

## EFEITO DE PARÂMETROS DO MÉTODO *MULTIGRID* SOBRE O TEMPO DE CPU PARA A EQUAÇÃO DE BURGERS UNIDIMENSIONAL

**M. A. V. Pinto**

*marciovp@demec.ufpr.br*

Departamento de Matemática e Estatística – UEPG, Ponta Grossa, PR, Brasil

**C. D. Santiago**

*cosmo@unibrasil.com.br*

Faculdades do Brasil – Unibrasil, Curitiba, PR, Brasil

**C. H. Marchi**

*marchi@demec.ufpr.br*

Departamento de Engenharia Mecânica, UFPR, Curitiba, PR, Brasil

**Resumo.** Neste trabalho estuda-se a influência de vários parâmetros do método Multigrid geométrico sobre o tempo de CPU. Os parâmetros considerados são: a razão de engrossamento das malhas, número de iterações internas, número de níveis, tolerâncias e estimativas iniciais. O modelo matemático considerado envolve um problema de escoamento não-linear unidimensional, equação de Burgers, com condições de contorno de Dirichlet. A equação diferencial é discretizada com o método de diferenças finitas. No caso Multigrid, os sistemas de equações algébricas são resolvidos com o método de Gauss-Seidel. Comparações são feitas com dois métodos Singlegrid: Gauss-Seidel e TDMA. O algoritmo Multigrid usado é o Esquema de Aproximação Completa (Full Approximation Scheme) com ciclo V. A restrição é feita por injeção e a prolongação por interpolação linear. Alguns resultados confirmam os resultados da literatura e outros novos são apresentados.

**Palavras-chave:** Diferenças finitas, Solvers, Razão de engrossamento, Escoamento unidimensional

### 1. INTRODUÇÃO

A discretização de modelos matemáticos que surgem na dinâmica dos fluidos computacional conduz a grandes sistemas de equações algébricas do tipo

$$A\vec{x} = \vec{b}, \quad (1)$$

em que  $A$  é uma matriz quadrada,  $\vec{b}$  é o vetor independente e  $\vec{x}$  é o vetor de incógnitas. Estes modelos surgem, geralmente, em fenômenos físicos que envolvem fluidos em movimento, com ou sem troca de calor (Fortuna, 2000; Maliska, 2004). A estrutura da matriz  $A$  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 unidimensionais, o domínio  $x \in \mathcal{R} : 0 \leq x \leq 1$  é particionado em  $N_f$  subintervalos, introduzindo uma malha com os pontos  $x_i = (i-1)h$ , onde  $i = 1, \dots, N_f + 1$  e  $h = 1/N_f$  é o comprimento de cada intervalo. Isto estabelece uma malha de tamanho  $h$  que denota-se por  $\Omega^h$ . A cada um dos  $N_f - 1$  pontos interiores à malha, a

equação diferencial que representa o fenômeno físico é substituída por aproximações de diferenças finitas de primeira e segunda ordem (Tannehill et al., 1997; Ferziger e Peric, 1999).

Várias técnicas numéricas têm sido estudadas para resolver o sistema (1) com o menor custo computacional e a solução a mais próxima possível da exata (sem erros de iteração). 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). A partir de uma estimativa inicial, o método gera uma seqüência de soluções que, em geral, aproxima da solução exata do problema. Os métodos do gradiente conjugado (Burden e Faires, 1997) e do gradiente conjugado pré-condicionado (Dennis e Schnabel, 1983) usam técnicas que são mais específicas para geometrias simples e problemas cuja matriz dos coeficientes seja mal-condicionada. Para geometrias mais complexas, a técnica de decomposição de domínio (Perng e Street, 1991; Lai e Przekwas, 1996) é também usada.

O método *Multigrid*, proposto originalmente por Fedorenko (1964) é apresentado como uma técnica numérica alternativa para resolver iterativamente sistemas de equações do tipo (1). A idéia básica é usar um conjunto de malhas e alternar entre as iterações em cada nível de malha e as aproximações destas soluções em uma malha mais grossa (Briggs et al., 2000). Então, operadores que transferem vetores da malha fina para a malha imediatamente mais grossa (restrição) são definidos. O mesmo procedimento é aplicado da malha grossa para a malha imediatamente mais fina (prolongação). Os sistemas lineares em cada malha são resolvidos com um método iterativo com propriedades de reduzir rapidamente os erros oscilatórios (propriedades de suavização). Com este conceito, vários trabalhos foram publicados apresentando bons resultados numéricos para os problemas da dinâmica dos fluidos. As conclusões de Fedorenko (1964) mostraram que a velocidade de convergência com o uso da técnica *Multigrid* é melhor que a dos métodos iterativos puros. O objetivo da técnica *Multigrid* é acelerar a convergência de um esquema iterativo (Tannehill et al., 1997). Os melhores desempenhos do método *Multigrid* são obtidos em problemas totalmente elípticos (Wesseling, 1992), ou seja, problemas dominados pela difusão; e os menores em problemas dominados pela advecção (Ferziger e Peric, 1999).

