



Proceedings of the XXVII Iberian Latin American Congress on  
Computational Methods in Engineering  
September 3 to 6, 2006 - Belém, Pará - BRAZIL.

## **EFEITO DA RAZÃO DE ASPECTO DE MALHA SOBRE O TEMPO DE CPU PARA A EQUAÇÃO DE LAPLACE 2D RESOLVIDA COM O MÉTODO MULTIGRID**

**Márcio Augusto Villela Pinto**

*marciovp@demec.ufpr.br*

Departamento de Matemática e Estatística, Universidade Estadual de Ponta Grossa  
Ponta Grossa, PR – Brasil

**Carlos Henrique Marchi**

*marchi@demec.ufpr.br*

Departamento de Engenharia Mecânica, Universidade Federal do Paraná (UFPR)  
Curitiba, PR – Brasil

**Resumo.** *Sobre o tempo de CPU necessário para resolver um problema, verifica-se o efeito causado por: diversos valores de razão de aspecto de malha, vários tamanhos de malha e quatro algoritmos multigrid. O problema considerado é a equação de Laplace bidimensional com condições de contorno de Dirichlet. O método de diferenças finitas é usado para discretizar a equação diferencial. Os sistemas de equações algébricas são resolvidos com o método de Gauss-Seidel, associado ao método multigrid geométrico com esquema de correção (CS) e ciclo V. Quatro tipos de algoritmo de engrossamento são considerados: a) engrossamento padrão, b) semi-engrossamento, c) engrossamento padrão seguido de semi-engrossamento, e d) semi-engrossamento seguido de engrossamento padrão. Verificou-se que o algoritmo do tipo semi-engrossamento seguido de engrossamento padrão é o mais rápido entre os quatro testados.*

**Palavras-chave:** *Solver, CFD, diferenças finitas, otimização, métodos numéricos.*

## 1. INTRODUÇÃO

A discretização de modelos matemáticos de problemas de dinâmica dos fluidos computacional resulta em sistemas de equações algébricas do tipo

$$\mathbf{A}\vec{v} = \vec{f}, \quad (1)$$

onde  $\mathbf{A}$  é uma matriz quadrada,  $\vec{f}$  é o vetor independente e  $\vec{v}$  é o vetor de incógnitas. Estes modelos representam fenômenos físicos que envolvem fluidos em movimento, com ou sem troca de calor (Fortuna, 2000; Maliska, 2004). A estrutura da matriz  $\mathbf{A}$  depende do método usado para discretizar o modelo matemático.

Um método muito usado é o método de diferenças finitas (Golub e Ortega, 1992; Tannehill et al., 1997), onde, em problemas bidimensionais, o domínio  $(x, y) \in \mathcal{R}^2 : 0 \leq x, y \leq 1$  é particionado em um número de incógnitas (ou número de pontos), dado por

$$N = N_x N_y, \quad (2)$$

onde  $N_x$  e  $N_y$  são o número de incógnitas nas direções coordenadas  $x$  e  $y$ , respectivamente.

Isto introduz uma malha com os pontos

$$(x_i, y_j) = ((i-1)h_x, (j-1)h_y), \text{ com } h_x = \frac{1}{N_x - 1} \text{ e } h_y = \frac{1}{N_y - 1}, \quad (3)$$

onde  $i = 1, \dots, N_x$ ,  $j = 1, \dots, N_y$  e  $h_x$  e  $h_y$  são os comprimentos de cada elemento nas direções coordenadas  $x$  e  $y$ , respectivamente. Neste caso, estabelece-se uma malha com elementos de tamanho  $h_x$  por  $h_y$  que se denota por  $\Omega^h$ , com razão de aspecto ( $RA$ ) definida por

$$RA = \frac{h_x}{h_y}. \quad (4)$$

Se  $RA = 1$ , o problema chama-se isotrópico; caso contrário, anisotrópico (Briggs et al., 2000; Dendy et al., 1989). Quanto mais distinta a razão de aspecto ( $RA$ ) está da unidade, ou seja,  $0 < RA \ll 1$  ou  $RA \gg 1$ , mais o desempenho do método *multigrid* se deteriora (Wesseling e Oosterlee, 2001); com isso a taxa de convergência piora, podendo até mesmo ocorrer divergência (Larsson et al., 2005). Este tipo de anisotropia é muito comum em problemas práticos de engenharia, como em problemas de camada limite, onde a razão de aspecto ( $RA$ ) pode ser da ordem de  $10^3$ ,  $10^4$  ou mais (Wesseling e Oosterlee, 2001); por isto a importância de se estudar algoritmos eficientes para razões de aspecto distintas da unidade.

Várias técnicas numéricas têm sido estudadas para resolver o sistema dado pela Eq. (1) com o menor custo computacional e a solução a mais próxima possível da exata (sem erros de iteração; Ferziger e Peric, 1999). A resolução por métodos diretos não é recomendável, visto que na prática, a matriz dos coeficientes é muito grande e o custo da inversão da matriz é alto (Golub e Van Loan, 1984). Os métodos iterativos são mais adequados para problemas de grande porte (Burden e Faires, 1997). O método do gradiente conjugado (Burden e Faires, 1997; Golub e Ortega, 1992), introduzido por Hestenes e Stiefel (1952), usa técnicas que são

mais específicas para geometrias simples e é um método sensível ao condicionamento da matriz.

