



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

## **EFEITO DE PARÂMETROS DO MÉTODO MULTIGRID CS E FAS SOBRE O TEMPO DE CPU EM PROBLEMAS 1D LINEARES E NÃO-LINEARES**

**Fabiane de Oliveira**

**Márcio Augusto Villela Pinto**

*faboliveira@uepg.br*

*marciovp@demec.ufpr.br*

Departamento de Matemática e Estatística, Universidade Estadual de Ponta Grossa (UEPG)  
Ponta Grossa, PR - Brasil

**Cosmo Damião Santiago**

*cosmo@unibrasil.com.br*

Faculdades do Brasil (Unibrasil)

Curitiba, 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 dois problemas, verifica-se o efeito causado por: vários tamanhos de malhas, número de iterações internas, número de malhas, razão de engrossamento ( $r = 2, 3, 4$  e  $5$ ), tolerâncias, estimativas iniciais e esquemas de correção (CS) e de aproximação completa (FAS). Os problemas considerados envolvem um problema unidimensional linear (equação de advecção-difusão) e outro não-linear (equação de Burgers), ambos com condições de contorno de Dirichlet. O método de diferenças finitas é usado para discretizar as equações diferenciais. Os sistemas de equações algébricas são resolvidos com o método de Gauss-Seidel, associado ao método multigrid geométrico com ciclo V. Entre outros resultados, verificou-se que o esquema FAS é mais rápido do que o CS, para problemas lineares e não-lineares.*

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

## 1. INTRODUÇÃO

Modelos matemáticos na dinâmica dos fluidos computacional surgem em fenômenos físicos que envolvem fluidos em movimento, com ou sem troca de calor (Fortuna, 2000; Maliska, 2004). Estes modelos matemáticos, em geral, não têm soluções analíticas conhecidas. Buscam-se então soluções numéricas transformando-se o modelo contínuo em um modelo discreto. Um método de discretização 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:

$$x_i = (i-1)h, \quad i = 1, \dots, N_f + 1 \quad \text{e} \quad h = 1/N_f, \quad (1)$$

onde  $h$  é o comprimento de cada intervalo. Isto estabelece uma malha com elementos de tamanho  $h$  que denota-se por  $\Omega^h$ . A cada um dos  $N_f - 1$  pontos interiores da malha, a equação diferencial é substituída por aproximações de diferenças finitas (Tannehill et al., 1997; Ferziger e Peric, 1999).

A discretização desses modelos matemáticos resulta em grandes sistemas de equações algébricas do tipo

$$A\vec{v} = \vec{f}, \quad (2)$$

onde  $A$  é uma matriz quadrada,  $\vec{f}$  é o vetor independente e  $\vec{v}$  é o vetor de incógnitas. A estrutura da matriz  $A$  depende do método e das aproximações numéricas usadas para discretizar o modelo matemático.

Várias técnicas numéricas têm sido estudadas para resolver o sistema dado pela Eq. (2) com 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). Para problemas de grande porte os métodos iterativos são mais adequados (Burden e Faires, 1997). O método do gradiente conjugado (Burden e Faires, 1997; Golub e Ortega, 1992), introduzido por Hestenes e Stiefeld (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. (2). 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 que transferem informações da malha fina para a malha imediatamente mais grossa (processo chamado de restrição) e da malha grossa para a malha imediatamente mais fina (processo de prolongação). Briggs et al. (2000) trabalharam com a razão de aspecto  $r = 2$  afirmando ser uma prática universal e que  $r \neq 2$  não traz vantagens. Os sistemas lineares são resolvidos com um método iterativo com propriedades de reduzir rapidamente os erros oscilatórios (propriedades de suavização). Com este conceito, vários trabalhos (Brandt, 1977; Stüben, 1999; Wesseling e Oosterlee, 2001; Moro, 2004; Pinto et al., 2005a), apresentaram bons resultados numéricos para problemas de dinâmica dos fluidos. Os resultados de Fedorenko (1964) mostram que a taxa de velocidade de convergência com o uso do método *multigrid* é muito melhor que a dos métodos iterativos puros. O objetivo do método *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, conhecido como *Multigrid* geométrico (Wesseling e Oosterlee, 2001), bem como a malhas não-estruturadas, conhecido como *Multigrid* algébrico (Stüben, 2001). 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 alongadas, ou mesmo a paralelização de algoritmos.

