.	1,1196
GMRES dim. 10, 1 iter.	1,3043
MINRES, 2 iter.	1,0237

Para melhor entender os resultados anteriores pode-se supor que há apenas dois níveis do multigrid, sendo a solução do nível mais grosseiro calculada exatamente. A solução no nível fino é corrigida em cada ciclo pelo operador de interpolação do multigrid, de acordo com a equação (4.18), e pelo *solver* selecionado nesse nível, que executa um número fixo de

iterações. Ou seja,

$$\Delta x^{(k+1)} = \Delta x_{\text{solver}}^{(k)} + \Delta x_{\text{interpolação}}^{(k)}. \quad (6.3)$$

Como a solução no nível mais grosseiro é obtida exatamente, o número de operações executadas nesse nível é constante. No nível mais fino, por outro lado, a quantidade de operações depende do número de iterações especificado para o *solver*. Quanto maior for esse número, maior é a contribuição do *solver* do nível fino na *performance* global do multigrid. Caso esse número seja suficiente para a convergência da solução, o efeito do uso do multigrid é nulo e, assim, o grau de complexidade bem como os tempos computacionais se aproximam dos do *solver* do nível fino.

6.1.2 ESQUEMA TOTALMENTE IMPLÍCITO

Os resultados dos métodos utilizando-se o método totalmente implícito foram análogos aos obtidos com o método semi-implícito. Eles foram melhores para apenas uma iteração por nível, exceto nos métodos do Gradiente Conjugado e MINRES. No caso do Gradiente Conjugado, cujos resultados estão presentes na figura 6.6, com apenas uma iteração não houve convergência nas duas malhas mais refinadas, embora não tenha havido, como no caso do semi-implícito, um tempo computacional excessivo na malha de 4128 elementos. Novamente o desempenho com esse método foi superior para duas iterações por nível. Percebe-se, entretanto, que com um maior número de iterações os pontos da curva do tempo computacional ficam melhor alinhados, o que pode indicar maior robustez do multigrid.

Os resultados do multigrid com método MINRES, cujo gráfico é o da figura 6.7, também se mostraram interessantes. Não houve convergência nas duas malhas mais refinadas e na malha de 4128 elementos ao se aplicar apenas uma iteração por nível. Para maior quantidade de iterações houve convergência em todas as malhas, embora os tempos computacionais na malha de 4128 não tenham seguido o padrão determinado pelas outras. Cabe salientar que as malhas aqui denominadas de regulares não são uniformes. Deve haver algum pequeno detalhe na malha de 4128 elementos que faz com que os resultados destoem das outras.

Na figura 6.8 estão os resultados dos métodos avaliados na situação de número de iterações ótimo. Novamente o multigrid com método de Gauss-Seidel e uma iteração por nível foi superior aos demais. Já a figura 6.9 contém os resultados obtidos na malha mais refinada do decaimento do erro, definido nesse capítulo como a norma infinito do vetor resíduo, em

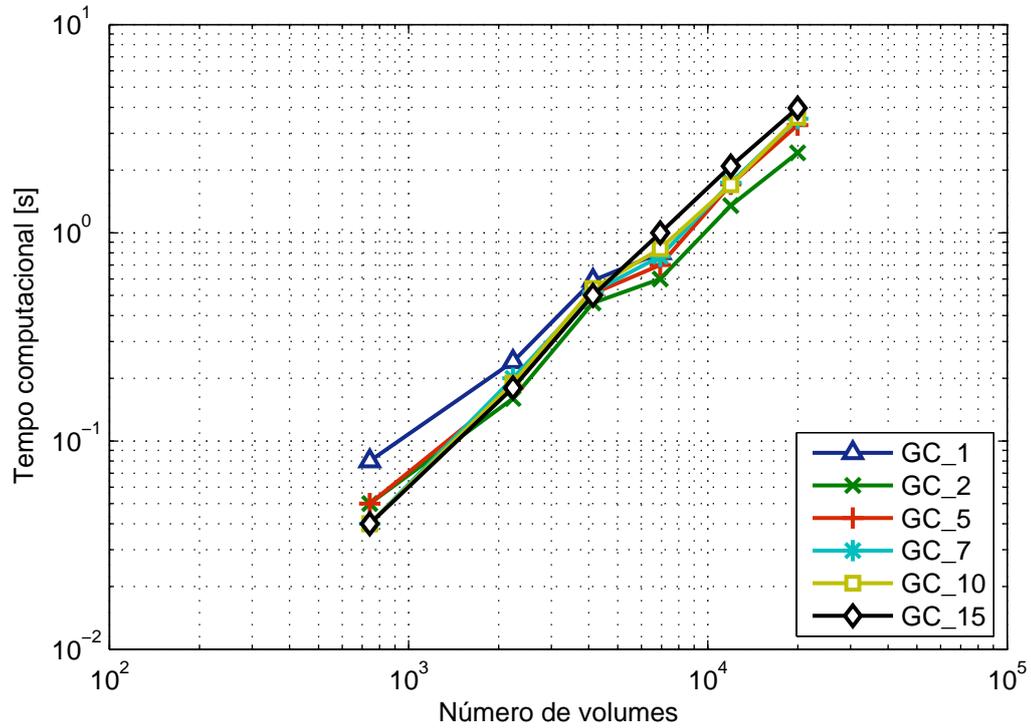


Figura 6.6: Resultados do múltigrid com Gradiente Conjugado e método totalmente implícito

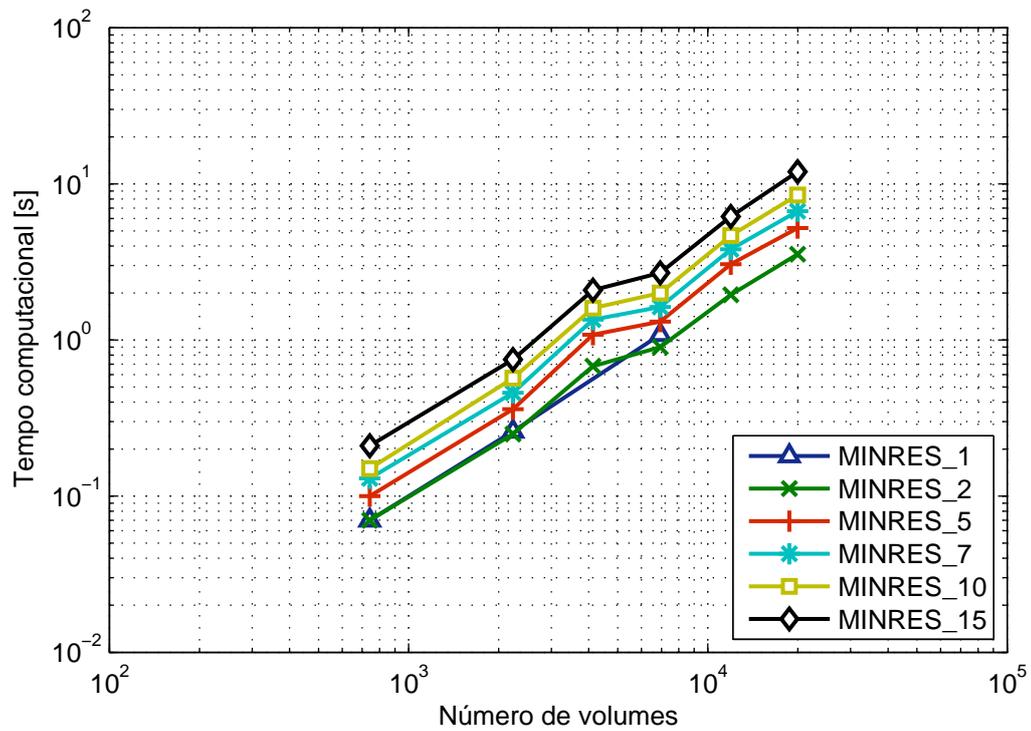


Figura 6.7: Resultados do multigrid com método MINRES e esquema totalmente implícito

função do número de ciclos multigrid executados. No caso do multigrid com MINRES, as flutuações do erro ao longo dos ciclos multigrid indicam que o método é instável. De fato, com apenas uma iteração por nível o método nem chega a convergir em todas as malhas. O multigrid com GMRES, por outro lado, aparentemente é tão mais estável quanto maior é a dimensão do subespaço de Krylov. Além disso, quanto maior for essa dimensão, menor será o número de ciclos necessários para que o critério de convergência seja atendido, algo natural visto que uma maior dimensão induz a melhores soluções parciais em cada iteração, embora acarrete em maior custo computacional. Já o multigrid com os métodos de Gauss-Seidel e do Gradiente Conjugado se mostrou bastante estável, praticamente não havendo flutuações no decaimento do erro. No caso do método de Gauss-Seidel, em particular, o decaimento do erro, na escala logarítmica, é significativamente maior nos primeiros ciclos, sendo progressivamente menor nos seguintes. Com o método do Gradiente Conjugado, em contrapartida, esse decaimento foi praticamente constante.

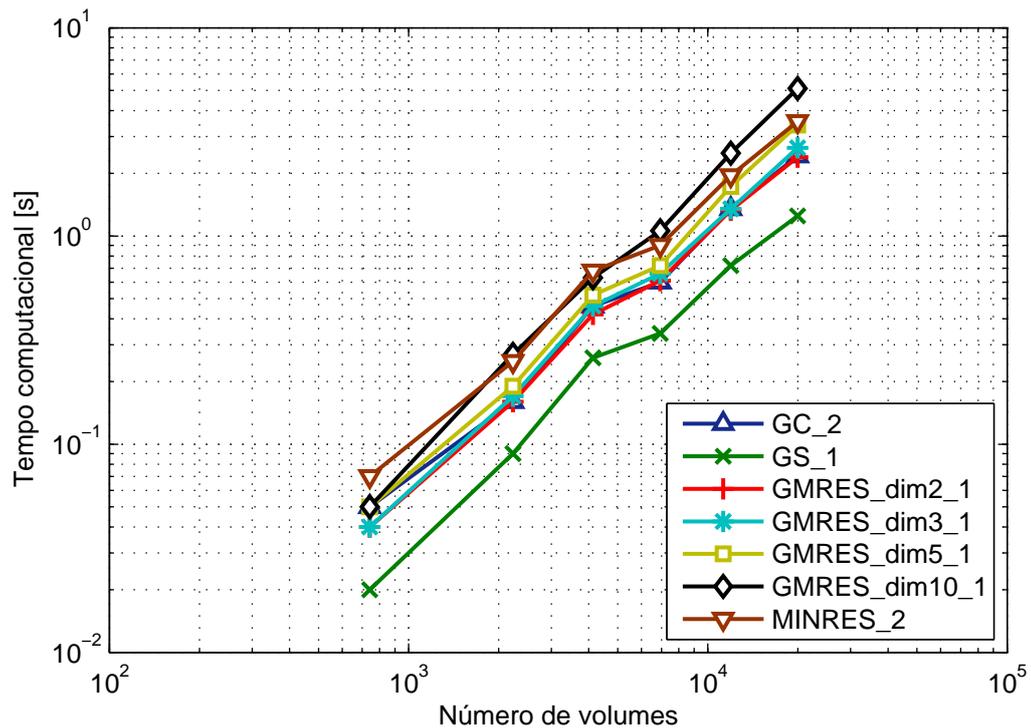


Figura 6.8: Resultados dos diferentes métodos com número de iterações ótimo e esquema totalmente implícito

Na tabela 6.3 se tem o grau de complexidade dos algoritmos, o qual foi calculado descon siderando-se os resultados da malha de 4128 elementos, que não seguiram o padrão. Fica, pois, claro que o grau de complexidade aumentou em relação ao do esquema semi-implícito. O provável motivo é a redução da esparsidade da matriz do sistema linear.