O método *multigrid*, proposto originalmente por Fedorenko (1964), é atualmente um método numérico muito usado para resolver iterativamente sistemas de equações do tipo da Eq. (1). A idéia básica é usar um conjunto de malhas e executar alternativamente iterações em cada nível de malha e soluções aproximadas desta equação em malhas mais grossas (Briggs et al., 2000). São usados operadores para transferir informações da malha fina para a malha imediatamente mais grossa (restrição); e da malha grossa para a malha imediatamente mais fina (prolongação). Briggs et al. (2000) trabalharam com a razão de engrossamento  $r = 2$  afirmando ser uma prática universal e que  $r \neq 2$  não traz vantagens. Em cada malha, o sistema de equações é resolvido com um método iterativo com propriedades de reduzir rapidamente os erros oscilatórios (propriedades de suavização). O objetivo do método *multigrid* é acelerar a convergência para reduzir o tempo de CPU necessário para resolver a Eq. (1). Em Wesseling e Oosterlee (2001) são apontados vários desafios para o *multigrid* geométrico, como: solução das equações de Navier-Stokes, problemas com perturbações singulares, problemas de camada limite onde aparecem malhas fortemente alongadas (malhas com razão de aspecto muito distinta da unidade), a paralelização de algoritmos, etc.

Problemas anisotrópicos (anisotropia física e geométrica) foram abordados por Mulder (1989), Dendy et al. (1989), Naik e Van Rosendale (1993) e Radespiel e Swanson (1995), fazendo-se uso de semi-engrossamento (engrossamento em apenas uma das direções) e suas variantes. Em Sivaloganatan (1991), Oosterlee e Wesseling (1993), Sockol (1993) e Pierce e Alonso (1998) os problemas anisotrópicos foram discutidos pelo ponto de vista de variações e adaptações dos suavizadores. Problemas anisotrópicos comparando a técnica de semi-engrossamento com a técnica de suavizadores foram abordados por Montero et al. (2001) e Larsson et al. (2005). Em Zhang (2002), são abordados problemas anisotrópicos acoplando-se a técnica de semi-engrossamento seguida de engrossamento padrão (engrossamento em ambas as direções) intitulado como “partial semicoarsening”, denominado no presente artigo por SE-EP. Comparações foram feitas para a equação de Poisson bidimensional com a técnica que se chama de engrossamento padrão (*full coarsening*), abreviada no presente artigo por EP. Zhang (2002) comparou estas técnicas para as razões de aspecto  $RA = 2, 4, 8$  e  $16$  (razões de aspecto modestas). Zhang constatou que o algoritmo SE-EP com os suavizadores red-black e four-color Gauss-Seidel são eficientes para as anisotropias estudadas.

Neste trabalho, quatro algoritmos baseados no método *multigrid* são comparados em problemas anisotrópicos: engrossamento padrão (EP), engrossamento padrão seguido de semi-engrossamento (EP-SE), semi-engrossamento (SE) e semi-engrossamento seguido de engrossamento padrão (SE-EP). São considerados os seguintes parâmetros: número de iterações internas, número de incógnitas e razões de aspecto  $RA = 1/1024, 2, 16, 128, 1024$  e  $8192$ . Deve-se lembrar que, para  $RA = 1$ , tem-se um problema isotrópico.

O objetivo deste artigo é verificar o efeito desses parâmetros sobre o tempo de CPU para o método *multigrid*. O algoritmo *multigrid* adotado é o Esquema de Correção (*Correction Scheme*) com ciclo  $V$  e operadores de restrição (injeção) e prolongação (interpolação bilinear) para a razão de engrossamento padrão ( $r = 2$ ), como proposto em Wesseling (1992). Os resultados são comparados com os obtidos na bibliografia. O modelo matemático considerado neste trabalho envolve um problema linear de transferência de calor bidimensional, isto é, equação de Laplace, com condições de contorno de Dirichlet.

Este artigo está organizado da seguinte forma: na seção 2 apresenta-se uma visão geral do método *multigrid* e os quatro algoritmos que são comparados. Na seção 3, são definidos os modelos matemático e numérico. Nas seções 4 e 5 são descritos os experimentos numéricos com seus resultados e a conclusão do trabalho.

## 2. O MÉTODO MULTIGRID

Para reduzir o erro de discretização, malhas muito refinadas são necessárias a fim de se resolver problemas de mecânica dos fluidos e transferência de calor. Isso gera sistemas de equações muito grandes. A resolução destes problemas através de métodos numéricos requer um custo computacional demasiadamente alto e muitas vezes inviável devido ao grande número de equações a serem resolvidas em cada passo iterativo na solução do sistema dado pela Eq. (1). Um método alternativo usado para melhorar a taxa de convergência nestes problemas é o método *multigrid* (Briggs et al., 2000), que acelera consideravelmente a resolução dos sistemas de equações envolvidos nos problemas. Métodos *multigrid* são métodos iterativos para solução de sistemas lineares, sendo, portanto, dependentes da estimativa inicial atribuída às incógnitas do problema.

Uma técnica eficiente usada para aliviar as fortes oscilações do resíduo em cada malha, que é definido por

$$\vec{R} = \vec{f} - \mathbf{A}\vec{v}, \quad (5)$$

é suavizar as oscilações por um método de relaxação. Neste trabalho optou-se pelo método de Gauss-Seidel, uma vez que ele possui boas propriedades de suavização (Briggs et al., 2000) e é de fácil programação. Nas primeiras iterações, geralmente se tem rápida convergência, caracterizando a presença de modos oscilatórios de erro. Porém, após algumas iterações, o processo torna-se lento, sinalizando a predominância de modos suaves (Brandt, 1977). Este é exatamente o momento onde é recomendável transferir o problema de relaxação para a malha mais grossa. Pois, os modos de erros suaves na malha fina tornam-se erros oscilatórios na malha grossa (Wesseling, 1992). Para problemas lineares, o método *multigrid* é uma técnica que alterna entre passos de relaxação e soluções aproximadas da equação residual em uma malha mais grossa (*Correction Scheme*), acelerando a convergência do esquema de relaxação (Briggs et al., 2000). A taxa de convergência ideal (teórica) do *multigrid* é independente do tamanho da malha, isto é, independe do número de pontos da malha (Hirsch, 1988; Ferziger e Peric, 1999). Não é muito efetivo usar somente dois níveis de malha (Roache, 1998). Para obter o bom desempenho do *multigrid*, diversos níveis de malhas devem ser usados (Tannehill et al., 1997).