A razão de engrossamento, para o caso unidimensional e considerando malhas uniformes, é definida como:

$$r = \frac{h_{\Omega^H}}{h_{\Omega^h}}, \quad (3)$$

onde  $\Omega^h$  representa uma malha fina,  $\Omega^H$  a malha imediatamente mais grossa e  $h$  o tamanho dos elementos da malha.

Brandt (1977) trabalhou com o método *multigrid* geométrico em diversos problemas de transferência de calor e escoamento de fluidos, unidimensionais e bidimensionais, lineares e não-lineares. Em seu trabalho fez comparações com as razões de engrossamento  $r = 2, 3$  e  $3/2$ . Para os problemas testados, Brandt (1977) concluiu que a razão  $r = 2$  é a recomendável. Stüben (1999) desenvolveu um estudo com as razões  $r = 2$  e  $4$  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  $r = 4$  se mostrou eficiente para problemas anisotrópicos (anisotropia dos coeficientes e anisotropia devido à malhas fortemente alongadas). Briggs et al. (2000) afirmam que a razão  $r = 2$  é uma prática universal e que  $r \neq 2$  não traz vantagens. Moro (2004) trabalhou com as razões  $r = 2$  e  $4$  em malhas estruturadas para problemas de difusão com termo fonte. Em seu trabalho a razão  $r = 4$  apresentou tempo de CPU menor que a razão  $r = 2$ . Pinto et al. (2005a) e Pinto et al. (2005b) trabalharam com as razões  $r = 2, 3, 4, 5$  e  $8$  para problemas unidimensionais lineares (equação de difusão e equação de advecção-difusão) com o uso do esquema de correção (CS) e não-linear (equação de Burgers) com o uso do esquema de aproximação completa (FAS). Nestes trabalhos as razões  $r = 2$  e  $3$  resultaram em menor tempo de CPU para os esquemas CS e FAS, respectivamente. Neste trabalho estuda-se o método *multigrid* em uma dimensão para mostrar os princípios do método e derivar alguns procedimentos usados nos casos mais gerais (Ferziger e Peric, 1999). Além disto, uma maior quantidade de testes pode ser feita devido à rapidez de obtenção das soluções, facilitando o estudo de uma grande variação de parâmetros.

Neste artigo os seguintes parâmetros são estudados: número de iterações internas (número de iterações do método numérico a fim de suavizar as componentes de erro), o número de níveis (número de malhas percorridas) e diversas razões de engrossamento ( $r = 2, 3, 4$  e  $5$ ). Os objetivos são: verificar o efeito desses parâmetros sobre o tempo de CPU para o *multigrid* geométrico e realizar comparações entre os esquemas CS e FAS com ciclo V, proposto em Wesseling (1992). Os resultados são comparados com os obtidos na bibliografia. São apresentados operadores de restrição por injeção e prolongação por interpolação linear para qualquer razão de engrossamento no intervalo  $(1, \infty)$ . Os modelos matemáticos considerados neste trabalho envolvem um problema unidimensional linear de transferência de calor, ou seja, a equação de advecção-difusão, e um problema unidimensional não-linear de escoamento, ou seja, a equação de Burgers, ambos com condições de contorno de Dirichlet.

Este artigo está organizado da seguinte forma: na seção 2 é apresentada uma visão geral do método *multigrid*, incluindo os operadores de restrição e prolongação. Na seção 3, são apresentados os modelos matemáticos e numéricos. Na seção 4 são descritos os experimentos numéricos e seus resultados. E na seção 5 é apresentada 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. Uma opção 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 para a solução 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 da Eq. (2) em cada malha, definido por:

$$\vec{R} = \vec{f} - A.\vec{v} \quad (4)$$

é suavizar as oscilações por um método de relaxação (método iterativo). 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).