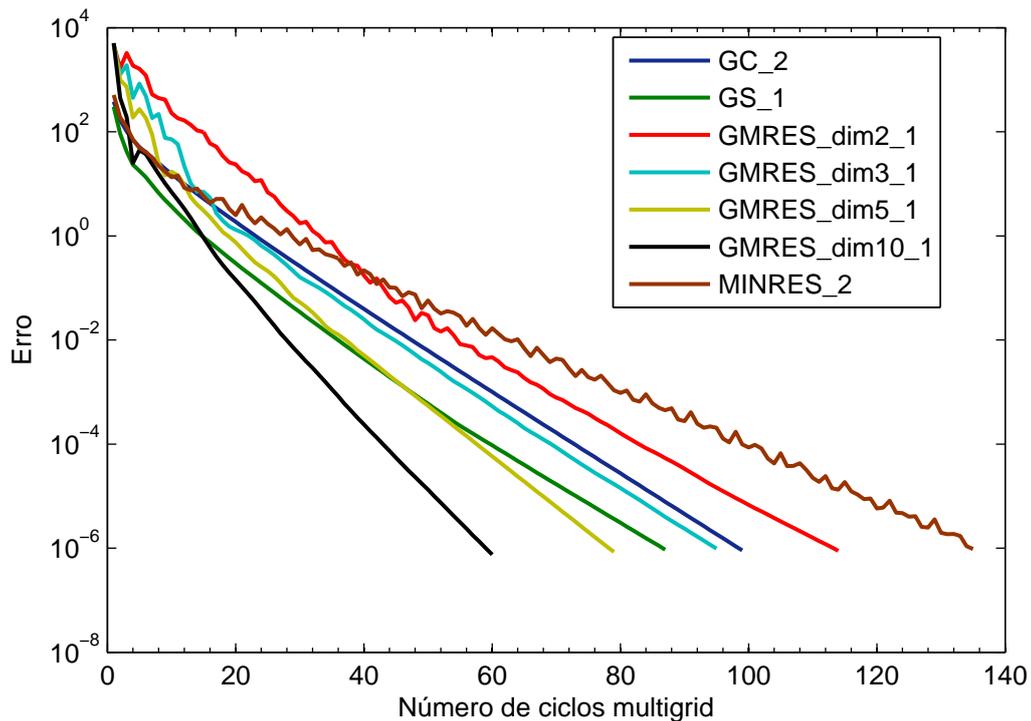


Figura 6.9: Erro em função do número de ciclos multigrid executados

Tabela 6.3: Complexidade do multigrid com diversos métodos e esquema totalmente implícito

Método	Complexidade
Gradiente Conjugado, 2 iter.	1,1888
Gauss-Seidel, 1 iter.	1,2562
GMRES dim. 2, 1 iter.	1,2447
GMRES dim. 3, 1 iter.	1,2644
GMRES dim. 5, 1 iter.	1,2789
GMRES dim. 10, 1 iter.	1,3870
MINRES, 2 iter.	1,1934

6.2 MALHA COM GRANDE RAZÃO DE ASPECTO

As malhas de grande razão de aspecto (*stretched*) aqui utilizadas se caracterizam essencialmente por serem compostas por hexágonos cuja dimensão vertical é bem superior à horizontal, conforme mostra a figura 6.10. Isso acaba por introduzir anisotropia nos coeficientes do sistema linear. O fato de os coeficientes relativos à direção vertical serem bem inferiores aos da horizontal dificulta a propagação de informação nessa direção (KELLER, 2007).

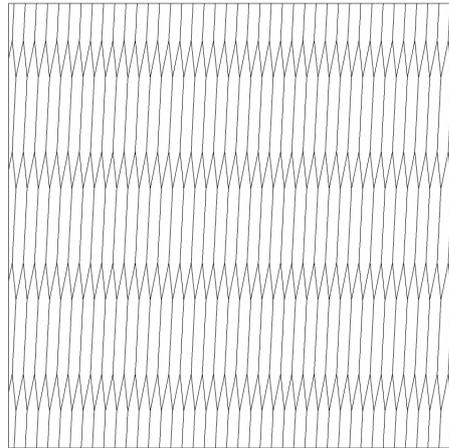


Figura 6.10: Malha bidimensional de grande razão de aspecto

6.2.1 ESQUEMA SEMI-IMPLÍCITO

Constatou-se nas malhas de grande razão de aspecto, ou *stretched*, dificuldade de convergência bem maior que nas malhas regulares. No multigrid com método do Gradiente Conjugado, cujos resultados constam na figura 6.11, com uma e duas iterações por nível só houve convergência nas três malhas mais grosseiras. O aumento do número de iterações, por sua vez, acabou por tornar o método mais robusto e, para valores superiores a sete, os tempos computacionais foram semelhantes.

Os resultados do multigrid com Gauss-Seidel, presentes na figura 6.12, foram semelhantes aos do com Gradiente Conjugado. Quanto menor o número de iterações por nível menos robusto é o método. Dentre os casos avaliados, somente aquele com 15 iterações convergiu em todas as malhas.

A relação entre robustez e número de iterações também foi confirmada nos casos envolvendo GMRES. No GMRES, em particular, verificou-se que a robustez, além de melhorar com o aumento do número de iterações, também aumenta com o incremento da dimensão do subespaço de Krylov. De fato, na figura 6.13 estão os resultados para dimensão dois e na figura 6.14, para a 10. Com dimensão dois, só se atingiu a convergência em todas as malhas com número de iterações superior a 10, ao passo que com dimensão 10 houve convergência em todas as malhas para qualquer número de iterações. É interessante notar também que, caso haja convergência, o tempo computacional sempre é menor para o caso em que o número de iterações é menor.

O multigrid com *solver* MINRES foi o de pior desempenho dentre os avaliados. Da figura 6.15 vê-se que em nenhum dos casos houve convergência em todas as malhas. Em particu-

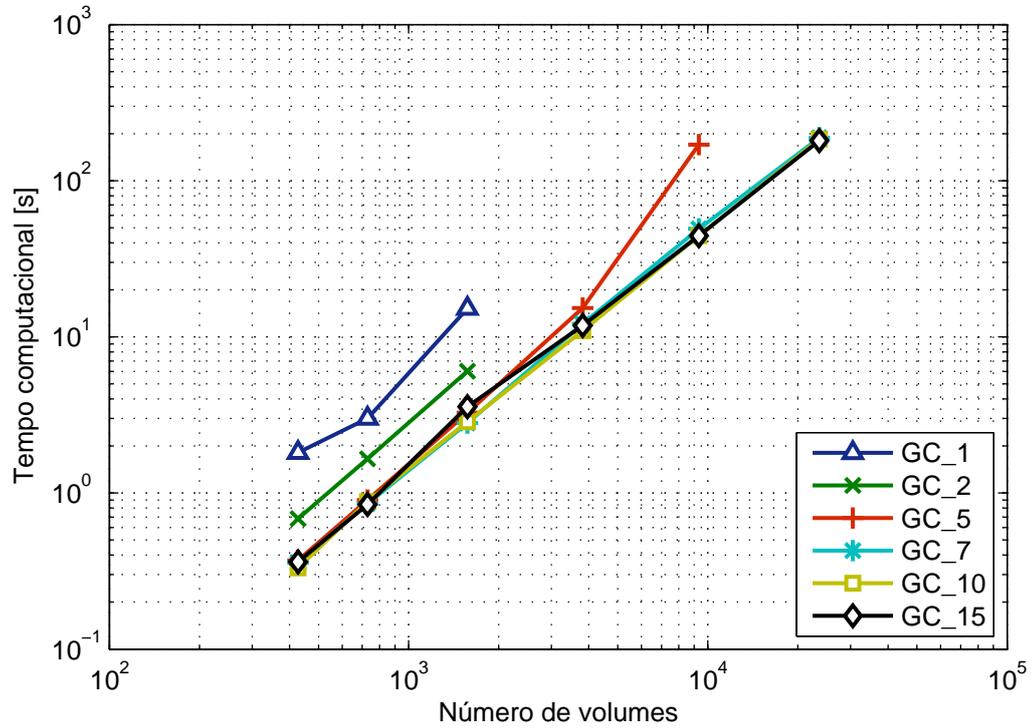


Figura 6.11: Resultados do multigrid com método do Gradiente Conjugado e esquema semi-implícito

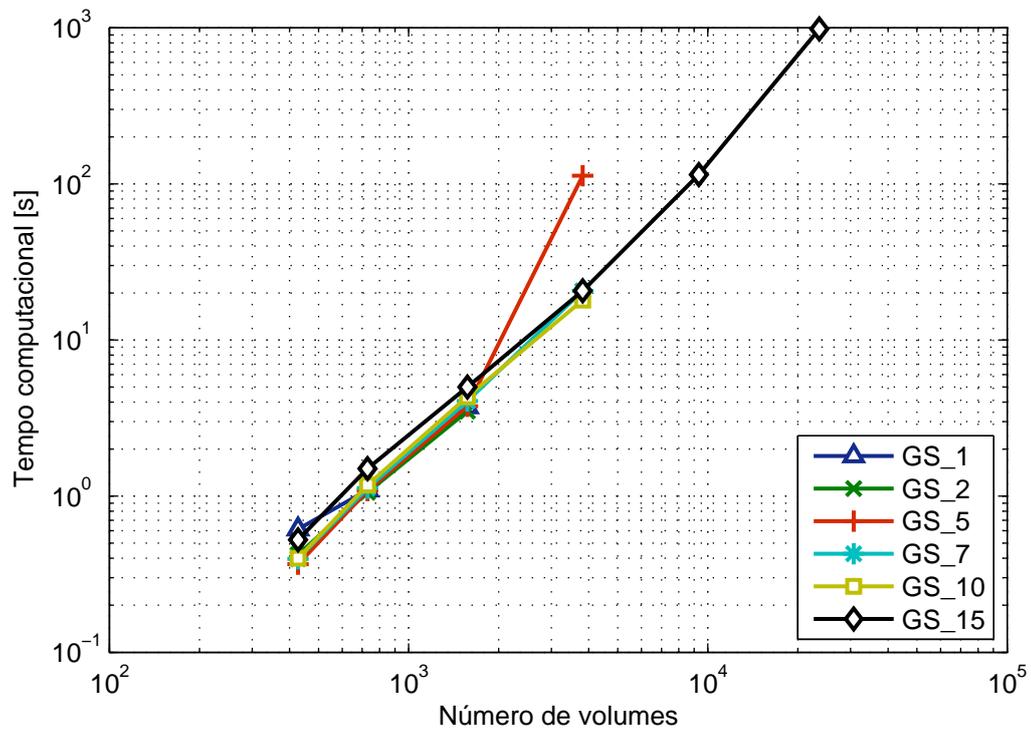


Figura 6.12: Resultados do multigrid com método de Gauss-Seidel e esquema semi-implícito