Neste trabalho são testados quatro algoritmos em problemas anisotrópicos (anisotropia devido aos elementos da malha serem altamente alongados):

- 1) Engrossamento padrão (EP): neste algoritmo, apresentado em Brandt (1977), é realizado simultaneamente o engrossamento em ambas as direções coordenadas. Desta forma, mantém-se a mesma razão de aspecto em todas as malhas. Este algoritmo é mostrado nos passos 1a a 1c da Fig. 1. A razão de engrossamento usada é dois, isto é,  $h_x$  e  $h_y$  de cada malha grossa são o dobro da malha fina que a originou.
- 2) Semi-engrossamento (SE): neste algoritmo, apresentado em Mulder (1989), é realizado o engrossamento em apenas uma das direções coordenadas, onde isto seja possível ou desejado. Com isso, diminui-se a razão de aspecto a cada nível percorrido até se atingir a malha mais grossa possível ou desejada, que idealmente é isotrópica, ou seja,  $RA = 1$ . Este algoritmo é mostrado nos passos 3a a 3c da Fig. 1.
- 3) Engrossamento padrão seguido de semi-engrossamento (EP-SE): neste algoritmo, proposto neste trabalho, primeiro aplica-se o algoritmo EP (passos 2a a 2c da Fig. 1) e, em seguida, o SE (passos 2c a 2e da Fig. 1).

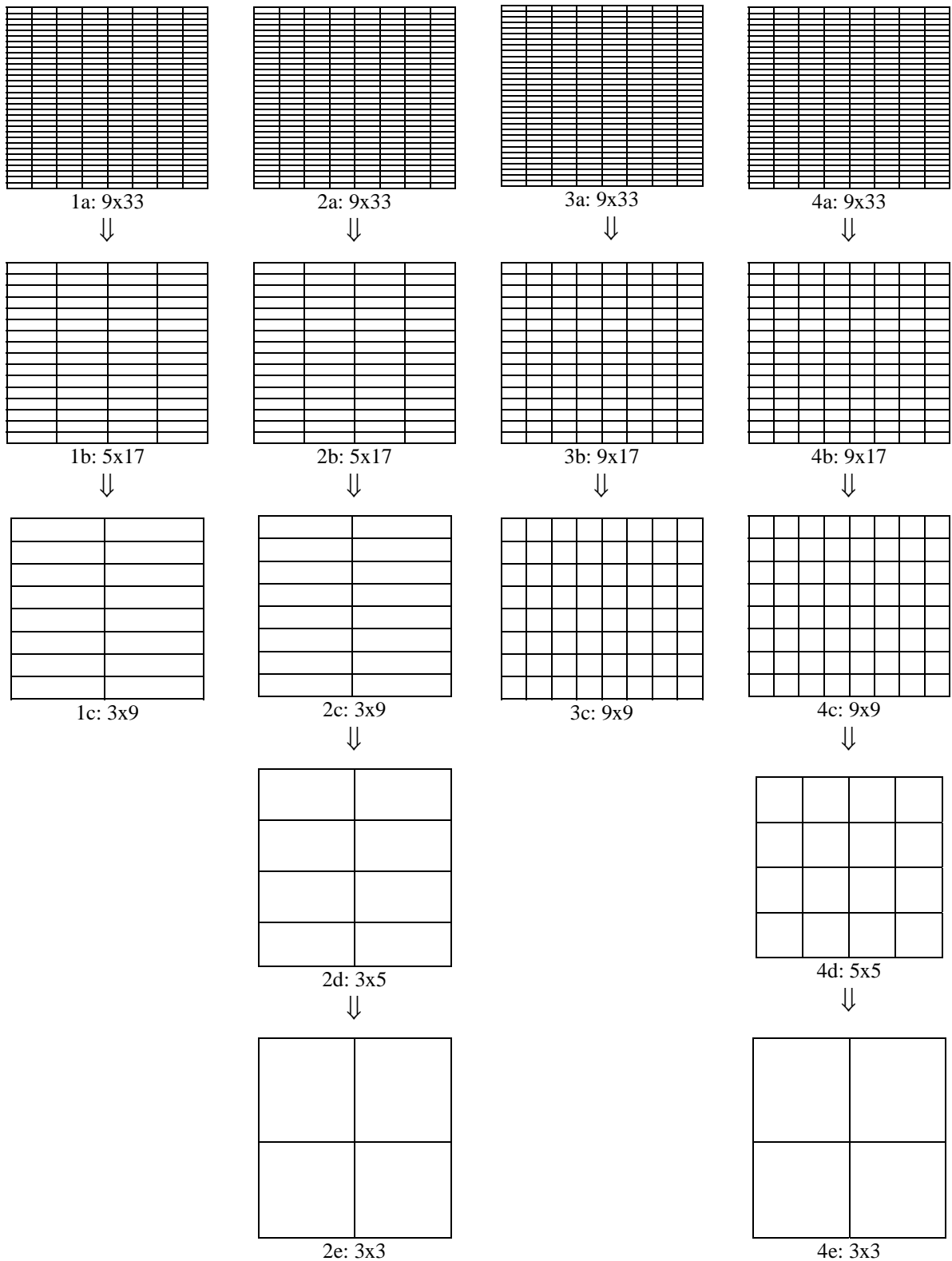


Figura 1 - Engrossamentos EP (1a-1c), EP-SE (2a-2e), SE (3a-3c) e SE-EP (4a-4e).

- 4) Semi-engrossamento seguido de engrossamento padrão (SE-EP): neste algoritmo, apresentado em Zhang (2002), primeiro aplica-se o algoritmo SE (passos 4a a 4c da Fig. 1) e, em seguida, o EP (passos 4c a 4e da Fig. 1).

No presente trabalho, adota-se o número máximo possível de níveis de engrossamento padrão para os algoritmos EP e EP-SE. Como se tratam de problemas anisotrópicos, para os algoritmos EP-SE, SE e SE-EP, a direção de semi-engrossamento sempre existe (Zhang, 2002).