O método *Multigrid* pode ser aplicado a malhas estruturadas, também chamado de *Multigrid* geométrico (Wesseling e Oosterlee, 2001), assim como em malhas não-estruturadas, conhecido como *Multigrid* algébrico (Stüben, 2001). Em Briggs et al. (2000) é comparado o uso do *Multigrid* geométrico e algébrico em um problema de Poisson bidimensional com condição de fronteira de Dirichlet. Em Wesseling e Oosterlee (2001) muitos desafios ainda são vistos na área geométrica, como a solução das Equações de Navier-Stokes, problemas com perturbações singulares, problemas de camada limite onde aparecem as malhas fortemente distorcidas, ou mesmo a paralelização de algoritmos.

Brandt (1977) trabalhou com o *Multigrid* geométrico em diversos problemas de transferência de calor e escoamento de fluidos, unidimensionais e bidimensionais, lineares e não-lineares. Ele fez comparações com as razões de engrossamento  $r = 1/2$ ,  $r = 1/3$  e  $r = 2/3$ , onde a razão de engrossamento é definida como

$$r = \frac{h_{\Omega^h}}{h_{\Omega^H}}, \quad (2)$$

onde  $\Omega^h$  representa uma malha fina,  $\Omega^H$  a malha imediatamente mais grossa e  $h$  o tamanho do elemento da malha, para o caso unidimensional e malha uniforme. Para os problemas

testados, a conclusão de Brandt (1977) é que a razão  $r = 1/2$  é a recomendável, pois, segundo ele, é a razão próxima da ótima e a mais fácil de programar. Stüben (1999) desenvolveu um estudo com as razões  $r = 1/2$  e  $r = 1/4$  em malhas não-estruturadas para diversos problemas de transferência de calor, escoamento e eletromagnetismo, bidimensionais e tridimensionais, lineares e não-lineares. Em seu trabalho, a razão  $r = 1/4$  se mostrou eficiente para problemas anisotrópicos (anisotropia dos coeficientes e anisotropia devido à malhas altamente distorcidas). Briggs et al. (2000) trabalharam com a razão  $r = 1/2$  afirmando ser uma prática universal e que  $r \neq 1/2$  não traz vantagens. Moro (2004) trabalhou com as razões  $r = 1/2$  e  $r = 1/4$  em malhas estruturadas para problemas de difusão com termo fonte. Em seu trabalho a razão  $r = 1/4$  se mostrou bastante eficiente, com o tempo de CPU menor do que para a razão  $r = 1/2$ . Pinto et al. (2005) trabalharam com as razões  $r = 1/2$ ,  $r = 1/3$ ,  $r = 1/4$ ,  $r = 1/5$  e  $r = 1/8$  para problemas unidimensionais lineares (equação de difusão e equação de advecção-difusão). Neste trabalho a razão  $r = 1/2$  mostrou-se mais eficiente.

Embora não haja razão para usar o método *Multigrid* em uma dimensão, é fácil ilustrar os princípios do método e derivar alguns procedimentos usados no caso geral (Ferziger e Peric, 1999). Pois, desta forma, uma grande quantidade de testes poderá ser feita devido à rapidez da obtenção das soluções. Isto facilita o estudo de uma grande variação de parâmetros.

Neste trabalho os seguintes parâmetros são estudados: diversas razões de engrossamento ( $r = 1/2$ ,  $r = 1/3$ ,  $r = 1/4$ ,  $r = 1/5$  e  $r = 1/8$ ), o número de iterações internas e o número de níveis. O objetivo é verificar o efeito desses parâmetros sobre o tempo de CPU para o Multigrid geométrico. O algoritmo *Multigrid* adotado é o Esquema de Aproximação Completa (*Full Approximation Scheme*) com ciclo  $V$  proposto em Wesseling (1992). Os resultados são comparados com os obtidos na bibliografia. São apresentados operadores de restrição e prolongação para qualquer razão de engrossamento no intervalo (0,1). O modelo matemático considerado neste trabalho envolve o problema não-linear unidimensional de escoamento, isto é, equação de Burgers com condições de contorno de Dirichlet.

Este artigo está organizado da seguinte forma: na seção 2 é dada uma visão geral do método *Multigrid*, apresentando os operadores de restrição, prolongação e o método iterativo de Gauss-Seidel. Na seção 3, é dado o modelo matemático e numérico. Nas seções 4 e 5 são descritos os experimentos numéricos com seus resultados e a conclusão do trabalho.

## 2. MÉTODO MULTIGRID

A resolução de problemas de mecânica dos fluidos e transferência de calor 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. Um método alternativo usado para melhorar a taxa de convergência destes problemas é o método *Multigrid* (Briggs et al., 2000), que acelera consideravelmente a resolução dos sistemas lineares envolvidos no problema. Métodos *Multigrid* são métodos iterativos de resolução de sistemas lineares, sendo, portanto, fortemente dependentes da estimativa inicial atribuída às incógnitas do problema.

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

$$\vec{R} = \vec{b} - A\vec{x}, \quad (3)$$

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).