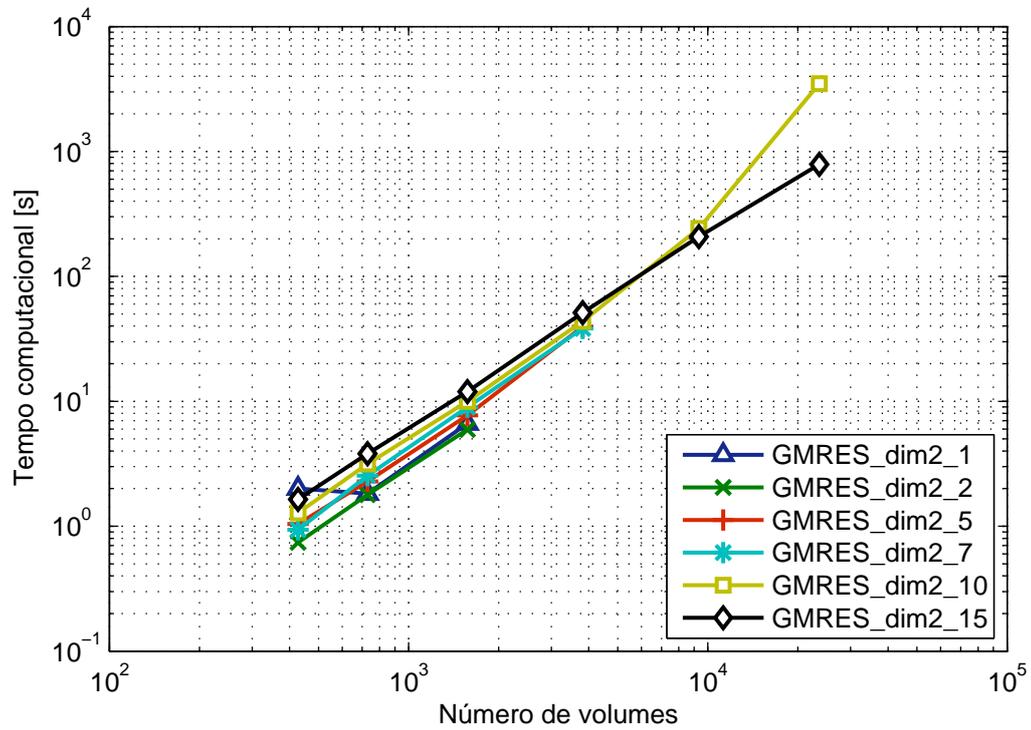


Figura 6.13: Resultados do multigrid com método GMRES de dimensão 2 e esquema semi-implícito

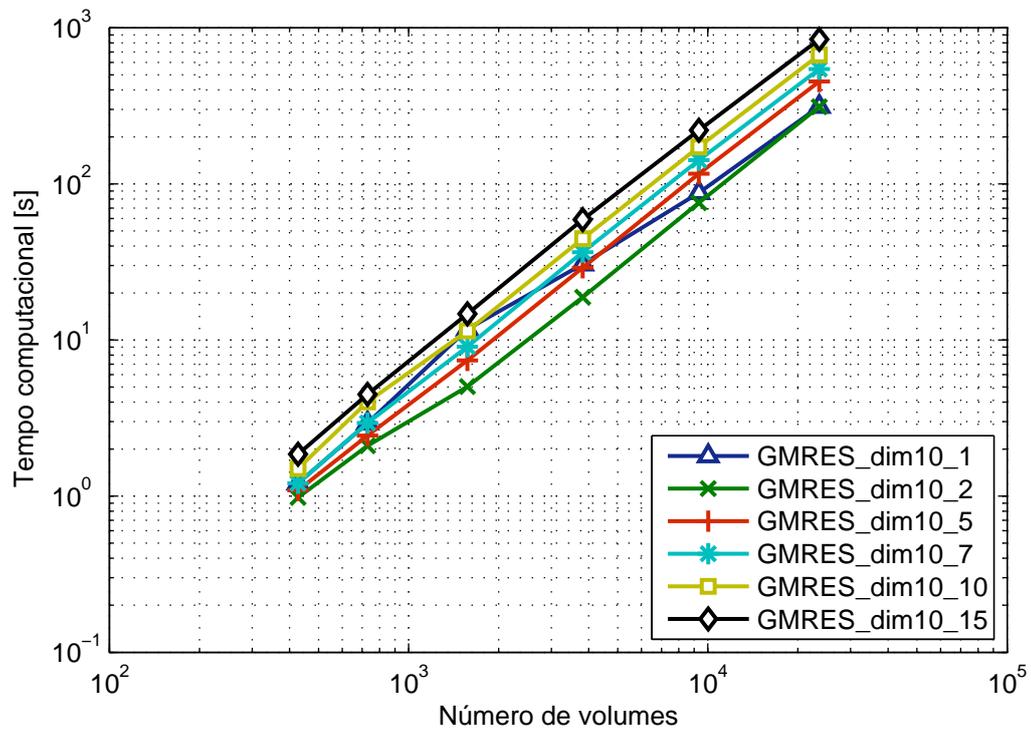


Figura 6.14: Resultados do multigrid com método GMRES de dimensão 10 e esquema semi-implícito

lar, para apenas uma iteração por nível não houve sequer convergência. Porém, o método aparenta ficar mais robusto com o aumento do número de iterações. Talvez com um valor suficientemente alto se consigam melhores resultados.

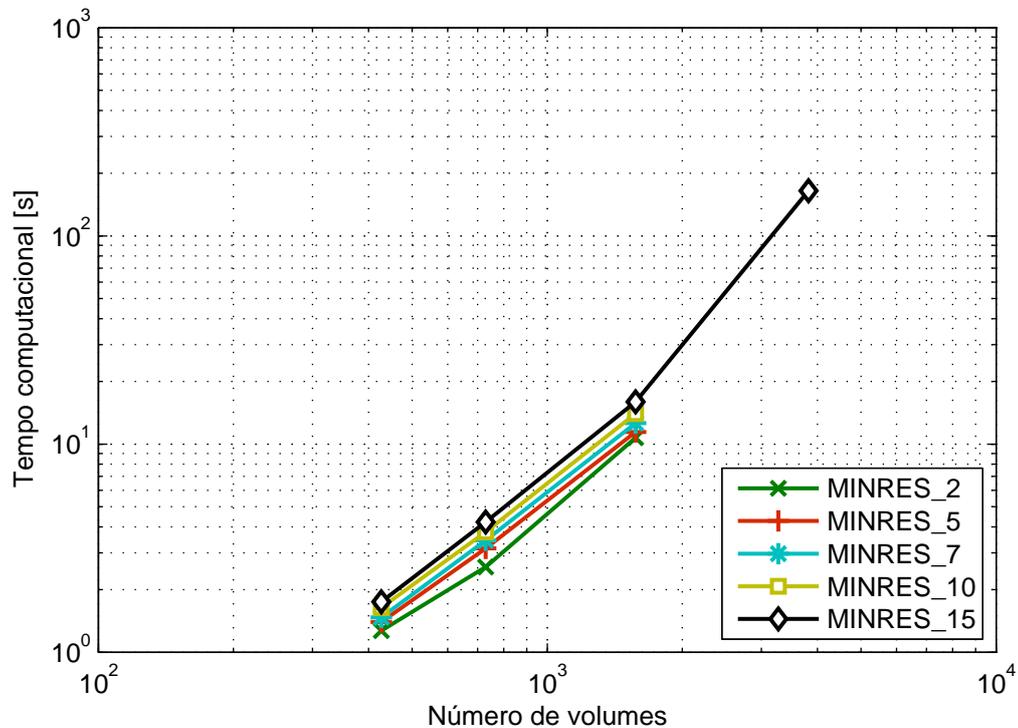


Figura 6.15: Resultados do multigrid com método MINRES e esquema semi-implícito

Na figura 6.16 estão os melhores resultados encontrados para cada método. Verifica-se que, diferentemente do que se observou na malha regular, foi o multigrid com Gradiente Conjugado que apresentou o melhor desempenho. Dentre os casos envolvendo o GMRES, aqueles com maior dimensão do subespaço de Krylov foram melhores. Na tabela 6.4 estão os valores obtidos para as complexidades dos métodos. Não foram considerados os métodos de Gauss-Seidel e MINRES por seus pontos não terem ficado bem alinhados no gráfico. Para o GMRES com dimensão 10 foram eliminados os dois primeiros pontos por eles destoarem do padrão.