Os operadores de transferência da malha fina para a malha grossa são chamados de operadores de restrição e são denotados genericamente por  $I_h^H \vec{\phi}^h = \vec{\phi}^H$ , onde  $\vec{\phi}$  assume o resíduo  $\vec{R}$  dado pela Eq. (5). Neste trabalho usa-se o operador de injeção para problemas bidimensionais, que pode ser visto em Briggs et al. (2000). Este operador não é calculado nos contornos, pois neste trabalho adotam-se condições de contorno de Dirichlet que implicam em  $\vec{R} = (0,0,\dots,0)$ .

Os operadores de transferência da malha grossa para a malha fina são chamados de operadores de prolongação, ou interpolação, e são denotados genericamente por  $I_H^h \vec{\phi}^H = \vec{\phi}^h$ , onde  $\vec{\phi}$  assume a aproximação do erro na equação residual, ou seja, a correção. Neste trabalho, usa-se o operador de interpolação bilinear, que também pode ser visto em Briggs et al. (2000).

### 3. MODELOS MATEMÁTICO E NUMÉRICO

O problema linear de condução de calor bidimensional (equação de Laplace), com condições de contorno de Dirichlet, em coordenadas cartesianas, considerado neste trabalho é (Maliska, 2004):

$$\begin{cases} \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0, & 0 < x, y < 1 \\ T(x,1) = \text{sen}(\pi \cdot x), & T(x,0) = T(0, y) = T(1, y) = 0 \end{cases}, \quad (6)$$

onde  $T$  é a incógnita e representa a temperatura. A solução analítica do problema é

$$T(x, y) = \text{sen}(\pi \cdot x) \frac{\text{senh}(\pi \cdot y)}{\text{senh}(\pi)}. \quad (7)$$

A discretização do domínio é feita com malhas uniformes cujos pontos são dados pela Eq. (3). Para cada um dos  $(N_x - 2) \times (N_y - 2)$  pontos interiores da malha, a Eq. (6) é discretizada com o método de diferenças finitas com diferença central (CDS) (Tannehill et al., 1997), resultando em

$$\begin{cases} \frac{v_{i-1,j} - 2v_{i,j} + v_{i+1,j}}{h_x^2} + \frac{v_{i,j-1} - 2v_{i,j} + v_{i,j+1}}{h_y^2} = 0, & 2 \leq i \leq N_x - 1, \quad 2 \leq j \leq N_y - 1 \\ v_{i,N_y} = \text{sen}(\pi \cdot x_{i,N_y}), & v_{i,1} = v_{1,j} = v_{N_x,j} = 0 \end{cases}, \quad (8)$$

onde  $v_{i,j}$  é uma aproximação (solução numérica) para a solução exata  $T(x_i, y_j)$ .

Rearranjando os termos da Eq. (8), obtém-se

$$a_p v_p = a_n v_n + a_s v_s + a_w v_w + a_e v_e + b_p, \quad (9)$$

onde os coeficientes são dados por  $a_p = 2/h_x^2 + 2/h_y^2$ ,  $a_n = 1/h_y^2$ ,  $a_s = 1/h_y^2$ ,  $a_w = 1/h_x^2$ ,  $a_e = 1/h_x^2$  e  $b_p = 0$ .

Se  $\vec{v}$  e  $\vec{f}$  são denotados por  $\vec{v} = (v_1, \dots, v_N)^t$  e  $\vec{f} = (f_1, \dots, f_N)^t$ , respectivamente, onde  $\vec{f}$  é o vetor independente formado pelos termos  $b_p$ , então o sistema da Eq. (8), pode ser representado por um sistema de equações algébricas do tipo da Eq. (1), onde  $\mathbf{A}$  é uma matriz pentadiagonal  $N \times N$ , simétrica e definida positiva (Briggs et al., 2000; Burden e Faires, 1997).

A Eq. (1) é resolvida com o método *multigrid* através dos quatro algoritmos descritos na seção anterior. Neste caso, os sistemas de equações deste tipo, onde  $\vec{f}$  representa o termo fonte (resíduo) a cada nível de malha, são resolvidos com o método de Gauss-Seidel.

## 4. RESULTADOS

Os algoritmos foram implementados na linguagem FORTRAN 95 com o aplicativo Compaq Visual Fortran 6.6 usando-se precisão dupla. As simulações foram realizadas num microcomputador com processador Intel Pentium 4 com 2.66 GHz e 1 GB RAM.

Cerca de 500 simulações foram realizadas com as seguintes variantes: algoritmos EP, EP-SE, SE e SE-EP; número de incógnitas de milhares a milhões; razões de aspecto ( $RA$ ) 1/1024, 1, 2, 16, 128, 1024 e 8192; e número de iterações internas. A razão de engrossamento padrão, ou seja,  $r=2$  foi utilizada em todos os algoritmos. São apresentados neste trabalho os resultados mais representativos.

O critério de convergência para as iterações externas é baseado na razão entre a norma  $L_1$  do resíduo (Ferziger e Peric, 1999) numa determinada iteração e a norma do resíduo da estimativa inicial. O resíduo de cada nó é calculado através da Eq. (5). Adotou-se  $\varepsilon = 10^{-10}$  e  $\vec{v} = (0,0,\dots,0)$  para a tolerância sobre o critério de convergência e a estimativa inicial, respectivamente.

O foco deste trabalho é a minimização do tempo de CPU. Entende-se por tempo de CPU, o tempo gasto para gerar as malhas, atribuir a estimativa inicial, calcular os coeficientes e resolver o sistema da Eq. (1). Este tempo é medido usando-se a função TIMEF da biblioteca PORTLIB do FORTRAN 95. Através de testes realizados verificou-se que a incerteza desta função é aproximadamente de  $\pm 0.05$  s.