Podem ser usados dois tipos de esquemas com o método *multigrid* (Briggs et al., 2000): o esquema de correção (*Correction Scheme*, CS) e o esquema de aproximação completa (*Full Approximation Scheme*, FAS). No esquema CS, a Eq. (2) é resolvida apenas na malha mais fina; nas malhas grossas, resolve-se a equação do resíduo. Já no caso do esquema FAS, a Eq. (2) é resolvida em todas as malhas. O esquema CS é geralmente utilizado em problemas lineares e o esquema FAS, em problemas não-lineares.

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 um bom desempenho do *multigrid*, diversos níveis de malhas devem ser usados (Tannehill et al., 1997). Pinto et al. (2005a e 2005b) recomendam usar todos os níveis.

A razão de engrossamento para malhas uniformes é dada pela Eq. (3), onde:  $r \in (1,2)$  é denominado engrossamento fraco;  $r = 2$ , engrossamento padrão e  $r \in (2, \infty)$ , engrossamento forte. Neste trabalho estuda-se o engrossamento forte e engrossamento padrão, ambos aplicados aos esquemas CS e FAS encontrados em Wesseling (1992).

Uma forma alternativa da Eq. (3) é:

$$r = \frac{p}{q}, \quad p, q \in \mathbb{Z}_+^*, \quad p > q. \quad (5)$$

Neste caso, a razão de engrossamento  $r$  é chamada de razão pura se  $q = 1$ . Para as razões da forma  $r = p/q = p/1$ , neste artigo utiliza-se  $p = r$ .

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  $\vec{\phi}$ , no caso linear, assume o resíduo  $\vec{R}$  dado pela Eq. (4) e no caso não-linear assume a solução aproximada do problema além do resíduo  $\vec{R}$ . Em Pinto et al. (2005a) desenvolveu-se um operador de injeção com a sua forma generalizada para qualquer  $r \in (1, \infty)$ , dado por:

$$\vec{\phi}_i^H = K_r \vec{\phi}_{cr}^h + (1 - K_r) \vec{\phi}_{cr}^h; \quad 2 \leq i \leq N^H, \quad (6)$$

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

$$\text{ceiling} : \mathfrak{R} \rightarrow \mathbb{Z}, \quad \text{com } x \mapsto \text{ceiling}(x) = \min\{z \in \mathbb{Z} / z \geq x\}. \quad (7)$$

A Eq. (6) não é calculada para  $i = 1$  e  $N^H + 1$ , pois no presente trabalho são usadas 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  $\vec{\phi}$ , no caso linear assume aproximação do erro na equação residual, ou seja, a correção, e no caso não-linear assume a solução aproximada do problema, além da correção. Como no caso do operador de restrição, em Pinto et al. (2005a) desenvolveu-se um operador de interpolação linear com a sua forma generalizada para qualquer  $r \in (1, \infty)$ , dada por:

$$\vec{\phi}_i^h = K_p \vec{\phi}_{cp}^H + (1 - K_p) \vec{\phi}_{cp+1}^H; \quad 2 \leq i \leq N^h, \quad (8)$$

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

### 3. MODELOS MATEMÁTICOS E NUMÉRICOS

#### 3.1 Modelo linear: equação de advecção-difusão

A convecção de calor unidimensional (equação de advecção-difusão), com condições de contorno de Dirichlet, em regime permanente e em coordenadas cartesianas pode ser representada matematicamente por (Ferziger e Peric, 1999):

$$Peu_x = u_{xx}, \quad 0 < x < 1; \quad u(0) = 0, \quad u(1) = 1, \quad (9)$$

onde  $u$  é a incógnita e  $Pe = 20$  é o número adimensional de Peclet. Na Eq. (9) a variável  $u$  representa a temperatura, e suas derivadas de primeira e segunda ordens são representadas por  $u_x$  e  $u_{xx}$ , respectivamente. Para as condições de contorno acima a solução analítica é dada por:

$$u(x) = \frac{e^{xPe} - 1}{e^{Pe} - 1}. \quad (10)$$

A discretização do domínio é feita com malhas uniformes em  $N_f$  subintervalos (elementos) introduzidos pelos pontos da malha dados pela Eq. (1). A equação diferencial dada pela Eq. (9) é discretizada de acordo com o método de diferenças finitas. Nesta equação são utilizadas a diferença à montante (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). Com estas aproximações a Eq. (9) resulta em

$$Pe \frac{v_i - v_{i-1}}{h} = \frac{v_{i-1} - 2v_i + v_{i+1}}{h^2}, \quad 2 < i < N_f; \quad v_1 = 0, v_{N_f+1} = 1, \quad (11)$$

onde  $v_i$  é uma aproximação para a solução exata  $u(x_i)$ . Rearranjando os termos da Eq. (11), obtém-se a forma geral da equação discretizada, dada por:

$$a_p v_i = a_w v_{i-1} + a_e v_{i+1} + b_p, \quad (12)$$

onde os coeficientes são dados por:

$$a_p = \frac{Pe}{h} + \frac{2}{h^2}, \quad a_w = \frac{1}{h^2} + \frac{Pe}{h}, \quad a_e = \frac{1}{h^2} \quad \text{e} \quad b_p = 0. \quad (13)$$

Se  $\vec{v}$  e  $\vec{f}$  são denotados por  $\vec{v} = (v_1, \dots, v_{N_f+1})^t$  e  $\vec{f} = (f_1, \dots, f_{N_f+1})^t$ , respectivamente, então o sistema dado pela Eq. (12), pode ser representado por um sistema de equações algébricas do tipo da Eq. (2) onde  $A$  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 formado por  $b_p$  e  $\vec{v}$  é o vetor de incógnitas.

A Eq. (2) é resolvida com o método *multigrid* fazendo-se uso dos dois esquemas (CS e FAS) citados na seção anterior. Neste caso, os sistemas de equações do tipo da Eq. (2), onde  $\vec{f}$  agora representa o termo fonte (resíduo ou solução aproximada) a cada nível de malha, são resolvidos apenas com o método de Gauss-Seidel. Resolve-se também o problema, apenas na malha mais fina, com os métodos *Singlegrid*: Gauss-Seidel e TDMA (Maliska, 2004). O número de iterações internas (*ITI*) do método *multigrid* é o número de iterações do *solver* (método numérico) a fim de suavizar os componentes do erro.

### 3.2 Modelo não-linear: equação de Burgers

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

$$Re u_x^2 = u_{xx} + S, \quad 0 < x < 1; \quad u(0) = 0, u(1) = 1, \quad (14)$$

onde  $u$  é a incógnita e representa a velocidade,  $S$  é o termo fonte dado por  $S = Re^2 e^{xRe} (2e^{xRe} - e^{Re} - 1)/(e^{Re} - 1)^2$ , e  $Re = 20$  é o número adimensional de Reynolds. Para estas condições de contorno e termo fonte, a solução analítica é dada por:

$$u(x) = \frac{e^{xRe} - 1}{e^{Re} - 1}. \quad (15)$$

Considerando-se o mesmo modelo numérico da seção 3.1, a discretização da Eq. (14) resulta em

$$Re \frac{v_i^2 - v_{i-1}^2}{h} = \frac{v_{i-1} - 2v_i + v_{i+1}}{h^2} + S_i, \quad 2 \leq i \leq N_f; \quad v_1 = 0, v_{N_f+1} = 1, \quad (16)$$

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. (16), como em Ferziger e Peric (1999), é da forma:

$$Re \frac{v_i^* v_i - v_{i-1}^* v_{i-1}}{h} = \frac{v_{i-1} - 2v_i + v_{i+1}}{h^2} + S_i, \quad 2 \leq i \leq N_f; \quad v_1 = 0, v_{N_f+1} = 1, \quad (17)$$

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. (17), os coeficientes da Eq. (12) resultam em

$$a_p = (Re v_i^*)/h + 2/h^2, \quad a_w = (Re v_{i-1}^*)/h + 1/h^2, \quad a_e = 1/h^2 \quad \text{e} \quad b_p = S_i. \quad (18)$$

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 após a linearização pode ser representado por um sistema de equações algébricas do tipo da Eq. (2), 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 formado por  $b_p$  e  $\vec{v}$ , o vetor de incógnitas.