Tabela 6.4: Complexidade do multigrid com diversos métodos e esquema semi-implícito

Método	Complexidade
Gradiente Conjugado, 10 iter.	1,5617
GMRES dim. 2, 15 iter.	1,5510
GMRES dim. 3, 7 iter.	1,5623
GMRES dim. 5, 2 iter.	1,5509
GMRES dim. 10, 2 iter.	1,5275

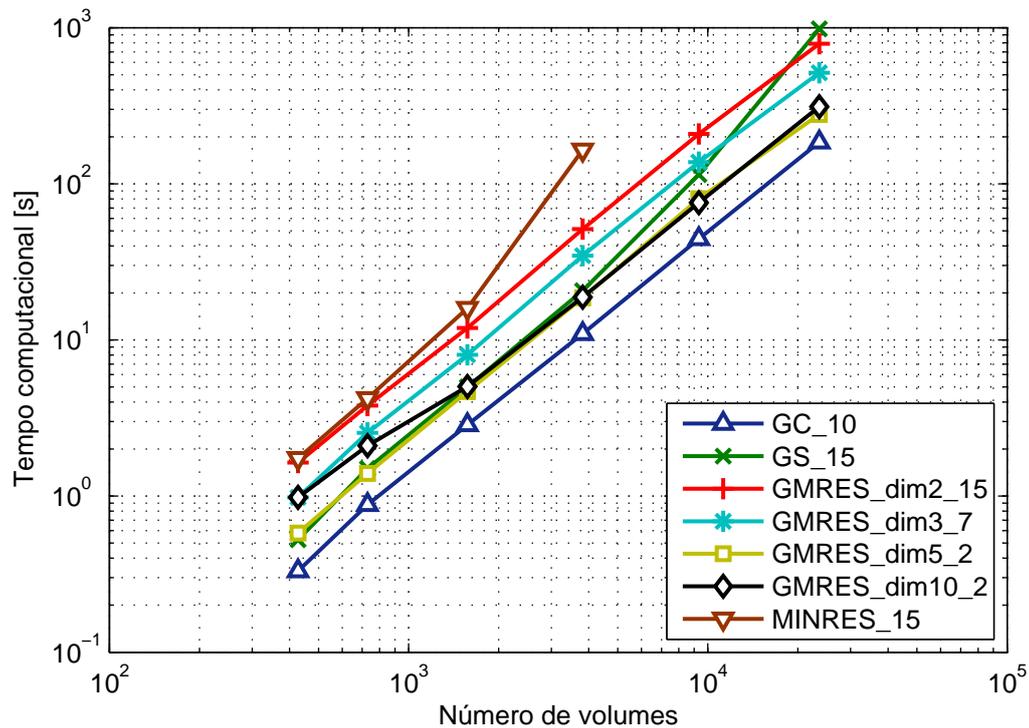


Figura 6.16: Resultados dos diferentes métodos com número de iterações ótimo e esquema semi-implícito

6.2.2 ESQUEMA TOTALMENTE IMPLÍCITO

Mesmo sendo as matrizes do esquema totalmente implícito menos esparsas e possivelmente não diagonal dominantes, houve mais casos que convergiram do que no esquema semi-implícito. Na figura 6.17 estão os resultados do multigrid com Gauss-Seidel. Enquanto que no esquema semi-implícito apenas o caso com 15 iterações por nível tinha convergido em todas as malhas, aqui cinco iterações foram suficientes para assegurar a convergência. Entretanto, especificar um número excessivo de iterações acaba por tornar o método mais lento.

Os resultados do multigrid com Gradiente Conjugado, contidos na figura 6.18, foram análogos àqueles do esquema semi-implícito. Com poucas iterações, além de o método ficar mais lento, ele torna-se menos robusto. Porém, diferentemente do caso anterior, o aumento do número de iterações, estando a convergência assegurada, levou a uma redução dos tempos computacionais.

Para o GMRES novamente se observou a relação descrita anteriormente entre a dimensão do subespaço de Krylov e o número de iterações. Dimensões pequenas requerem um número de iterações maior para a convergência, enquanto que dimensões maiores admi-

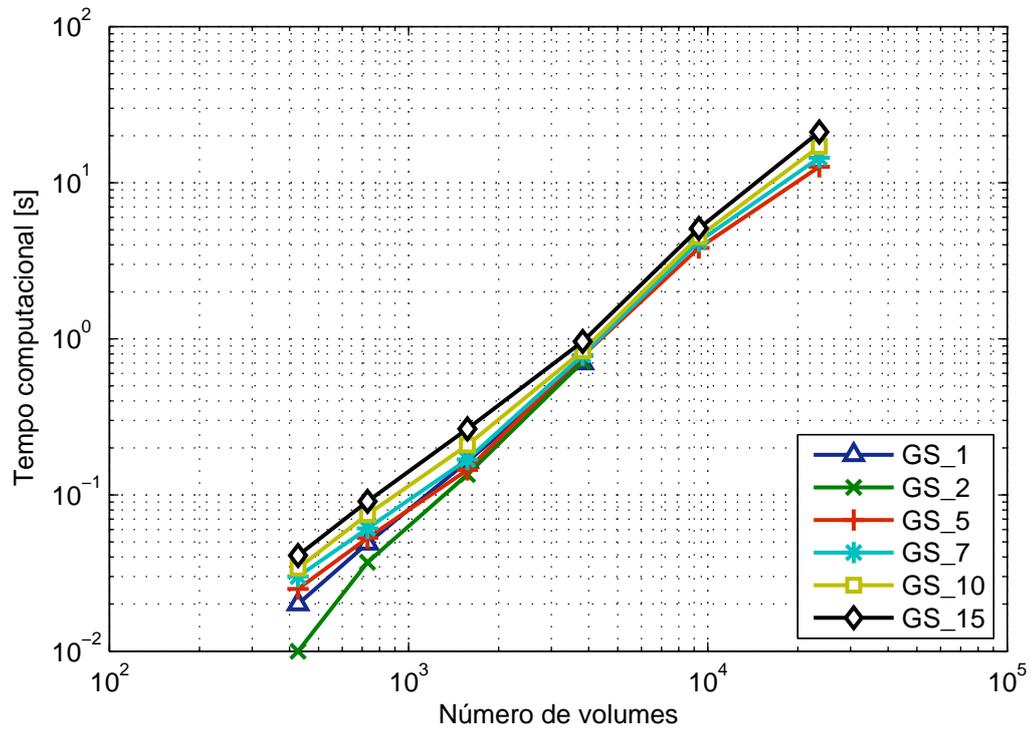


Figura 6.17: Resultados do multigrid com método de Gauss-Seidel e esquema totalmente implícito

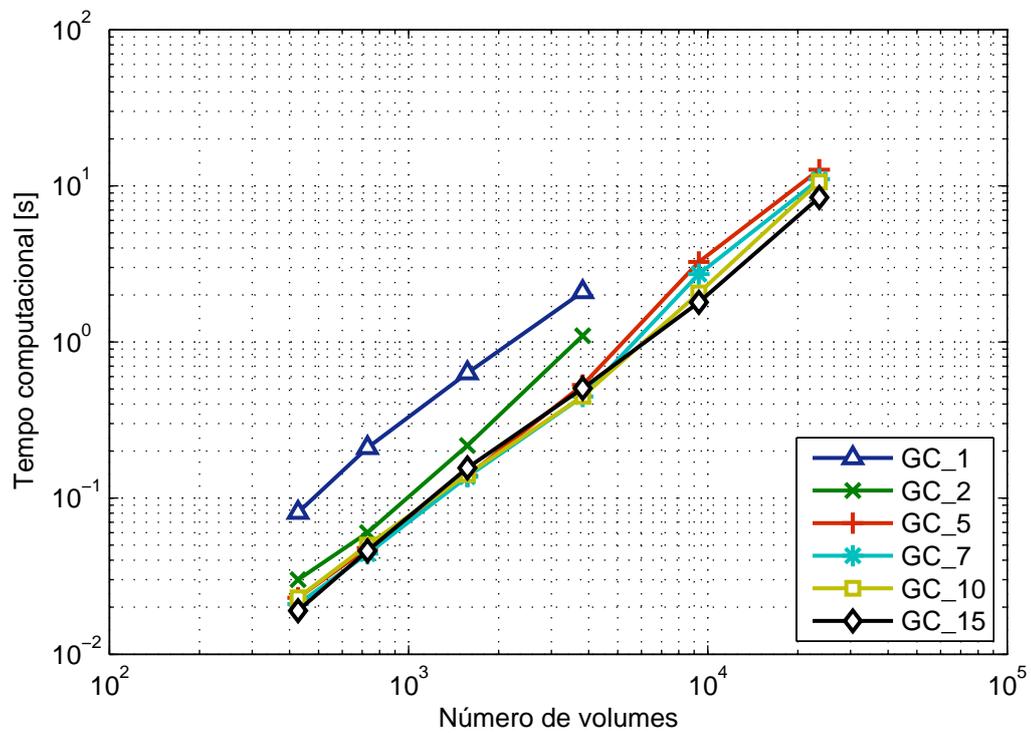


Figura 6.18: Resultados do multigrid com método do Gradiente Conjugado e esquema totalmente implícito

tem número de iterações menor. Nas figuras 6.19 e 6.20 estão os resultados do multigrid com GMRES de dimensão dois e 10, respectivamente. Nota-se algumas peculiaridades nesses gráficos. Com dimensão dois, o caso com apenas uma iteração por nível não convergiu na malha de 9331 elementos, embora tenha convergido na de 23569 elementos. Além disso, com mais de duas iterações há aparentemente uma mudança da inclinação da reta que interpola os pontos a partir da malha de 3820 elementos. Com dimensão 10 os pontos acabaram descrevendo curvas ao invés de ficarem alinhados.

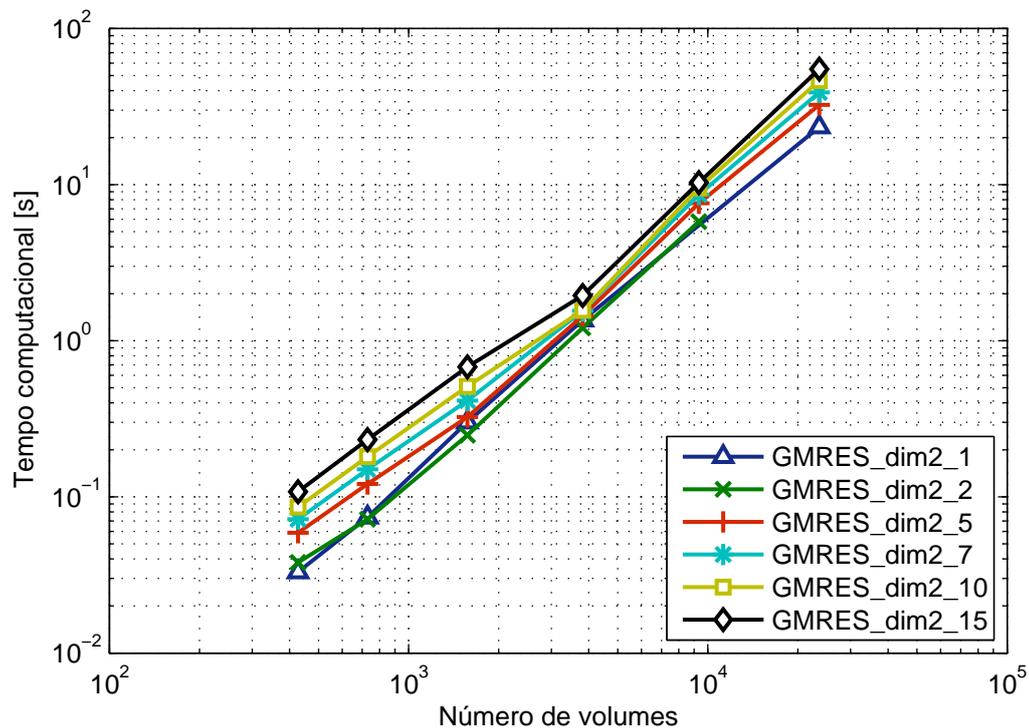


Figura 6.19: Resultados do multigrid com método GMRES de dimensão dois e esquema totalmente implícito

No multigrid com método MINRES, de acordo com a figura 6.21 ficou bastante clara a relação entre a robustez do método e o número de iterações. À medida que se aumenta o número de iterações o multigrid fica mais lento, porém mais robusto. Somente com 15 iterações houve convergência em todas as malhas.