### 4.1 Iterações internas ( $ITI$ ) e razões de aspecto ( $RA$ )

A Tabela 1 mostra as dimensões das malhas finas ( $N_x$  por  $N_y$ ) e as respectivas razões de aspecto ( $RA$ ) utilizadas para determinar o número ótimo de iterações internas ( $ITI_{\text{ótimo}}$ ). Este parâmetro é definido como o número de iterações internas no Gauss-Seidel que resulta no menor tempo de CPU para cada conjunto de dados constituído por: algoritmo,  $RA$  e número de incógnitas ( $N$ ).

A Tabela 2 mostra o  $ITI_{\text{ótimo}}$  para as malhas da Tabela 1. Pode-se notar que o  $ITI_{\text{ótimo}}$  é fortemente afetado pelo valor de  $RA$  e também pelo algoritmo utilizado, exceto o algoritmo SE-EP. Os algoritmos EP, EP-SE apresentam um comportamento crescente e acentuado do  $ITI_{\text{ótimo}}$  à medida que se aumenta  $RA$ . O algoritmo SE apresentou o comportamento oposto.

Finalmente, o algoritmo SE-EP apresentou o mesmo comportamento crescente dos algoritmos EP e EP-SE, porém de uma forma muito mais suave e parece ter estabilizado em  $ITI_{\text{ótimo}} = 3$ .

Tabela 1. Malhas usadas para determinar  $ITI_{\text{ótimo}}$

<i>RA</i>	<b>EP</b>	<b>EP-SE</b>	<b>SE</b>	<b>SE-EP</b>
1	1025x1025	-	-	-
2	257x513	257x513	129x257	257x513
16	33x513	33x513	33x513	33x513
128	9x1025	9x1025	9x1025	17x2049

Tabela 2. Valor do  $ITI_{\text{ótimo}}$

<i>RA</i>	<b>EP</b>	<b>EP-SE</b>	<b>SE</b>	<b>SE-EP</b>
1	1	-	-	-
2	2	2	350	2
16	70	82	25	3
128	450	250	7	3

#### 4.2 Efeito do número de incógnitas (*N*) sobre *RA*

As Figs. 2 a 4 apresentam uma comparação entre os diversos algoritmos para as razões de aspecto  $RA = 2, 16$  e  $128$ . Estas figuras mostram o tempo de CPU em função do número de incógnitas (*N*). Na Fig. 2 pode-se notar que o algoritmo SE-EP é o mais rápido entre os quatro algoritmos testados. Ele é seguido de perto pelos algoritmos EP e EP-SE, que quantitativamente são muito próximos, e finalmente pelo algoritmo SE, com tempo de CPU muito maior. Nesta figura, as seguintes malhas foram usadas:  $65 \times 129, 129 \times 257, 257 \times 513, 513 \times 1025$  e  $1025 \times 2049$  para os algoritmos EP e EP-SE;  $33 \times 65, 65 \times 129, 129 \times 257$  e  $257 \times 513$  para o algoritmo SE; e  $65 \times 129, 129 \times 257, 257 \times 513, 513 \times 1025, 1025 \times 2049$  e  $2049 \times 4097$  para o algoritmo SE-EP.

Conforme a Fig. 3, o algoritmo SE-EP também é o mais rápido. Ele é seguido pelo algoritmo SE em uma primeira parte dos pontos do domínio e a partir daí pelos algoritmos EP e EP-SE, que quantitativamente são muito próximos em praticamente todos os pontos do domínio. Nesta figura, as seguintes malhas foram usadas:  $9 \times 129, 17 \times 257, 33 \times 513, 65 \times 1025$  e  $129 \times 2049$  para os algoritmos EP, EP-SE e SE; e  $17 \times 257, 33 \times 513, 65 \times 1025, 129 \times 2049$  e  $257 \times 4097$  para o algoritmo SE-EP.

Na Fig. 4 pode-se notar que o algoritmo SE-EP continua sendo o mais rápido. Ele é seguido pelo algoritmo SE e finalmente pelos algoritmos EP e EP-SE, que quantitativamente são muito próximos. Nesta figura, as seguintes malhas foram usadas:  $5 \times 513, 9 \times 1025$  e  $17 \times 2049$  para o algoritmo EP;  $5 \times 513$  e  $9 \times 1025$  para o algoritmo EP-SE;  $5 \times 513, 9 \times 1025, 17 \times 2049$  e  $33 \times 4097$  para o algoritmo SE; e  $5 \times 513, 9 \times 1025, 17 \times 2049, 33 \times 4097, 65 \times 8193$  e  $129 \times 16385$  para o algoritmo SE-EP.

Portanto, o algoritmo SE-EP de Zhang (2002) é o mais rápido entre os quatro algoritmos e razões de aspecto testados. Pode-se notar que os algoritmos EP e EP-SE tornam-se mais lentos à medida que se aumenta *RA*. O algoritmo SE torna-se mais rápido, mas nunca ultrapassa o algoritmo SE-EP. Em resumo, para os mesmos dados ( $\varepsilon, \vec{v}, N$  e *RA*), dentre os quatro algoritmos testados, tem-se



$$t_{CPU}(\text{SE-EP}) < t_{CPU}(\text{EP, EP-SE, SE}), \quad (10)$$

em todos os pontos do domínio em estudo. Como comentado na introdução, Zhang (2002) já havia constatado que o algoritmo SE-EP, com os suavizadores red-black e four-color Gauss-Seidel, é o mais rápido, se comparado ao algoritmo EP, para  $RA = 2, 4, 8$  e  $16$  e a equação de Poisson bidimensional.

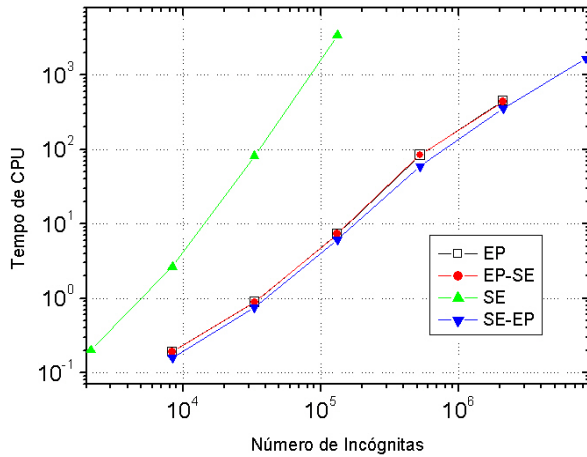


Fig. 2. Tempo de CPU *versus*  $N$  para  $RA=2$ .