As primeiras iterações deste processo, geralmente, têm 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 não-lineares, o método *Multigrid* é uma técnica que alterna entre passos de relaxação e soluções aproximadas do problema e da equação residual em uma malha mais grossa (*Full Approximation Scheme*) acelerando a convergência do esquema de relaxação (Briggs et al., 2000). A taxa de convergência 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 a razão de engrossamento para malhas uniformes é dada pela Eq. (2), onde:  $r \in (0 ; 0,5)$  será denominado engrossamento forte;  $r = 0,5$ , engrossamento padrão e  $r \in (0,5 ; 1)$ , engrossamento fraco. Neste trabalho estudou-se os engrossamentos forte e padrão aplicados ao algoritmo encontrado em Wesseling (1992). Uma forma alternativa da Eq. (2) é

$$r = \frac{p}{q}, \quad p, q \in Z_+, \quad p < q. \quad (4)$$

Neste caso, a razão de engrossamento  $r$  é chamada de razão pura se  $p = 1$ .

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 a solução aproximada do problema e também o resíduo  $\vec{R}$  dado pela Eq. (3). Em Pinto et al. (2005) foi desenvolvido um operador de injeção com a sua forma generalizada para qualquer  $r \in (0,1)$  dado por:

$$\vec{\phi}_i^H = K_r \cdot \vec{\phi}_{cr}^h + (1 - K_r) \cdot \vec{\phi}_{cr+1}^h; \quad 2 \leq i \leq N^H, \quad (5)$$

com  $cr = \text{ceiling}\left(\frac{q}{p}(i-1)\right)$ ,  $K_r = cr - \frac{q}{p}(i-1)$ ,  $N^H = N^h \cdot \frac{p}{q}$  e  $N^h$  o número de intervalos da malha imediatamente mais fina. A função *ceiling* é definida por:

$$\text{ceiling} : \mathfrak{R} \rightarrow Z, \quad \text{com } x \mapsto \text{ceiling}(x) = \min\{z \in Z / z \geq x\}. \quad (6)$$

A Eq. (5) não é calculada para  $i = 1$  e  $N^H + 1$ , pois neste trabalho adota-se as condições de contorno de Dirichlet. Portanto tem-se  $\vec{R} = (0,0,\dots,0)$  nestes pontos.

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 solução aproximada do problema e também a aproximação do erro na equação residual, ou seja, a correção. Como no caso do operador de restrição, em Pinto et al.

(2005) foi desenvolvido um operador de interpolação linear com a sua forma generalizada para qualquer  $r \in (0,1)$  dada por

$$\vec{\phi}_i^h = K_p \cdot \vec{\phi}_{cp}^H + (1 - K_p) \vec{\phi}_{cp+1}^H; \quad 2 \leq i \leq N^h, \quad (7)$$

com  $cp = \text{ceiling}\left(\frac{p}{q}(i-1)\right)$ ,  $K_p = cp - \frac{p}{q}(i-1)$ .

### 3. MODELOS MATEMÁTICO E NUMÉRICO

O problema de escoamento não-linear unidimensional (equação de Burgers) de um fluido incompressível com condições de contorno de Dirichlet em regime permanente e em coordenadas cartesianas pode ser adaptado de Tannehill et al. (1997) e representado matematicamente por:

$$Re \cdot u_x^2 = u_{xx} + S, \quad 0 < x < 1; \quad u(0) = 0, \quad u(1) = 1, \quad (8)$$

onde  $u$  é a incógnita,  $S$  é um termo fonte dado por  $S = Re^2 e^{xRe} (2e^{xRe} - e^{Re} - 1) / (e^{Re} - 1)^2$  e  $Re = 20$ . Para estas condições de contorno e termo fonte, a solução analítica é dada por  $u(x) = (e^{xRe} - 1) / (e^{Re} - 1)$ . Na Eq. (8) a variável  $u$  representa a velocidade. Suas derivadas primeira e segunda são representadas por  $u_x$  e  $u_{xx}$ , respectivamente.

A discretização do domínio é desenvolvida fazendo-se uso de malhas uniformes em  $N_f$  subintervalos (elementos) introduzidos pelos pontos da malha dados por  $x_i = (i-1)h$ , onde  $i = 1, \dots, N_f + 1$  e  $h = 1 / N_f$  é o comprimento de cada subintervalo. A equação diferencial dada pela Eq. (8) é discretizada de acordo com o método de diferenças finitas. Nesta equação é utilizada diferença atrasada (UDS) para a derivada de primeira ordem (termo advectivo) e diferença centrada (CDS) para a derivada de segunda ordem (termo difusivo). Os esquemas UDS e CDS podem ser vistos em Tannehill et al. (1997). A equação resultante é:

$$Re \cdot \frac{v_i^2 - v_{i-1}^2}{h} = \frac{v_{i-1} - 2v_i + v_{i+1}}{h^2} + S_i, \quad 2 \leq i \leq N_f; \quad v_1 = 0, \quad v_{N_f+1} = 1, \quad (9)$$

onde  $v_i$  é uma aproximação (solução numérica) para a solução exata  $u(x_i)$  e  $S_i = S(x_i)$ .