Na figura 6.22 estão os resultados dos métodos aplicando-se o número de iterações por nível ótimo dentre aqueles estudados. Está bastante clara a superioridade do método do Gradiente Conjugado em relação aos demais. Já os resultados do GMRES apresentaram certa oscilação, de modo que não é possível distinguir uma dimensão do subespaço de Krylov recomendável. Inclusive, o nível de dispersão dos resultados não só com o GMRES, mas também com outros métodos, inviabilizou o cálculo do grau de complexidade dos métodos.

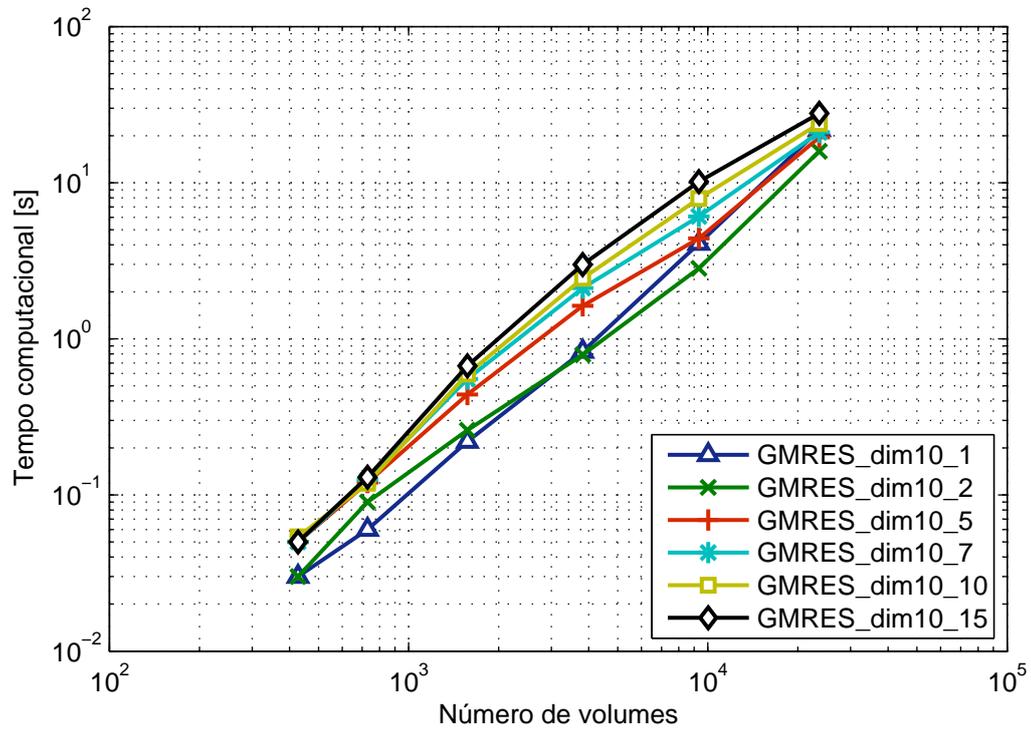


Figura 6.20: Resultados do multigrid com método GMRES de dimensão 10 e esquema totalmente implícito

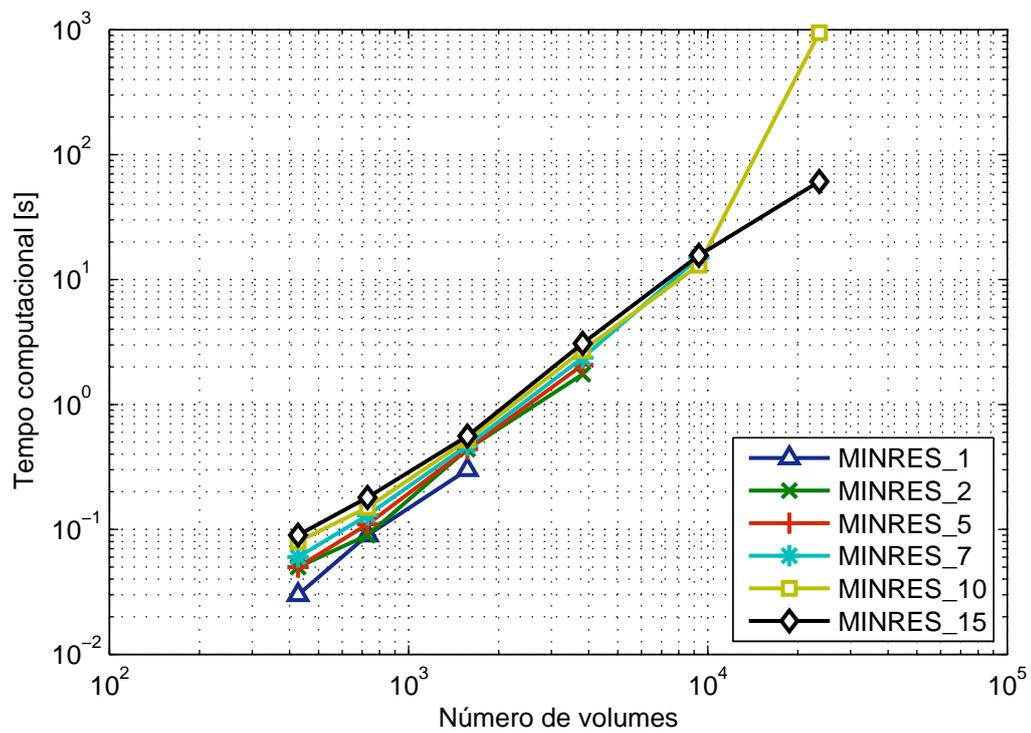


Figura 6.21: Resultados do multigrid com método MINRES e esquema totalmente implícito

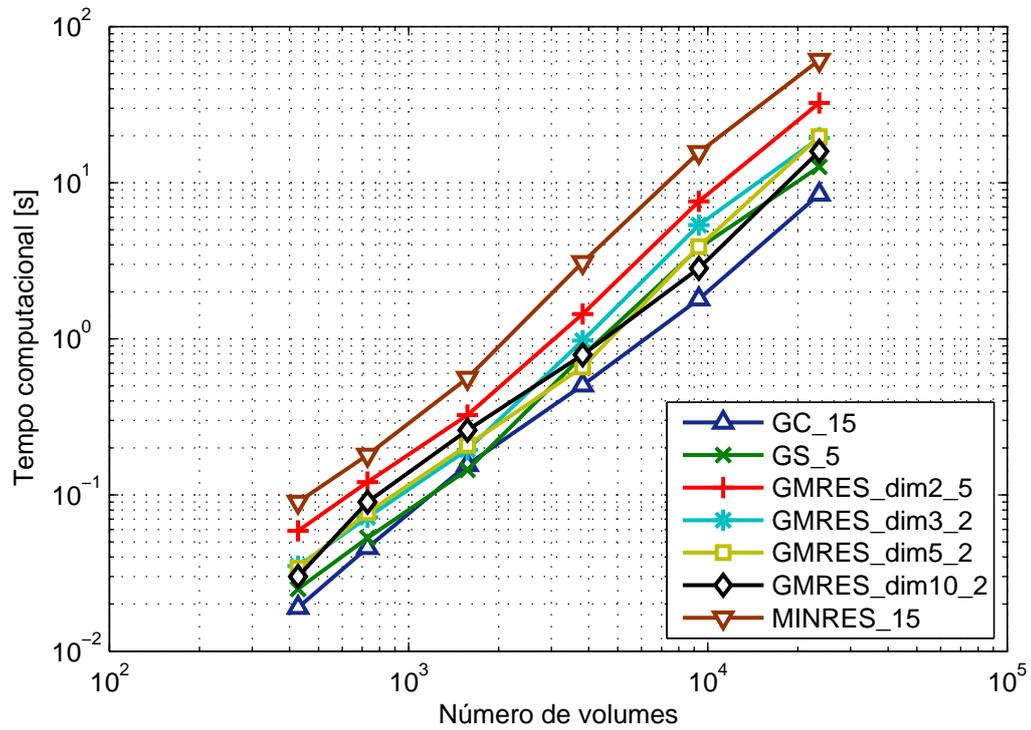


Figura 6.22: Resultados dos diferentes métodos com número de iterações ótimo e esquema totalmente implícito

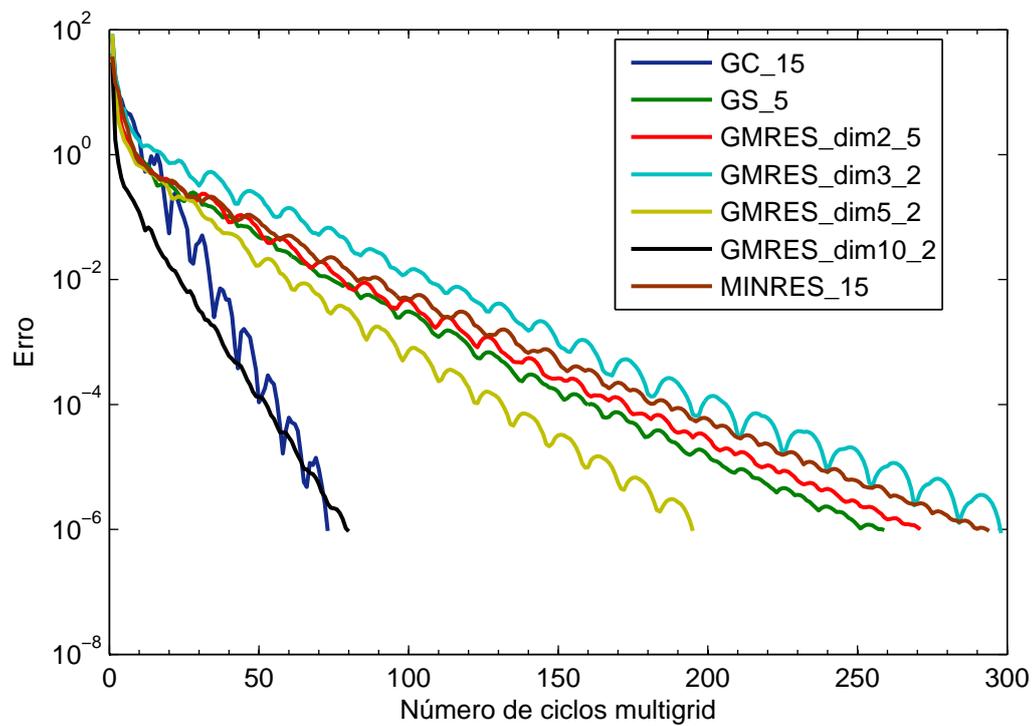


Figura 6.23: Decaimento do erro em função do número de ciclos multigrid

Os resultados na malha mais refinada do decaimento do erro, definido pela equação (6.1), em função do número de ciclos multigrid executados constam na figura 6.23. Em contrapartida aos resultados da figura 6.9, relativos à malha regular, houve flutuações no decaimento do erro em todos os métodos avaliados. É interessante notar que no multigrid com método GMRES tanto o número de ciclos multigrid quanto as flutuações diminuíram à medida que se aumenta o número de iterações ou a dimensão do subespaço de Krylov. Isso é um indicativo de que a escolha do GMRES como *solver* do multigrid é interessante quando se deseja obter a solução de sistemas lineares relativamente difíceis de se resolver. Caso haja problemas de convergência, basta aumentar o número de iterações ou a dimensão do subespaço.