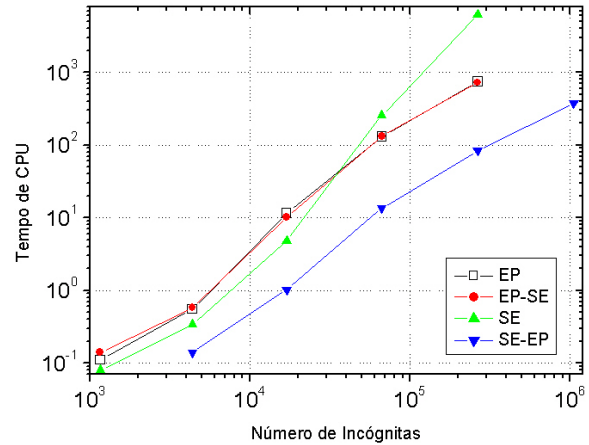


Fig. 3. Tempo de CPU *versus*  $N$  para  $RA=16$ .

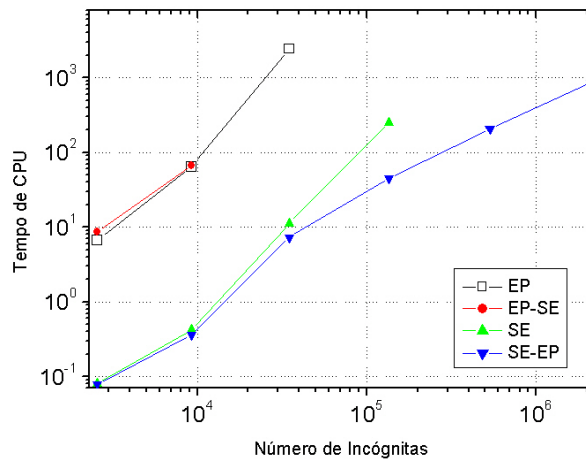


Fig. 4. Tempo de CPU *versus*  $N$  para  $RA=128$ .

Estes resultados já eram esperados, pois o semi-engrossamento reduz a razão de aspecto rapidamente e o problema anisotrópico aproxima-se do problema isotrópico em poucos níveis. Este procedimento seguido de engrossamento padrão aproveita a estrutura da malha fazendo com que o algoritmo SE-EP tenha bom desempenho. Por outro lado, o mau desempenho do algoritmo EP-SE é devido à permanência da alta razão de aspecto por diversos níveis no processo de engrossamento padrão. O algoritmo torna-se mais eficiente quando se inicia o processo de semi-engrossamento que, dependendo do valor da razão de aspecto, se dá no final do algoritmo. Isto faz com que este algoritmo (EP-SE) seja menos eficiente que o algoritmo anteriormente analisado (SE-EP), apesar do número de operações aritméticas, por ciclo, do algoritmo EP-SE ser bem inferior ao número de operações aritméticas do algoritmo SE-EP.

Portanto, o mais importante é reduzir o valor da razão de aspecto e não necessariamente o número de operações aritméticas.

### 4.3 Algoritmo de semi-engrossamento parcial (SE-EP)

Pelo fato do algoritmo SE-EP ser o mais rápido entre os quatro algoritmos testados, foi analisado o seu comportamento para alguns valores de  $RA$  ainda maiores. A Tabela 3 mostra as dimensões das malhas e as respectivas razões de aspecto complementares ( $RA=1/1024$ , 1, 1024 e 8192) utilizadas para obter o  $ITI_{\acute{o}timo}$ . Os resultados são apresentados na Tabela 4. Pode-se notar nas Tabelas 2 e 4 que  $ITI_{\acute{o}timo}$ , para o algoritmo SE-EP, é fracamente afetado por  $RA$ , sendo igual a três, exceto num caso.

Tabela 3. Malhas adicionais usadas para obter  $ITI_{\acute{o}timo}$  com o algoritmo SE-EP

$RA$	1/1024	1024	8192
$N_x$ por $N_y$	8193x9	9x8193	9x131072

Tabela 4.  $ITI_{\acute{o}timo}$  para o algoritmo SE-EP

$RA$	1/1024	1024	8192
$ITI_{\acute{o}timo}$	3	3	3

A Tabela 5 mostra o valor do expoente  $b$  da seguinte equação:

$$t_{CPU} = a N^b \quad (11)$$

Ele foi obtido por ajuste de mínimos quadrados para os resultados da Fig. 5 cujas malhas usadas foram: 4097x5, 8193x9 e 16385x17 para a  $RA=1/1024$ ; 129x129, 257x257, 513x513, 1025x1025, 2049x2049 para a  $RA=1$ ; 65x129, 129x257, 257x513, 513x1025, 1025x2049 e 2049x4097 para a  $RA=2$ ; 17x257, 33x513, 65x1025, 129x2049, 257x4097 e 513x8193 para a  $RA=16$ ; 9x1025, 17x2049, 33x4097, 65x8193 e 129x16385 para a  $RA=128$ ; 5x4097, 9x8193, 17x16385 e 33x32769 para a  $RA=1024$  e 5x32769, 9x65537 e 17x131073 para a  $RA=8192$ . Pode-se observar que  $b(1/RA) > b(RA)$ . E, em geral, dado  $1 \leq RA_1 < RA_2$ , tem-se que  $b(RA_1) < b(RA_2)$ . O comportamento crescente de  $b$  em função da  $RA \geq 1$  só é interrompido para  $RA=16$ . O crescimento de  $b$  em função da  $RA$  não ocorre de uma forma acentuada, ou seja, há uma fraca dependência entre  $b$  e  $RA$ .