A linearização da Eq. (9), como em Ferziger e Peric (1999), é da forma:

$$Re \cdot \frac{v_i^* v_i - v_{i-1}^* v_{i-1}}{h} = \frac{v_{i-1} - 2v_i + v_{i+1}}{h^2} + S_i, \quad 2 \leq i \leq N_f; \quad v_1 = 0, \quad v_{N_f+1} = 1, \quad (10)$$

onde  $v_i^2 \approx v_i^* v_i$  e o índice \* denota valores que são obtidos da iteração anterior.

Rearranjando os termos da Eq. (10), obtém-se a forma geral linearizada  $a_p v_p = a_w v_w + a_e v_e + b_p$ , onde os coeficientes são dados por  $a_p = (Re.v_p^*)/h + 2/h^2$ ,  $a_w = (Re.v_w^*)/h + 1/h^2$ ,  $a_e = 1/h^2$  e  $b_p = S_p$ .

Se  $\vec{v}$  e  $\vec{f}$  são denotados por  $\vec{v} = (v_1, \dots, v_{N_f+1})$  e  $\vec{f} = (f_1, \dots, f_{N_f+1})$ , respectivamente, então o sistema dado pela Eq. (9), após a linearização, pode ser representado por um sistema de equações algébricas do tipo:

$$A\vec{v} = \vec{f}, \quad (11)$$

onde  $A = A(\vec{v})$  é uma matriz tridiagonal  $(N_f+1) \times (N_f+1)$ , simétrica e definida positiva (Briggs et al., 2000; Burden e Faires, 1997),  $\vec{f}$  é o vetor independente e  $\vec{v}$  é o vetor de incógnitas.

A equação (11) é resolvida com o método direto TDMA (*Tridiagonal Matrix Algorithm*) dado em Ferziger e Peric (1999), para obter-se informações sobre a performance do método neste tipo de problema. Neste caso, a solução do problema não-linear assume um comportamento iterativo, mesmo com o uso do TDMA. Resolve-se também este problema, apenas na malha mais fina, com os métodos *Singlegrid*: Gauss-Seidel e TDMA. No caso do método *Multigrid*, os sistemas de equações  $A\vec{v} = \vec{b}$ , onde  $\vec{b}$  representa o termo fonte (resíduo) a cada nível de malha, são resolvidos com o método de Gauss-Seidel.

#### 4. RESULTADOS NUMÉRICOS

Os algoritmos foram implementados na linguagem FORTRAN/95 com o uso do Compaq Visual FORTRAN 6.6. Os testes foram realizados num microcomputador com Processador Intel Pentium 4 com 2,66 GHz e 1 GB RAM usando aritmética de dupla precisão. Os seguintes engrossamentos foram usados: engrossamento padrão, ou seja,  $r=1/2$  e alguns engrossamentos fortes usando-se razões puras, como:  $r=1/3$ ,  $r=1/4$ ,  $r=1/5$  e  $r=1/8$ . Outros métodos foram usados para comparação, por exemplo, o método iterativo de Gauss-Seidel e o método direto TDMA (solução iterativa devido à não-linearidade do problema).

Centenas de testes foram realizados com outras variantes, como: dimensão dos problemas (de pequenos problemas até problemas da ordem de milhões de variáveis), número de iterações internas, número de níveis de malhas, tolerâncias e estimativas iniciais. Somente alguns destes testes são apresentados neste trabalho, pois são os testes mais representativos, suficientes para mostrar qualitativamente o desempenho do método *Multigrid*.

O critério de parada para as iterações externas é dado pela razão entre a norma  $L_1$  do resíduo (Ferziger e Peric, 1999) e a norma do resíduo com base na estimativa inicial, onde o resíduo de cada nó é calculado através da Eq. (3). Neste trabalho adota-se como referência  $\varepsilon = 10^{-7}$  e  $\vec{v} = (0,0,\dots,0)$  para o critério de parada e estimativa inicial, respectivamente.

Na Fig. 1 têm-se as soluções analítica e numérica para o problema definido pela Eq. (8). A solução analítica é comparada com as soluções numéricas para três métodos distintos: *Multigrid* usando a razão de engrossamento  $r=1/2$ , o método de Gauss-Seidel e o método direto TDMA.

O foco de estudo deste trabalho é a minimização do tempo de CPU. Entende-se por tempo de CPU, o tempo gasto para realizar a geração de malhas, atribuição da estimativa inicial,

cálculo dos coeficientes e resolução do sistema linear (11). 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.

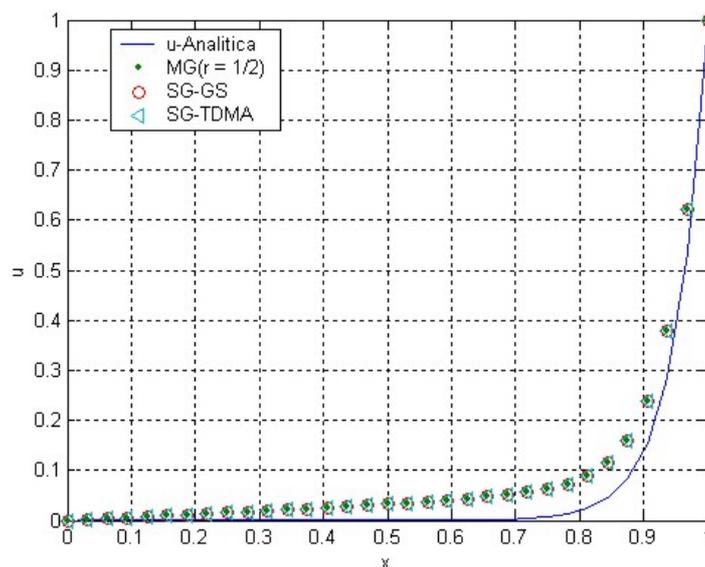


Figura 1- Solução analítica e soluções numéricas via MG( $r = 1/2$ ), Gauss-Seidel e TDMA para  $N_f = 32$