## 4. RESULTADOS

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

Centenas de simulações foram realizadas com as seguintes variantes: número de incógnitas ( $N_f$ ), número de iterações internas ( $ITI$ ), número de níveis de malhas ( $L$ ), tolerâncias ( $\varepsilon$ ), estimativas iniciais ( $\vec{v}$ ) e razões de engrossamento  $r = 2$  (padrão), 3, 4 e 5. São apresentados neste trabalho os resultados mais representativos.

O critério de convergência para as iterações externas  $ITE$  (número necessário de ciclos para suavizar as componentes de erro) 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. (4). Adotou-se como referência  $\varepsilon = 10^{-7}$  e  $\vec{v} = (0,0,\dots,0)$  para a tolerância sobre o critério de convergência e estimativa inicial, respectivamente.

O foco deste trabalho é a minimização do tempo de CPU, para isto, busca-se o número ótimo de iterações internas ( $ITI_{\text{ótimo}}$ ), número ótimo de níveis de malha ( $L_{\text{ótimo}}$ ) a cada ciclo, a razão de engrossamento ótima e o esquema ótimo (CS ou FAS). Entende-se por tempo de CPU, o tempo gasto para gerar malhas, atribuir estimativa inicial, calcular os coeficientes e resolver o sistema da Eq. (2). 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 Esquema de correção (CS)

**Iterações internas ( $ITI$ ).** A Fig. 1 mostra a influência do número de iterações internas ( $ITI$ ) sobre o tempo de CPU para a equação de Burgers. O número de iterações internas influencia o tempo de CPU e  $ITI_{\text{ótimo}} = 2r - 1$  para a equação advecção-difusão e  $ITI_{\text{ótimo}} \approx 2r$  para a equação de Burgers. Este resultado concorda com o de Tannehill et al. (1997) para  $r = 2$  no caso da equação de Laplace bidimensional linear, onde  $ITI_{\text{ótimo}} = 3, 4$ .

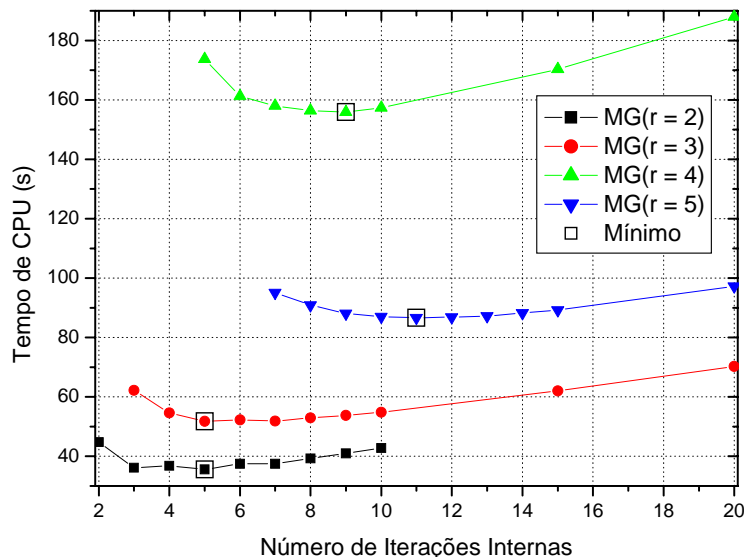


Figura 1: Tempo de CPU versus  $ITI$  para esquema CS e equação de Burgers.