Tabela 5. Valor de  $b$  da Eq. (11) para o algoritmo SE-EP

$RA$	1/1024	1	2	16	128	1024	8192
$b$	1.93	1.35	1.39	1.38	1.39	1.49	1.51

Pode-se observar na Fig. 5 que em geral

$$\begin{aligned} t_{CPU}(RA=1) &< t_{CPU}(RA=2) < t_{CPU}(RA=8192) < \\ t_{CPU}(RA=16) &< t_{CPU}(RA=1024) < t_{CPU}(RA=128). \end{aligned} \quad (12)$$

A primeira desigualdade da Eq. (12) era esperada, pois trata-se da comparação entre um problema isotrópico e um problema anisotrópico com a malha levemente alongada, ou seja, razões de aspecto próximas da unidade ( $0 < RA \approx 1$ ). A segunda desigualdade impressiona, pois trata-se da comparação entre dois problemas anisotrópicos, porém com razões de aspecto bem distintas: uma levemente alongada e a outra fortemente alongada, ou seja, razão de aspecto muito distinta da unidade ( $RA \gg 1$ ). De uma forma geral, pode-se concluir da Eq. (12) que, razões de aspecto próximas da unidade ou muito distintas da unidade são razões de aspecto para as quais o algoritmo SE-EP possui um bom desempenho, por exemplo,  $RA = 2$  e  $8192$ . Por outro lado, para razões de aspecto intermediárias ( $RA = 128$ ), o algoritmo SE-EP possui um dos piores desempenhos, entre as razões de aspecto testadas. Isto pode ser explicado pelo fato de que para problemas anisotrópicos com malhas levemente alongadas, cujas razões de aspecto são próximas da unidade, o algoritmo SE-EP comporta-se de maneira muito semelhante quando se trata de problema isotrópico, onde o *multigrid* funciona bem. Por outro lado, para problemas anisotrópicos com malhas fortemente alongadas, cujas razões de aspecto são muito distintas da unidade, o algoritmo SE-EP comporta-se muito semelhante quando se trata de problemas unidimensionais, devido à predominância de semi-engrossamentos; e neste caso, o *multigrid* também funciona bem.

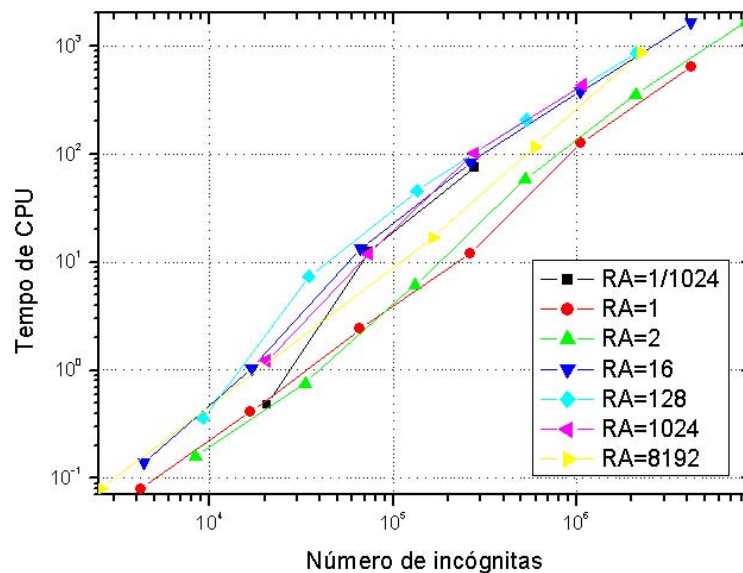


Figura 5 - Tempo de CPU *versus*  $N$  para o algoritmo SE-EP.

Os resultados observados na Fig. 5 são mais confiáveis do que os resultados da Tabela 5, pois estes últimos resultam do ajuste de curvas, que depende fortemente da quantidade e qualidade dos dados observados, tendo portanto, erros. E mais, os resultados da Tabela 5 fornecem apenas uma aproximação da ordem de complexidade dos algoritmos para as diversas razões de aspecto, enquanto que os resultados da Fig. 5 fornecem o tempo de CPU em função do número de incógnitas.

## 5. CONCLUSÃO

Neste trabalho estudou-se o efeito da razão de aspecto ( $RA$ ) de malhas sobre o tempo de CPU necessário para resolver a equação de Laplace bidimensional. A equação foi discretizada com o método de diferenças finitas. A solução do sistema de equações foi obtida com o

método *multigrid* geométrico do tipo esquema de correção (*Correction Scheme*) com ciclo V. Comparações foram feitas com quatro algoritmos (EP, EP-SE, SE e SE-EP) e diversas razões de aspecto ( $RA = 1/1024, 1, 2, 16, 128, 1024$  e  $8192$ ).

Com base nos resultados deste trabalho, verificou-se que:

- 1) Para os mesmos dados ( $\varepsilon, \bar{v}, N$  e  $RA$ ), o algoritmo SE-EP tem o melhor desempenho, isto é, resulta no menor tempo de CPU entre os quatro algoritmos testados.
- 2) A variação da razão de aspecto de malha resulta em grande variação do número ótimo de iterações internas para os algoritmos EP, EP-SE e SE, e pequena variação para o algoritmo SE-EP.
- 3) O número ótimo de iterações internas para o algoritmo SE-EP é o mesmo para  $RA$  e  $1/RA$ , e para razões de aspecto muito distintas da unidade, ou seja,  $0 < RA \ll 1$  ou  $RA \gg 1$ .
- 4) Para o algoritmo SE-EP, o valor do expoente  $b$  da Eq. (11), em geral, cresce com a razão de aspecto de malha: varia de 1.35 a 1.51 para  $RA = 1$  a 8192.
- 5) Para os mesmos dados ( $\varepsilon, \bar{v}$  e  $N$ ), o algoritmo SE-EP resulta em menor tempo de CPU para malhas levemente ( $0 < RA \approx 1$ ) e fortemente alongadas ( $RA \gg 1$ ); e maior tempo de CPU para malhas com  $RA$  intermediário.