#### 4.1 Iterações internas (ITI)

A Fig. 2 mostra a influência do número de iterações internas sobre o tempo de CPU e o número ótimo para diversas razões de engrossamento e dimensões de problemas dados pela Tab.1 para a Equação de Burgers. A Tab. 1 mostra em sua primeira linha as razões de engrossamento utilizadas em testes numéricos para a Equação de Burgers. Na segunda linha têm-se algumas dimensões de malhas e suas respectivas razões. Pode-se notar que o número de iterações internas afeta o tempo de CPU e que  $ITI_{otimo} \approx 3q$  para a maioria das razões testadas, exceto para a razão  $r = 1/8$ , onde  $ITI_{otimo} = 3q + 4$ , onde  $q$  é definido na Eq. (4).

Nota-se que, relativamente ao ponto mínimo, uma diminuição sensível do número de iterações internas aumenta drasticamente o tempo de CPU, podendo até mesmo ocorrer a divergência. Por outro lado, o aumento sensível do número de iterações internas, aumenta pouco o tempo de CPU. Portanto recomenda-se utilizar  $ITI = 3q$  para as razões puras da forma  $r = 1/q$ ; por exemplo, para  $r = 1/2$ , recomenda-se  $ITI = 6$ . Os resultados do estudo do número de iterações internas ótimo e do tempo de CPU ao adotar-se o número  $ITI = 3q$  sofreram, em média, pouca variação para as razões  $r = 1/2$ ,  $r = 1/3$ ,  $r = 1/4$  e  $r = 1/5$  (cerca de 1.48%); exceto para  $r = 1/8$ , cuja variação é cerca de 20.35%.

Em Pinto et al. (2005) recomenda-se o uso de  $ITI = 2q$  para as equações lineares 1D de Poisson e advecção-difusão. Verificou-se pouca variação do tempo de CPU ao adotar  $ITI = 2q$ ; cerca de 3.50% para o problema de Poisson e cerca de 3.40% para o problema de advecção-difusão.

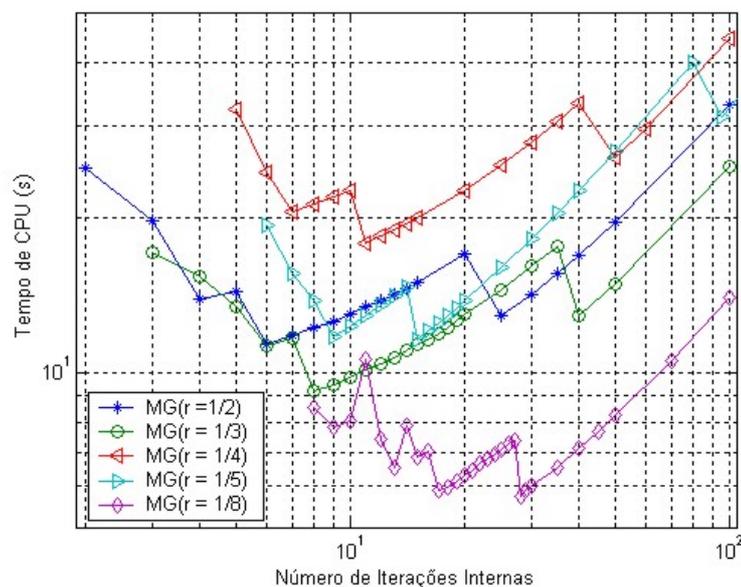


Figura 2 - Tempo de CPU em função do número de iterações internas

Tabela 1. Algumas malhas para cada uma das razões testadas

Razão	1/2	1/3	1/4	1/5	1/8
$N_f$	1,048,576	1,062,882	2,097,152	781,250	524,288

#### 4.2 Níveis de malha ( $L$ )

A Fig. 3 mostra a influência do número de níveis sobre o tempo de CPU e o número ótimo para as diversas razões de engrossamento e dimensões de problemas dadas pela Tab. 1 para a Equação de Burgers. Pode-se notar que o número de níveis pode afetar significativamente o tempo de CPU e que  $L_{otimo} \approx L_{maximo}$  para todas as razões testadas.