Verificou-se que para todas as razões estudadas, uma diminuição pequena do número de iterações internas, relativamente ao ponto mínimo, aumenta drasticamente o tempo de CPU, podendo até mesmo ocorrer divergência. Por outro lado, um aumento pequeno do número de



iterrações internas, relativamente ao ponto de m nimo, aumenta pouco o tempo de CPU (Pinto et al., 2005a). Portanto, recomenda-se utilizar  $ITI = 2r$  para as raz es puras; por exemplo, para  $r = 2$ , recomenda-se  $ITI = 4$  para ambos os problemas. Os resultados do estudo feito para a varia o do tempo de CPU ao adotar-se o n mero  $ITI = 2r$  para o problema em quest o e para as diversas raz es de engrossamento estudadas sofreram, em m dia, pouca varia o, cerca de 3.40% para a equa o de advec o-difus o e 2.39% para a equa o de Burgers.

Seja o problema de minimiza o do tempo de CPU, em fun o do n mero de iterrações internas, para as diversas raz es de engrossamento no esquema CS. Para este problema, verificou-se que o n mero  timo de iterrações internas tem uma forte influ ncia do n mero de vezes que o m todo de Gauss-Seidel necessita para atualizar o res duo em todo o dom nio unidimensional.

**N veis de malhas ( $L$ ).** A Fig. 2 mostra a influ ncia do n mero de n veis de malhas ( $L$ ) sobre o tempo de CPU para a equa o de Burgers. Verificou-se que o n mero de n veis pode afetar significativamente o tempo de CPU com o uso do esquema CS para as diversas raz es de engrossamento estudadas, tanto para a equa o de advec o-difus o, quanto para a equa o de Burgers e que  $L_{\acute{o}timo} \approx L_{m ximo}$ , onde  $L_{m ximo}$  representa o n mero m ximo poss vel de n veis. Este resultado   similar aos resultados de Tannehill et al. (1997), no caso da equa o de Laplace bidimensional (caso linear); e Mesquita e De-Lemos (2004), no caso da equa o de Navier-Stokes compress vel bidimensional (caso n o-linear), ambos para o caso onde  $r = 2$ .

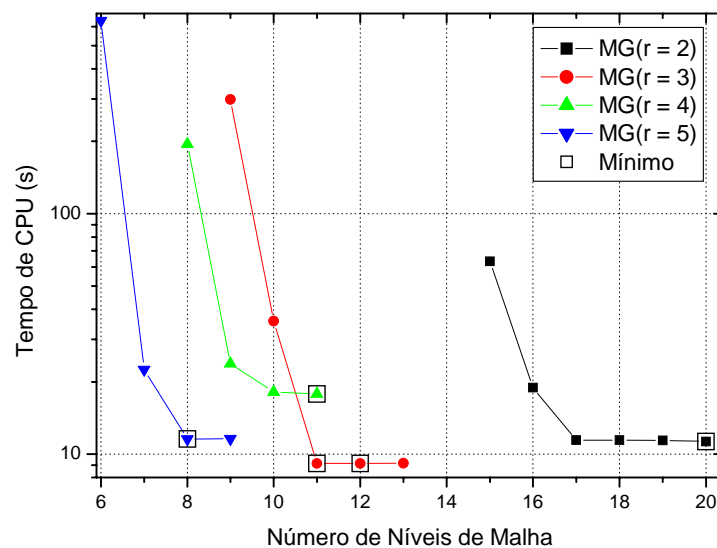


Figura 2: Tempo de CPU *versus*  $L$  para esquema CS e equa o de Burgers.

Para ambos os problemas e para todas as raz es testadas o tempo de CPU tem uma taxa de decr scimo bastante acentuada at  atingir os seus m nimos e, ap s estes pontos, a taxa de crescimento   bem menor at  atingirem  $L_{m ximo}$ . Portanto, recomenda-se utilizar  $L = L_{m ximo}$  para as raz es puras; por exemplo, na equa o de advec o-difus o, para  $r = 2$  e  $N_f = 2^{20} = 1048576$  elementos, recomenda-se usar  $L = 20$ , que   exatamente o  $L_{m ximo}$ . Os resultados do estudo feito para a varia o do tempo de CPU ao adotar-se  $L = L_{m ximo}$  para os problemas em quest o e para as diversas raz es de engrossamento estudadas   bastante

pequena, na média, cerca de 0.90% para a equação de advecção-difusão e 0.24% para a equação de Burgers.