### ***Agradecimentos***

O primeiro autor agradece o apoio do Laboratório de Experimentação Numérica (LENA), do Departamento de Engenharia Mecânica da UFPR, por disponibilizar sua infra-estrutura, à Universidade Estadual de Ponta Grossa pelo suporte financeiro e aos amigos do LENA.

### **REFERÊNCIAS**

- Brandt, A., "Multi-Level adaptive solutions to boundary-value problems, *Mathematics of Computation*", vol. 31, pp. 333-390, 1977.
- Briggs, W. L. and Henson, V.E., McCormick, S.F., "A Multigrid Tutorial", 2ª ed., SIAM, 2000.
- Burden, R. L. and Faires, J. D., "Numerical Analysis", 6ª ed., Brooks/Cole Publishing Company, 1997.
- Dendy Jr., J. E., McCormick, S. F., Ruge, J. W., Russel, T. F., Schaffer, T., "Multigrid methods for three-dimensional petroleum reservoir simulation", SPE 18409, pp. 19-25, 1989.
- Fedorenko, R. P., "On the Speed of Convergence of an Iteration Process", *USSR Comput. Math. And Math. Phys.*, vol. 4 (3), 1964.
- Ferziger, J. H. and Peric, M., "Computational Methods for Fluid Dynamics", 2ª ed., Springer, 1999.
- Fortuna, A. O., "Técnicas Computacionais para Dinâmica dos Fluidos", Edusp, 2000.

- Golub, G. H. and Ortega, J. M., “Scientific Computing and Differential Equations: An Introduction to Numerical Methods”, Academic Press, Inc., 1992.
- Golub, G. H. and Van Loan, C., “Matrix Computations”, 2<sup>a</sup> ed., Johns Hopkins Press, Baltimore, 1989.
- Greenberg, M. D., “Advanced Engineering Mathematics”, 2<sup>a</sup> ed., Printice Hall, Inc., New Jersey, 1998.
- Hestenes, M. and Stiefel, E., “Methods of Conjugate Gradients for Solving Linear Systems”, Journal of Research of the National Bureau of Standarts, vol. 49, pp. 409-436, 1952.
- Hirsch, C., “Numerical Computational of Internal and External Flows”, vol.1, Wiley, 1998.
- Larsson, J., Lien, F. S., yee, E., “Conditional Semicoarsening Multigrid Algorithm for the Poisson Equation on Anisotropic grids”, Journal of Computational Physics, vol. 208, pp. 368-383, 2005.
- Maliska, C. R., “Transferência de calor e mecânica dos fluidos computacional”, LTC, 2<sup>a</sup> ed., 2004.
- Montero, R. S., Llorense, I. M., Salas, M. D., “Robust Multigrid Algorithms for the Navier-Stokes Equations”, Journal of Computational Physics, vol. 173, pp. 412-432, 2001.
- Moro Filho, R. C., “Aplicação da Técnica Multigrid em Transferência de Calor Computacional”, XXV Iberian Latin\_American Congress on Computational Methods in Engineering, 2004.
- Mulder, W. A., “A New Multigrid Approach to Convection Problems”, Journal of Computational Physiscs, vol. 83, pp. 303-323, 1989.
- Naik, N. H.; Van Rosendale, J., “The improved robustness of Multgrid elliptic solvers based on multiple semicoarsened grids”, SIAM Journal in Numerical Analysis, vol. 30, pp. 215-229, 1993.
- Oosterlee, C. W.; Wesseling, P., “A robust Multgrid method for a discretization of the incompressible Navier-Stokes equations in general coordinates”, Impact of Computing in Science and Engineering, vol. 5, pp. 128-151, 1993.
- Pierce, N. A.; Alonso, J.J., “Efficient computation of unsteady viscous flows by an implicit preconditioned Multgrid method”, AIAA Journal, vol. 36, n. 3, pp. 401-408, 1998.
- Pinto, M. A. V.; Santiago, C. D.; Marchi, C. H., “Effect of Parameters of a Multigrid Method on the CPU Time for One-dimensional Problems”, Proceedings of COBEM, 2005.
- Radespiel, R.; Swanson, R.C., “Progress with Multgrid schemes for hyperbolic flow problems”, Journal of Computational Physics, vol. 116, pp. 103-122, 1995.
- Roache, P. J., “Fundamentals of Computational Fluid Dynamics”, Hermosa Publishers, 1998.

- Sivaloganathan, S., "The use of local mode analysis in the design na comparison of Multgrid methods", *Computer Physics Communications*, vol. 65, pp. 246-252, 1991.
- Sockol, P.M., "Multgrid solution of the Navier-Stokes equations on highly stretched grids", *International Journal for Numerical Methods in Fluids*, vol. 17, pp. 543-566, 1993.
- Stüben, K., "Algebraic Multgrid (AMG): an introduction with applications", in: *GMD-Report 70*, November 1999.
- Stüben, K., "A Review of Algebraic Multgrid", *Journal of Computation and Applied Mathematics*, vol. 128, pp. 281-309, 2001.
- Tannehill, J. C., Anderson, D. A. and Pletcher, R. H., "Computational Fluid Mechanics and Heat Transfer", 2<sup>a</sup> ed., Washington: Taylor & Francis, 1997.
- Wesseling, P., "An Introduction to *Multgrid* Methods", John Wiley & Sons, 1992.
- Wesseling, P. and Oosterlee, C. W., "Geometric Multgrid with Applications to Computational Fluid Dynamics", *Journal of Computation and Applied Mathematics*, vol. 128, pp. 311-334, 2001.
- Zhang, J., "Multigrid Method and Fourth-Order Compact Scheme for 2D Poisson Equation with Unequal Mesh-Size Discretization", *Journal of Computational Physics*, vol. 179, pp. 170-179, 2002.