6.3 MALHA TRIDIMENSIONAL COM DISTORÇÃO ALEATÓRIA

Esta é a única malha tridimensional avaliada nesse trabalho. Ela foi construída a partir de uma malha predominantemente regular, composta de elementos de faces octogonais, hexagonais e quadrangulares aplicando-se deslocamentos aleatórios em seus vértices. Na figura 6.24 está ilustrado um dos elementos da malha. A distorção deve provocar anisotropia nos coeficientes da matriz, podendo dificultar a solução dos sistemas lineares. Adicionalmente, os elementos da malha tridimensional apresentam maior número de vizinhos, resultando em matrizes de menor esparsidade. No esquema totalmente implícito, em particular, o fato de o estêncil apresentar um nível de vizinhança a mais deve resultar numa redução da esparsidade ainda maior. O tensor de permeabilidade utilizado é o tensor identidade.

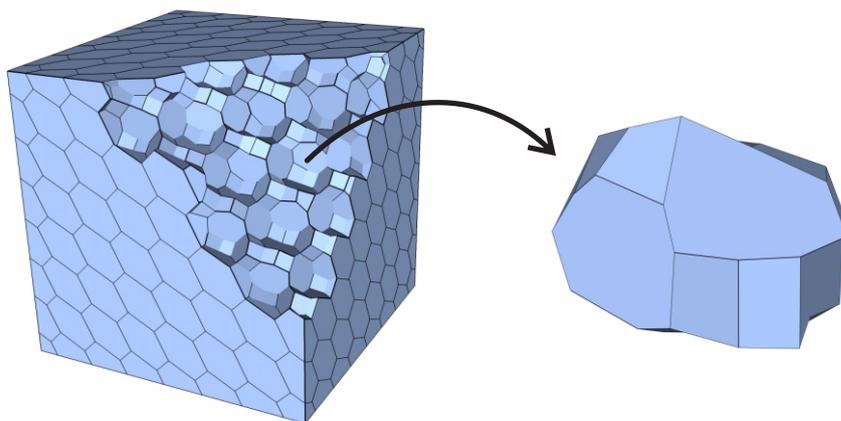


Figura 6.24: Elemento da malha tridimensional com distorção aleatória

6.3.1 ESQUEMA SEMI-IMPLÍCITO

Os resultados do multigrid com método do Gradiente Conjugado não foram favoráveis para apenas uma iteração por nível, de acordo com a figura 6.25. Houve convergência apenas nas duas primeiras malhas e o tempo computacional correspondente ficou bem acima dos outros casos. Para duas iterações, embora tenha havido convergência em todas as malhas, os tempos computacionais também ficaram bem acima das outras configurações. Acima de cinco iterações por nível os resultados ficaram próximos.

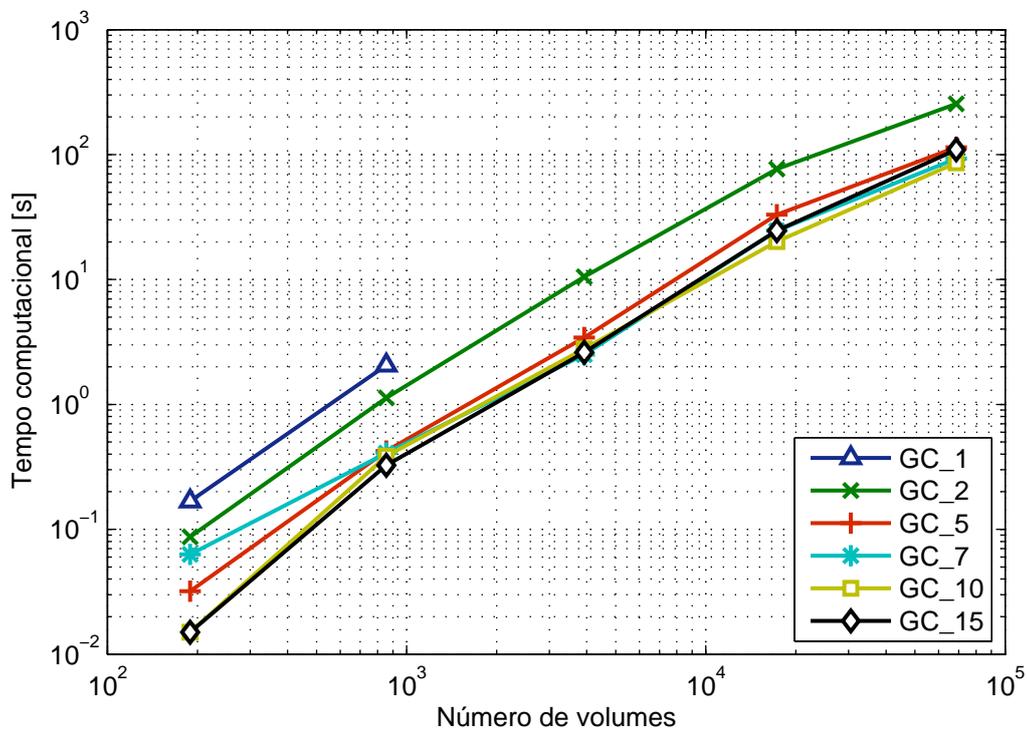


Figura 6.25: Resultados do multigrid com método do Gradiente Conjugado e esquema semi-implícito

Com base na figura 6.26, verifica-se que o desempenho do Gauss-Seidel foi superior para poucas iterações por nível: uma ou duas. Aumentar o número de iterações faz com que o tempo computacional também aumente. É interessante notar que nesse caso, assim como no da malha *stretched*, também há, devido ao formato dos elementos da malha, anisotropia nos coeficientes da matriz do sistema linear. Entretanto, a anisotropia aqui não favorece uma direção específica para a propagação da informação, fato que deve ser o provável motivo do porquê o perfil dos resultados ter ficado muito mais parecido com o da malha regular do que com o da malha *stretched*.

O multigrid com método GMRES, assim como o com o de Gauss-Seidel, apresentou melhor desempenho para poucas iterações por nível. Porém, com apenas uma iteração houve

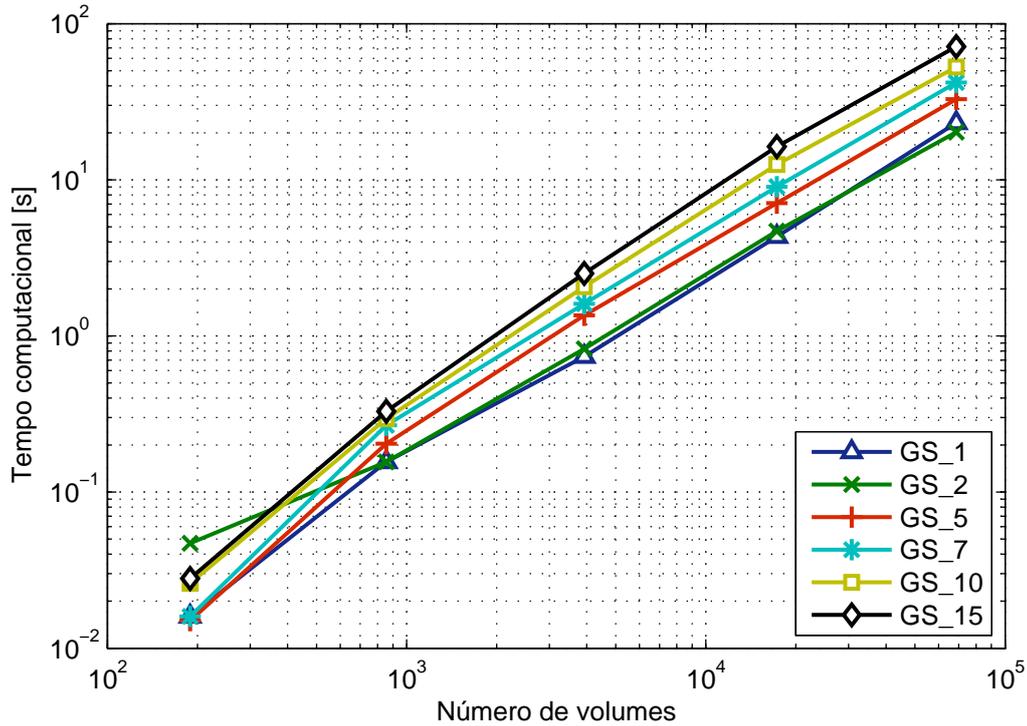


Figura 6.26: Resultados do multigrid com método de Gauss-Seidel e esquema semi-implícito

problemas de convergência. Para subespaço de Krylov de dimensão dois só houve convergência nas duas primeiras malhas. Aumentando-se então a dimensão do subespaço o método passou a convergir nas três primeiras, nas quatro e, com dimensão 10, em todas. Na figura 6.27 estão ilustrados os resultados para subespaço de dimensão cinco. Do mesmo modo que o GMRES, o MINRES também apresentou problemas de convergência ao se utilizarem poucas iterações por nível. Para apenas uma iteração só houve convergência nas duas primeiras malhas, ao passo que para duas o método só divergiu na malha mais refinada. Os melhores tempos computacionais do MINRES se deram especificando-se cinco iterações por nível.