Seja o problema de minimização do tempo de CPU, em função do número de níveis, para as diversas razões de engrossamento para o esquema CS. Para este problema, verificou-se que o número ótimo de níveis tem uma forte influência da velocidade com que é resolvido o problema em uma quantidade maior de malhas sucessivamente mais grossas, ou seja, malhas com uma quantidade de pontos menor.

**Razão de engrossamento ( $r$ ).** As Figs. 3 e 4 mostram o tempo de CPU para as diversas razões de engrossamento para o método *multigrid*, e também uma comparação entre os métodos *singlegrid* Gauss-Seidel e TDMA, em função de diversos  $N_f$  para a equação de advecção-difusão e para a equação de Burgers, ambas com o esquema CS. Pode-se notar que o método TDMA é o mais eficiente de todos os métodos testados para os problemas. Ele é seguido pelos métodos *multigrid* e finalmente pelo método de Gauss-Seidel. Portanto, para os problemas unidimensionais testados, recomenda-se o uso do método direto TDMA. Entretanto o objetivo nesta seção e na próxima é verificar o efeito dos parâmetros *multigrid* no tempo de CPU para o *multigrid* geométrico com o esquema CS e FAS (próxima seção) em problemas unidimensionais lineares e não-lineares a fim de se obter informações para outros problemas.

Para os mesmos  $\varepsilon$ ,  $\bar{v}$  e  $N_f$ , com as razões de engrossamento testadas e o método *multigrid* esquema CS, pode-se perceber que para os problemas testados (equação de advecção-difusão e equação de Burgers), tem-se:

$$t_{CPU}(r = 2) < t_{CPU}(r = 3) < t_{CPU}(r = 4) < t_{CPU}(r = 5). \quad (19)$$

Portanto, no caso do método *multigrid* com o esquema CS, recomenda-se o uso da razão  $r = 2$  para as equações em questão.

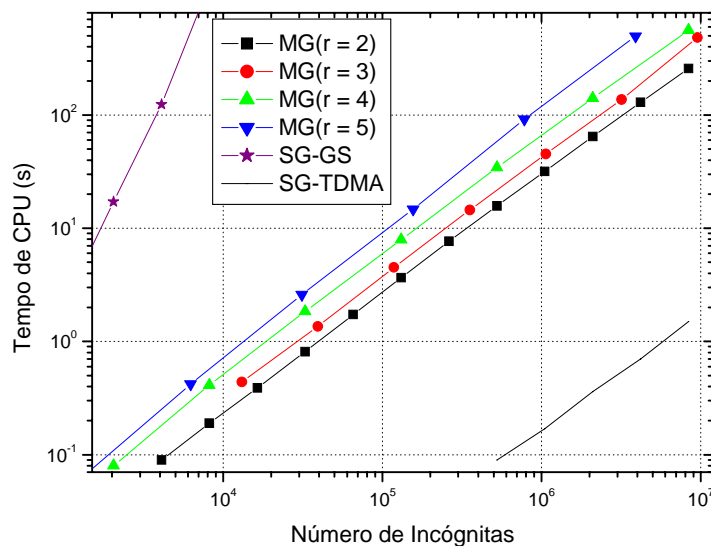


Figura 3: Tempo de CPU *versus*  $N$  para esquema CS e equação de advecção-difusão.

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

Para a variação da tolerância, estudam-se 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, tanto para a equação de advecção-difusão como para a equação de Burgers sofreram pouca variação sobre o tempo de CPU: cerca de 3.4% e 0.5% respectivamente, ao se adotar  $ITI = 2r$ .