Nota-se na Fig. 3 que em geral o tempo de CPU tem uma taxa de decréscimo bastante acentuada até atingir os seus mínimos e, após estes pontos, a taxa de crescimento é bem menor até atingirem  $L_{maximo}$ . Outros testes numéricos mostraram que, relativamente ao ponto mínimo, uma diminuição sensível do número de níveis de malha aumenta drasticamente o tempo de CPU. Por outro lado, o aumento sensível do número de níveis de malha, aumenta pouco o tempo de CPU. Portanto recomenda-se utilizar  $L = L_{maximo}$  para as razões puras da forma  $r = 1/q$ ; por exemplo, para  $r = 1/2$  e  $N_f = 2^{20} = 1,048,576$ , recomenda-se usar  $L = 20$ , que é exatamente o  $L_{maximo}$ . A variação do tempo de CPU ao adotar-se  $L = L_{maximo}$  para o problema em questão e as diversas razões testadas é bastante pequena: na média cerca de 0.26%.

Em Pinto et al. (2005) obteve-se a mesma constatação acima para problemas lineares 1D, ou seja,  $L_{otimo} \approx L_{maximo}$ , e verificou-se uma pequena variação sobre o tempo de CPU de cerca de 0.90% para o problema de Poisson e 0.58% para o problema de advecção-difusão ao adotar  $L = L_{maximo}$ .

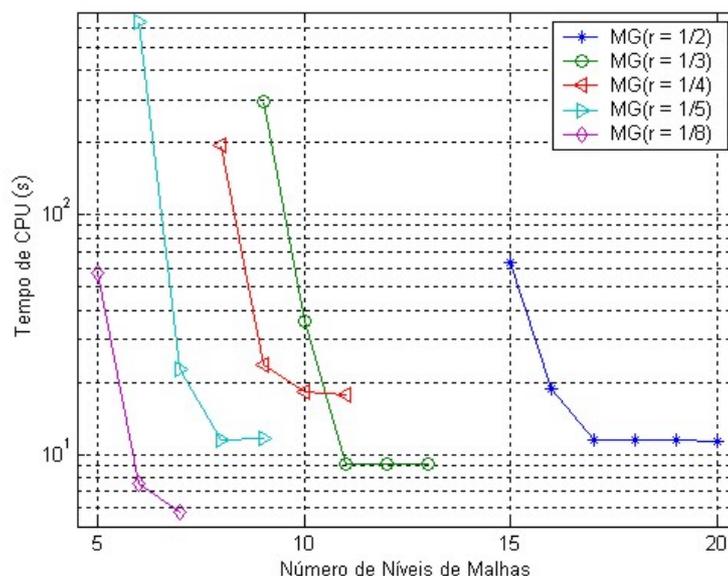


Figura 3 - Tempo de CPU em função do número de níveis de malhas

### 4.3 Razão de engrossamento

A Fig. 4 mostra as diversas razões de engrossamento para o método *Multigrid* e também uma comparação entre métodos *Singlegrid* Gauss-Seidel e TDMA em função de diversos  $N_f$  para a Equação de Burgers. Pode-se notar que o método TDMA é o mais eficiente de todos os métodos testados. Ele é seguido pelos métodos *Multigrid* e finalmente pelo método de Gauss-Seidel. Portanto, para o problema unidimensional testado, recomenda-se o uso do método direto TDMA. A Fig. 5 é sugerida para facilitar a visualização das curvas referentes às razões testadas nos métodos *Multigrid*.

Para os mesmos  $\varepsilon$ ,  $\bar{v}$  e  $N_f$ , dentre os métodos *Multigrid*, pode-se perceber que  $t_{CPU}(r=1/3) < t_{CPU}(r=1/4) < t_{CPU}(r=1/5) < t_{CPU}(r=1/2) < t_{CPU}(r=1/8)$ , na maioria dos pontos do domínio em estudo. Portanto, no caso do *Multigrid*, recomenda-se o uso da razão  $r=1/3$  para o problema em questão. Esta constatação é inédita e inesperada, além de mostrar que a razão  $r=1/2$  perde a hegemonia perante outras razões.

Brandt (1977) fez comparações entre as razões de engrossamento  $r=1/2$ ,  $r=1/3$  e  $r=2/3$  para diversos problemas, mas não se referiu à Eq. (8). Em seus estudos constatou que a razão  $r=1/2$  é a recomendável, por estar mais próxima do ótimo e ser mais conveniente e econômica para o processo de interpolação. Pode-se constatar também a boa eficiência da razão  $r=2/3$  para alguns de seus problemas. Briggs et al. (2000) afirmam que  $r \neq 1/2$ , em geral, não traz vantagens, sem especificar para quais problemas ou classes de problemas. Stüben (1999) desenvolveu um estudo com as razões  $r=1/2$  e  $r=1/4$  no caso do *Multigrid* algébrico, em que a razão  $r=1/4$  mostrou-se eficiente para os casos de anisotropia. Em Pinto et al. (2005) constatou-se que, dentre os métodos *Multigrid* testados,  $t_{CPU}(r=1/2) < t_{CPU}(r=1/3) < t_{CPU}(r=1/4) < t_{CPU}(r=1/5) < t_{CPU}(r=1/8)$ , na maioria dos

pontos do domínio em estudo, tanto para a equação de Poisson como para a de advecção-difusão, ambos problemas lineares 1D.