Na figura 6.28 estão os resultados dos métodos especificando-se número de iterações ótimo para cada. Claramente o multigrid com Gauss-Seidel foi bem superior aos demais. Seu tempo computacional na malha refinada, por exemplo, foi da ordem de vinte vezes menor que o do MINRES, que apresentou o pior desempenho. Já o GMRES foi melhor para as maiores dimensões do subespaço de Krylov avaliadas. Na tabela 6.5 estão os graus de complexidade observados para os diferentes métodos. Para a determinação desses valores foram eliminados os pontos da malha mais grosseira devido à menor exatidão na medição dos tempos computacionais. Novamente o grau de complexidade do multigrid com Gauss-Seidel foi melhor. Entretanto há de se ressaltar que alguns dos métodos não tiveram, no gráfico da fi-

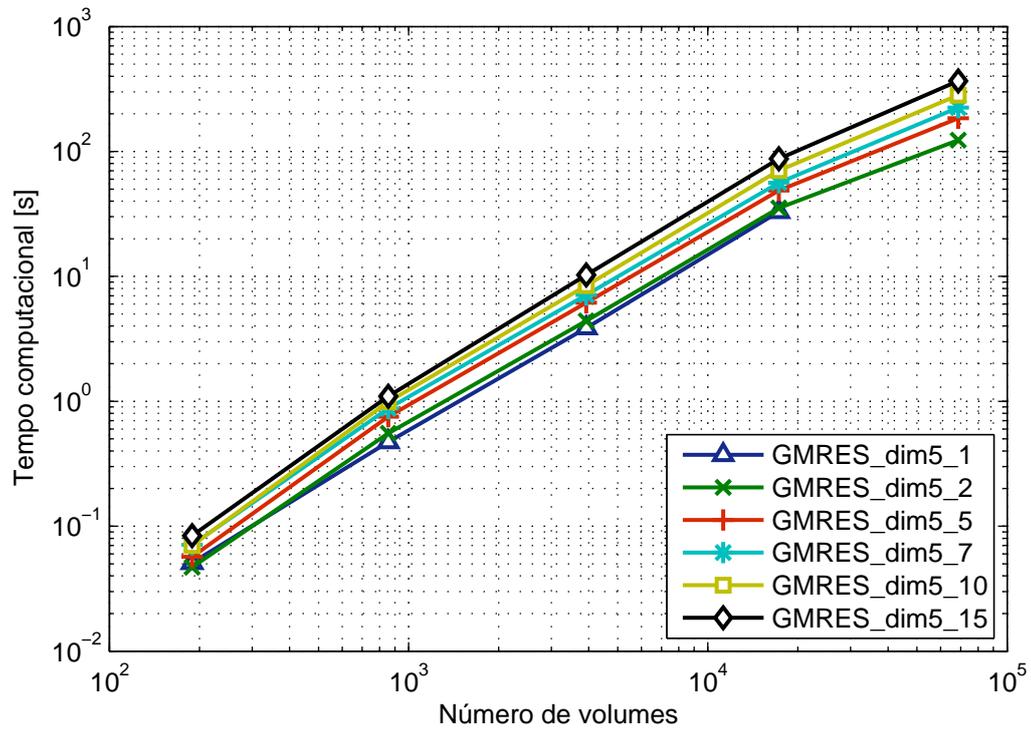


Figura 6.27: Resultados do multigrid com método GMRES de dimensão cinco e esquema semi-implícito

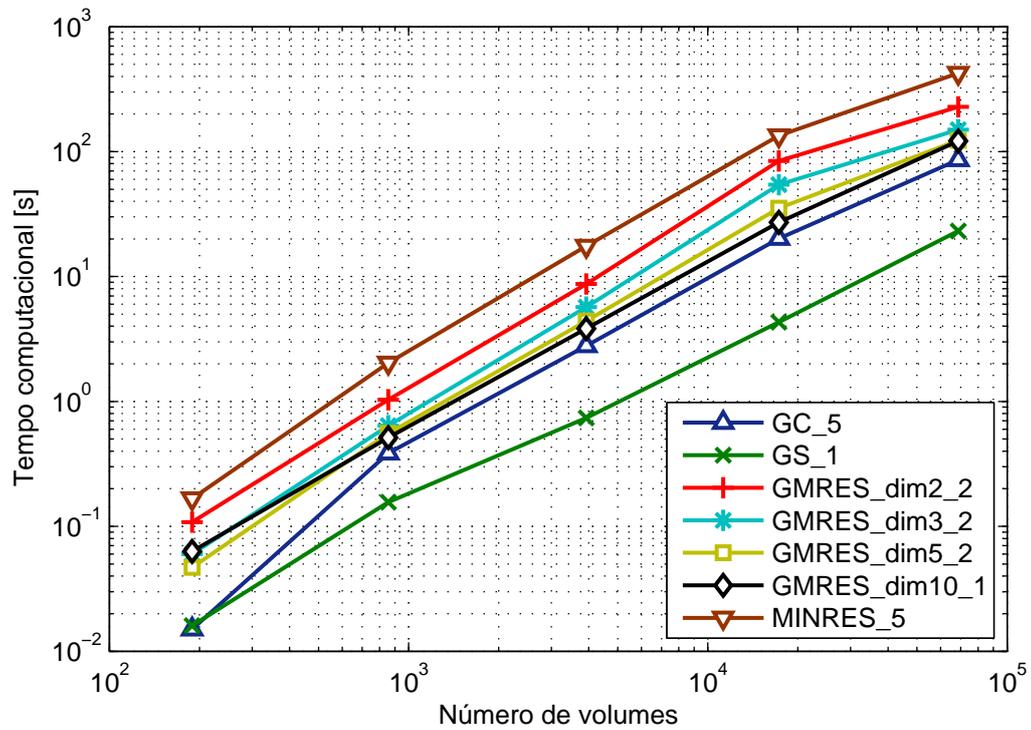


Figura 6.28: Resultados dos diferentes métodos com número de iterações ótimo e esquema semi-implícito

gura 6.28, seus pontos bem aliados e, por isso, seus resultados devem ser analisados com cautela.

Tabela 6.5: Complexidade do multigrid com diversos métodos e esquema semi-implícito

Método	Complexidade
Gradiente Conjugado, 10 iter.	1,2440
Gauss-Seidel, 1 iter.	1,1447
GMRES dim. 2, 2 iter.	1,2625
GMRES dim. 3, 2 iter.	1,2576
GMRES dim. 5, 2 iter.	1,2515
GMRES dim. 10, 1 iter.	1,2549
MINRES, 5 iter.	1,2363

6.3.2 ESQUEMA TOTALMENTE IMPLÍCITO

Os resultados do esquema totalmente implícito foram totalmente análogos aos obtidos com o esquema semi-implícito. O multigrid com método do Gradiente Conjugado com uma iteração por nível apenas convergiu nas duas malhas mais grosseiras e, com duas iterações por nível, apresentou tempo computacional bem acima das demais configurações. Com o método de Gauss-Seidel o desempenho foi novamente melhor aplicando-se apenas uma iteração em cada nível. Já o GMRES apresentou dificuldades de convergência para poucas iterações por nível, o que foi se atenuando a medida que se aumentou a dimensão do subespaço de Krylov. Quanto ao MINRES, não houve convergência nas malhas mais refinadas para poucas iterações por nível. Na figura 6.29 estão os resultados dos métodos para número de iterações ótimo.

Novamente o multigrid com Gauss-Seidel se mostrou superior aos demais. O MINRES apresentou o pior desempenho e o GMRES foi tão melhor quanto maior foi a dimensão do espaço de Krylov. Na tabela 6.6 estão os graus de complexidade estimados para os métodos. Nesse cálculo não foi utilizado o resultado da malha mais refinada por ele não seguir, sobretudo para o método GMRES, o padrão definido pelos outros pontos. Verifica-se novamente que o método de Gauss-Seidel apresentou o menor grau de complexidade e que o grau de complexidade do GMRES foi tanto menor quanto maior foi a dimensão de seu subespaço. Já o multigrid com MINRES, além de requerer mais tempo para a solução dos sistemas lineares, também apresentou o maior grau de complexidade.

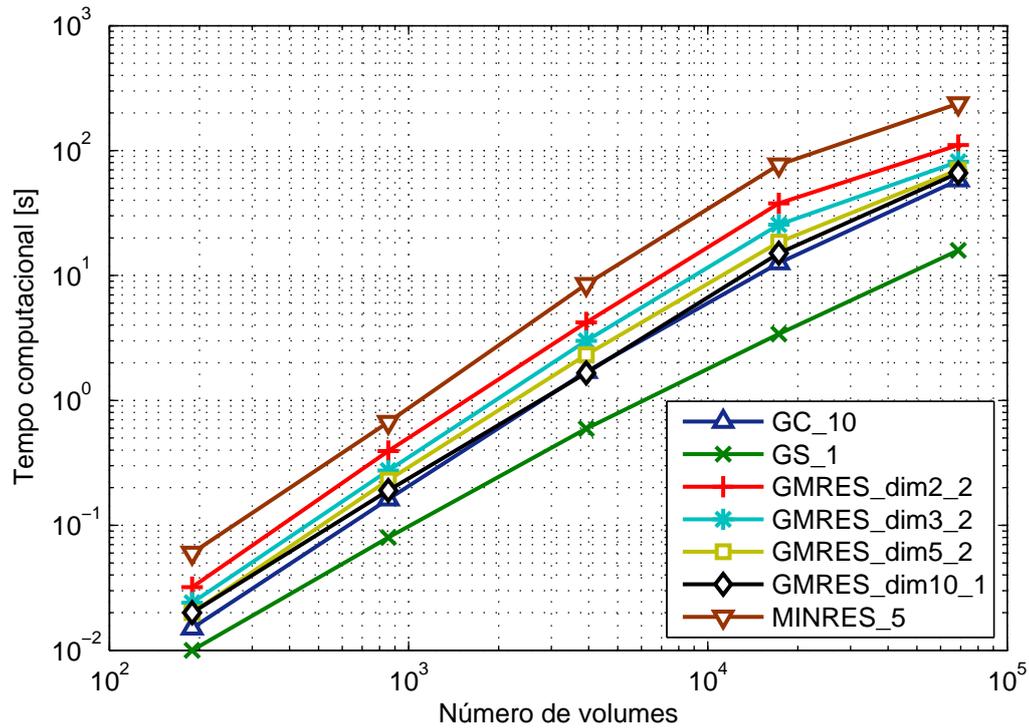


Figura 6.29: Resultados dos diferentes métodos com número de iterações ótimo e esquema totalmente implícito

Na figura 6.30 estão os resultados na malha mais refinada do decaimento do erro em função do número de ciclos multigrid executados. Observa-se que em praticamente todos os casos o erro decaiu de maneira estável, sem flutuações significativas. Há de se ressaltar que o número de iterações realizadas em cada nível do multigrid é superior ao utilizado no caso da malha regular bidimensional, o que provavelmente justifica o fato de se ter menores flutuações em relação àquele caso.