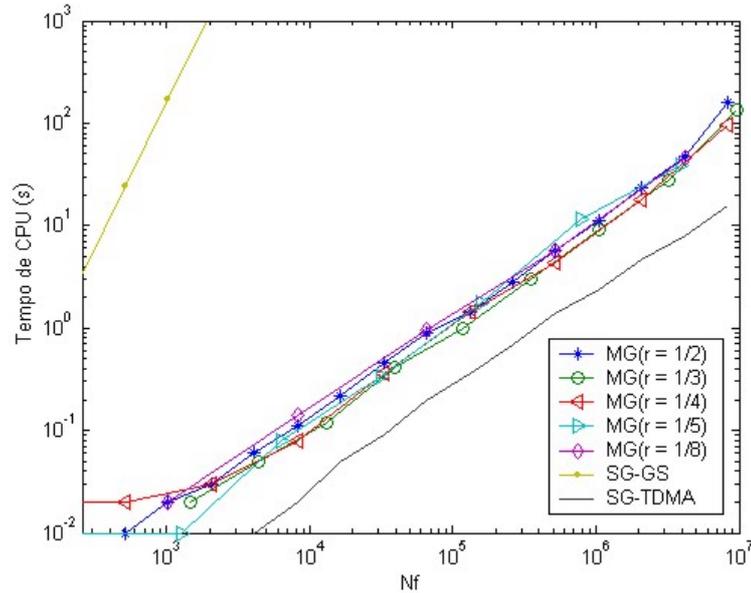


Figura 4 - Tempo de CPU em função da dimensão do problema para diversos métodos

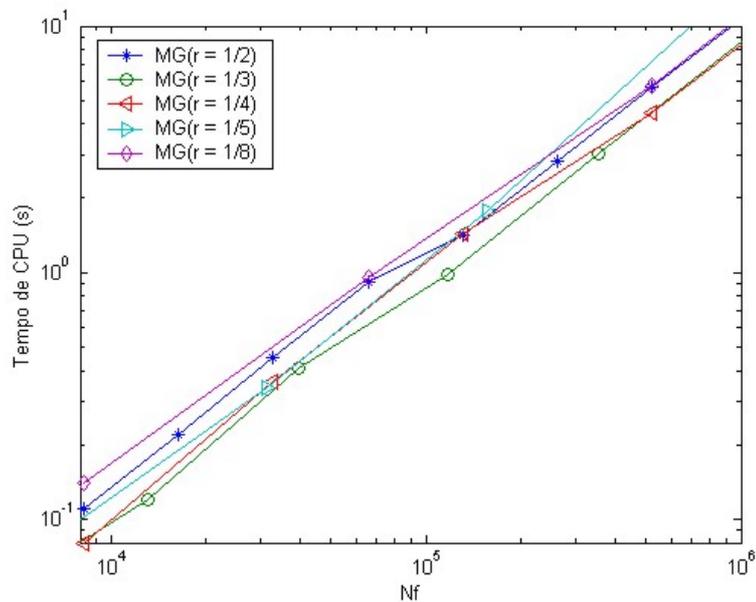


Figura 5 - Tempo de CPU em função da dimensão do problema para diversas razões para o método Multigrid

Fazendo-se um ajuste para as curvas da Fig. 4, com a curva geométrica  $t = bN_f^a$ , tem-se que:

- $a \approx 0.9206$  para o método TDMA;
- $a \approx 0.9994$  para os métodos Multigrid;
- $a \approx 2.8578$  para o método de Gauss-Seidel.

#### 4.4 Tolerância e estimativa inicial

Neste trabalho estudam-se também casos com variações tanto da tolerância  $\varepsilon$  como da estimativa inicial  $\vec{v}$ .

Para a variação da tolerância, estuda-se os casos onde  $\varepsilon = 10^{-4}$  e  $\varepsilon = 10^{-10}$ , além de  $\varepsilon = 10^{-7}$  usado nos resultados anteriores. Os resultados dos estudos do número de iterações internas ótimo e do número de níveis ótimo para a equação de Burgers sofreram pouca variação sobre o tempo de CPU: cerca de 10.81% ao se adotar  $ITI = 3q$  e, na média, cerca de 0.20% ao se adotar  $L = L_{maximo}$ .

Para a variação de estimativa inicial, estuda-se os casos onde  $\vec{v} = (1/2, 1/2, \dots, 1/2)$  e  $\vec{v} = (1, 1, \dots, 1)$ , além de  $\vec{v} = (0, 0, \dots, 0)$  usado nos resultados anteriores. Os resultados do estudo do número de iterações internas ótimo sofreram uma variação sobre o tempo de CPU mais substancial: variando entre 0.00% e 27.05% ao se adotar  $ITI = 3q$ . Os resultados do estudo do número de níveis ótimo sofreram pouca variação; na média, cerca de 0.11% ao se adotar  $L = L_{maximo}$ .

Isto mostra que o problema de minimização do tempo de CPU em função das iterações internas e do número de níveis é fracamente dependente da tolerância e fortemente dependente da estimativa inicial, pelo menos para a Equação de Burgers estudada.