Tabela 6.6: Complexidade do multigrid com diversos métodos e esquema totalmente implícito

Método	Complexidade
Gradiente Conjugado, 10 iter.	1,4965
Gauss-Seidel, 1 iter.	1,2943
GMRES dim. 2, 2 iter.	1,5658
GMRES dim. 3, 2 iter.	1,5468
GMRES dim. 5, 2 iter.	1,5133
GMRES dim. 10, 1 iter.	1,4624
MINRES, 5 iter.	1,5936

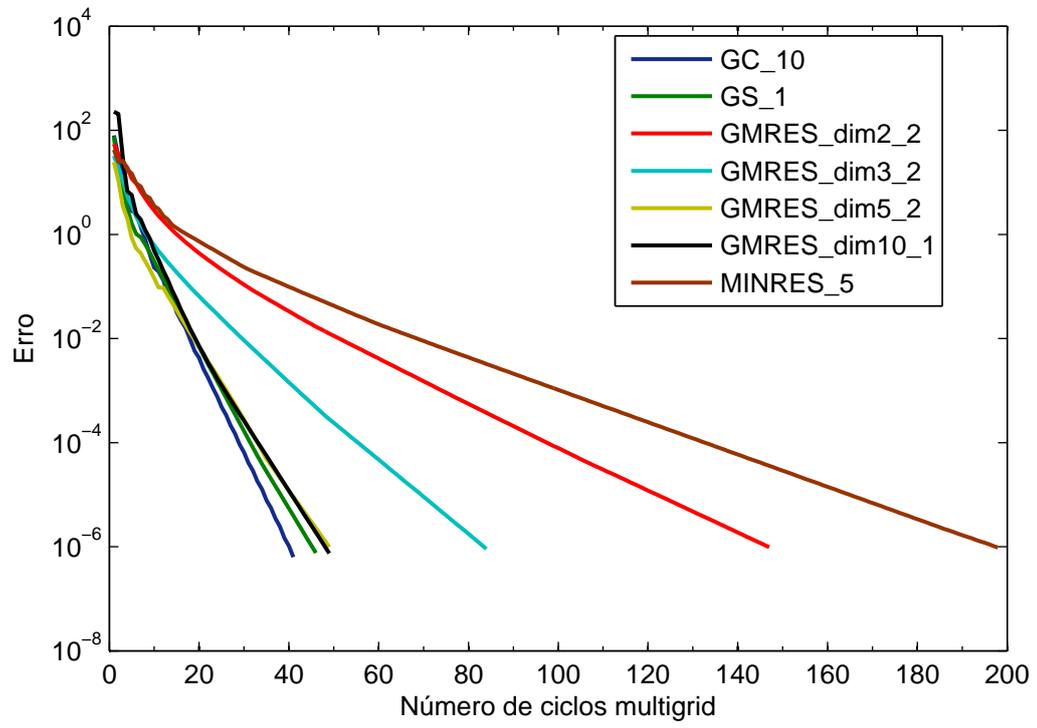


Figura 6.30: Decaimento do erro em função do número de ciclos multigrid executados

7 COMENTÁRIOS E CONCLUSÕES

O objetivo principal desse trabalho foi o estudo do multigrid ACM na solução de problemas oriundos da simulação numérica de reservatórios de petróleo. Desenvolveram-se a modelagem matemática e a discretização em malhas poligonais e poliédricas de um problema de escoamento monofásico em meio poroso. Dois esquemas de aproximação dos fluxos nas faces dos volumes de controle foram avaliados: o semi-implícito e o totalmente implícito. No primeiro os gradientes são aproximados de maneira explícita valendo-se de um método de reconstrução que se utiliza do campo escalar disponível. O estêncil resultante apresenta apenas um nível de vizinhança e a matriz correspondente é diagonal dominante. No outro, as expressões do método de reconstrução são inseridas nas equações do fluxo de massa, de sorte que estes são calculados utilizando-se apenas termos implícitos. O estêncil agora abrange dois níveis de vizinhança, tornando a matriz menos esparsa e possivelmente não diagonal dominante.

De forma geral, as configurações que apresentaram melhor desempenho no esquema semi-implícito também foram as melhores no totalmente implícito. Em malhas regulares verificou-se que para melhor *performance* é recomendável a execução de um número menor de iterações. Para o método de Gauss-Seidel o ideal é apenas uma iteração; para o do Gradiente Conjugado, duas; para o GMRES, uma e dimensão do subespaço dois; e para o MINRES, duas. Não se recomenda apenas uma iteração nos métodos do Gradiente Conjugado e MINRES devido a não convergência em diversas malhas. Comparando-se os métodos entre si, o multigrid com Gauss-Seidel e apenas uma iteração foi o melhor. Além de ele apresentar tempos computacionais inferiores aos outros, seu grau de complexidade foi o menor.

Na malha com grande razão de aspecto (*stretched*), houve vários casos que não convergiram. Para cada *solver* há um número mínimo de iterações que garante a convergência. No multigrid com Gradiente Conjugado esse número é de aproximadamente cinco e, para valores maiores, não há diferença significativa na *performance*. Com Gauss-Seidel são necessárias aproximadamente 15 iterações no esquema semi-implícito e cinco no totalmente implícito. Valores acima daqueles que garantem a convergência acabam por aumentar os tempos computacionais. Em relação ao GMRES, quanto maior for a dimensão do subespaço de Krylov, menor pode ser o número de iterações para que haja convergência. Recomenda-se em casos anisotrópicos como o da malha *stretched* utilizar uma dimensão maior do subespaço para que seja possível manter um pequeno número de iterações. Por fim, o MINRES foi aquele que apresentou os piores resultados. Além de ser significativamente mais lento que

os demais, ele não convergiu na maioria das malhas. O método de melhor desempenho foi o do Gradiente Conjugado com número de iterações da ordem de 10.

A malha com distorção aleatória foi o único tipo de malha tridimensional avaliado. Nela há anisotropia dos coeficientes, mas não há uma direção privilegiada para a propagação da informação. O comportamento do multigrid com Gauss-Seidel neste caso foi semelhante ao observado nas malhas regulares. Todos os casos convergiram e a melhor configuração foi aquela com apenas uma iteração por nível. Já o multigrid com método do Gradiente Conjugado teve resultados semelhantes aos da malha *stretched*: são necessárias aproximadamente cinco iterações por nível para tornar o método robusto e, para valores maiores, não há diferença significativa no desempenho. Quanto ao GMRES, não foram necessárias muitas iterações para que houvesse convergência nos casos com as menores dimensões do subespaço de Krylov. Entretanto, em contrapartida aos resultados das malhas regulares, os tempos computacionais foram menores para os casos com maior dimensão do subespaço. Comparando-se todos os métodos entre si, o multigrid com Gauss-Seidel e uma iteração foi superior, ao passo que com MINRES ele apresentou seu pior desempenho.

Conclui-se então desse trabalho que a escolha do *solver* a ser empregado no multigrid deve estar baseada na natureza do problema a ser resolvido. Caso a malha seja regular e o meio isotrópico, é preferível a utilização do método de Gauss-Seidel com apenas uma iteração em cada nível. Esse método também é recomendável nos casos em que, embora haja anisotropia nos coeficientes, não há uma direção preferencial de propagação de informação. Se essa direção de fato existe, então o método do Gradiente Conjugado com número de iterações da ordem de 10 é a melhor escolha. O multigrid com GMRES, embora em nenhum dos casos tenha sido o melhor, apresentou *performance* aceitável. Recomenda-se que nos casos isotrópicos sejam utilizados número de iterações e dimensão do subespaço pequenos. Caso haja anisotropia, é preferível manter o número de iterações pequeno, mas aumentar a dimensão do subespaço. Por fim, não é aconselhável utilizar o método MINRES no multigrid. Além de apresentar tempos computacionais bem acima dos demais, essa configuração se mostrou muito pouco robusta.

7.1 SUGESTÕES PARA TRABALHOS FUTUROS

Propõe-se para trabalhos futuros:

- (i) Implementar outros esquemas de aglomeração;

- (ii) Avaliar os efeitos do ciclo multigrid, do número médio de volumes aglomerados e do número de volumes na malha mais grosseira sobre os resultados obtidos;
- (iii) Implementar no multigrid outros *solvers* iterativos, como, por exemplo, a fatoração LU incompleta e o Gradiente Bi-conjugado;
- (iv) Empregar o multigrid em outras classes de problemas. Recomenda-se avaliá-lo na solução do sistema linear do problema da cavidade com tampa deslizante e formulação acoplada, no qual pode haver zeros na diagonal da matriz. Também é interessante avaliá-lo na solução de sistemas lineares provenientes do emprego do método de Newton na solução de sistemas de equações não-lineares.

REFERÊNCIAS

- BEJAN, A. *Convection Heat Transfer*. 3^a. ed. Durham, North Carolina: John Wiley & Sons, 2004.
- BRIGGS, W. L.; HENSON, V. E.; MCCORMICK, S. F. *A Multigrid Tutorial*. Philadelphia, PA: SIAM, 2000.
- DEMME, J. W. *Applied Numerical Linear Algebra*. Cambridge, MA: Society for Industrial and Applied Mathematics, 1996.
- HUTCHINSON, B. R.; RAITHBY, G. D. A multigrid method based on the additive correction strategy. *Numerical Heat Transfer*, v. 9, p. 511–537, 1986.
- KAVIANY, M. *Principles of Heat Transfer in Porous Media*. Ann Arbor, Michigan: Springer, 1995.
- KELLER, S. C. *O Método Multigrid de Correções Aditivas para a Solução Numérica Acoplada das Equações de Navier-Stokes com Malhas Não-Estruturadas*. Tese (Doutorado) — Universidade Federal de Santa Catarina, 2007.
- MALISKA, C. R. *Transferência de Calor e Mecânica dos Fluidos Computacional*. Rio de Janeiro, RJ: LTC Editora, 2004.
- MALISKA, C. R.; SILVA, A. F. C. da; HURTADO, F. S. V.; RIBEIRO, G. G.; JR, A. A. V. B. P.; GREIN, E. A.; CERBATO, G. *Introdução às malhas poliédricas e ao método baseado na reconstrução explícita do gradiente, Relatório Técnico SINMEC/SiGeR 01*. Florianópolis, SC, 2012.
- PEACEMAN, D. W. *Fundamentals of numerical reservoir simulation*. Houston, Texas: Elsevier Scientific Publishing Company, 1977.
- PESCADOR JR, A. A. V. B. *Determinação de Parâmetros Ótimos para o Método Multigrid de Correções Aditivas*. 2010. Trabalho de Curso, Universidade Federal de Santa Catarina, Departamento de Engenharia Mecânica.
- SAAD, Y. *Iterative Methods for Sparse Linear Systems*. 2^a. ed. Minneapolis, Minnesota: Society for Industrial and Applied Mathematics, 2000.
- STÜBEN, K. A review of algebraic multigrid. *Journal of Computational and Applied Mathematics*, v. 128, p. 281–309, 2001.
- STRANG, G. *Linear Algebra and its Application*. 3^a. ed. Cambridge, MA: Harcourt Brace Jovanovich, 1988.
- STRANG, G. *Lecture 18: Krylov Methods / Multigrid Continued, Mathematical Methods for Engineers II, Massachusetts Institute of Technology*. 2006. Vídeo aula.
- STRANG, G. *Lecture 19: Conjugate Gradient Method, Mathematical Methods for Engineers II, Massachusetts Institute of Technology*. 2006. Vídeo-aula.
- ZAMBALDI, M. C. Notas de aula. Disciplina de H Álgebra Linear, Universidade Federal de Santa Catarina. 1^o semestre 2009.