A fraca dependência da tolerância e a forte dependência da estimativa inicial, também foram constatadas em Pinto et al. (2005). Verificou-se que a variação sobre o tempo de CPU, considerando a variação da tolerância, foi cerca de 2.40% ao adotar  $ITI = 2q$  e 0.80% se  $L = L_{maximo}$  para o problema de Poisson; e 3.40% com  $ITI = 2q$  e 0.61% se  $L = L_{maximo}$  para o problema de advecção-difusão. A variação sobre o tempo de CPU, considerando a variação da estimativa inicial, foi cerca de 28.70% com  $ITI = 2q$  e 0.90% se  $L = L_{maximo}$  para o problema de Poisson; e 22.00% fazendo  $ITI = 2q$  e 0.57% se  $L = L_{maximo}$  para o problema de advecção-difusão.

## 5. CONCLUSÃO

Neste trabalho estudou-se a influência da razão de engrossamento de malhas sobre o tempo de CPU do método *Multigrid* geométrico. O modelo matemático considerado, equação de Burgers, representa um escoamento não-linear unidimensional com condições de contorno de Dirichlet. A equação foi discretizada com o método de diferenças finitas e o método *Multigrid* usado é o Esquema de Aproximação Completa (*Full Approximation Scheme*) com ciclo V. Comparações foram feitas com dois métodos *Singlegrid*: Gauss-Seidel e TDMA.

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

- 1) O número de iterações internas e o número de níveis de malhas afetam significativamente o tempo de CPU. Recomenda-se usar  $ITI = 3q$  e  $L = L_{maximo}$  para as razões puras  $r = 1/q$  e qualquer  $N_f$ .
- 2) Os números ótimos de iterações internas e de níveis de malha são fracamente dependentes da tolerância especificada para o problema, mas fortemente dependentes da estimativa inicial.
- 3) Entre as razões de engrossamento de malhas testadas com o método *Multigrid*, para os mesmos  $\varepsilon$ , condição inicial e  $N_f$ , com  $L_{maximo}$  para cada  $N_f$  e  $ITI_{otimo}$  para cada  $r$ , tem-se em geral que  
 $t_{CPU}(r=1/3) < t_{CPU}(r=1/4) < t_{CPU}(r=1/5) < t_{CPU}(r=1/2) < t_{CPU}(r=1/8)$ . Ou seja, a razão  $r = 1/3$  resulta em menor tempo de CPU entre todas as razões consideradas. Segundo o conhecimento dos autores, este resultado é inédito, visto que não se encontrou na literatura algo semelhante.

### ***Agradecimentos***

Agradecemos o apoio do Laboratório de Experimentação Numérica (LENA) do Departamento de Engenharia Mecânica da UFPR por disponibilizar sua infra-estrutura. O primeiro autor agradece a Universidade Estadual de Ponta Grossa pelo suporte financeiro. Agradecemos também aos amigos do LENA e especialmente ao Luciano K. Araki.

### **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”, 2<sup>a</sup> ed., SIAM, 2000.
- Burden, R. L. and Faires, J. D., “Numerical Analysis”, 6<sup>a</sup> ed., Brooks/Cole Publishing Company, 1997.
- Dennis, J. E. and Schnabel, R., “Numerical Methods for Unconstrained Optimization and Nonlinear Equations”. Prattice Hall, 1983.
- 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”, 2<sup>a</sup> 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.
- Hirsch, C., “Numerical Computational of Internal and External Flows”, Vol.1, Wiley, 1998.
- Lai, Y. G. and Przekwa, A. J., “A *Multigrid* algorithm for a Multiblock pressure-based flow and heat transfer solver”, Numerical Heat Transfer, Part B, Vol. 30, pp. 239-254, 1996.
- Maliska, C. R., “Transferência de calor e mecânica dos fluidos computacional”, LTC, 2<sup>a</sup> ed., 2004.
- Mesquita, M. S. and De-Lemos, M. J. S., “Optimal *Multigrid* solutions of two-dimensional convection-conduction problems”, Applied Mathematics and Computation, Vol. 152, p. 725-742, 2004.
- Moro Filho, R. C., “Aplicação da Técnica *Multigrid* em Transferência de Calor Computacional”, XXV Iberian Latin\_American Congress on Computational Methods in Engineering, 2004.
- Perng, C. Y. and Street, R. L., “A Coupled *Multigrid*-Domain-Splitting technique for simulating incompressible flows in geometrically complex domains”, International Journal for Numerical Methods in Fluids, Vol. 13, pp. 269-286, 1991.
- Pinto, M. A. V., Santiago, C. D., Marchi, C. H., “Effect of Parameters of a Multigrid Method on CPU Time for One-dimensional Problems”, Proceedings of COBEM 2005.
- Roache, P. J., “Fundamentals of Computational Fluid Dynamics”, Hermosa Publishers, 1998.
- Stüben, K., “Algebraic *Multigrid* (AMG): an introduction with applications”, in: GMD-Report 70, November 1999.
- Stüben, K., “A Review of Algebraic *Multigrid*”, 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 *Multigrid* Methods”, John Wiley & Sons, 1992.
- Wesseling, P. and Oosterlee, C. W., “Geometric *Multigrid* with Applications to Computational Fluid Dynamics”, Journal of Computation and Applied Mathematics, Vol. 128, pp. 311-334, 2001.