Para a variação da estimativa inicial, estudam-se casos onde  $\vec{v} = (1/2, 1/2, \dots, 1/2)$ ,  $\vec{v} = (1, 1, \dots, 1)$ , além de  $\vec{v} = (0, 0, \dots, 0)$  usado nos resultados anteriores. Os resultados dos estudos do número de iterações internas ótimo tanto para a equação de advecção-difusão como para a equação de Burgers sofreram uma variação sobre o tempo de CPU mais substancial: 3.4% para a equação de advecção-difusão e 15.75% para a equação de Burgers ao se adotar  $ITI = 2r$ .

Em relação ao número de níveis, ao adotar-se  $L_{\text{ótimo}} = L_{\text{máximo}}$ , a maior variação encontrada foi de 0.61% para a equação de advecção-difusão e 6.33% para a equação de Burgers, independente de estar variando a tolerância ou a estimativa inicial.

Isto mostra que o problema de minimização do tempo de CPU é fracamente dependente da tolerância e fortemente dependente da estimativa inicial quando se estuda o número de iterações internas; e fracamente dependente da tolerância e estimativa inicial quando se estuda o número de níveis. Conclui-se que pode-se usar  $ITI_{\text{ótimo}} = 2r$  e  $L_{\text{ótimo}} = L_{\text{máximo}}$  para ambos os problemas.

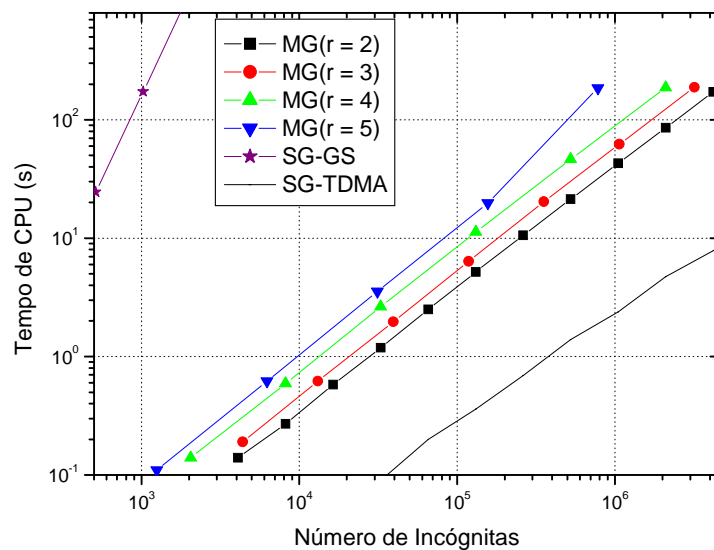


Figura 4: Tempo de CPU *versus*  $N$  para esquema CS e equação de Burgers.

## 4.2 Esquema de aproximação completa (FAS)

**Iterações internas (ITI).** Através da Fig. 1 concluiu-se que o número de iterações internas influencia o tempo de CPU para o esquema CS. O mesmo ocorre com o uso do esquema FAS, sendo  $ITI_{\text{ótimo}} \approx 3r$ .

Para todas as razões estudadas, uma diminuição sensível do número de iterações internas, relativamente ao ponto mínimo, aumenta drasticamente o tempo de CPU, podendo até mesmo ocorrer divergência. Por outro lado, o aumento pequeno do número de iterações internas, relativamente ao ponto mínimo, aumenta sensivelmente o tempo de CPU (Pinto et al. 2005b). Verificou-se que  $ITI = 3r$  para as razões puras; por exemplo, para  $r = 2$ , recomenda-se  $ITI = 6$

para ambos os problemas. Os resultados do estudo feito para a variação do tempo de CPU ao adotar-se o número  $ITI = 3r$  para o problema de advecção-difusão sofreram, em média pouca variação, cerca de 4.96% para a razão  $r = 3$ , não se cometendo erro nas demais razões. Para a equação de Burgers o erro variou de 0 a 2.98% para todas as razões estudadas.

Seja o problema de minimização do tempo de CPU em função do número de iterações internas para as diversas razões de engrossamento para o esquema FAS. Para este problema, verificou-se que o número ótimo de iterações internas tem uma forte influência do número de vezes que o método de Gauss-Seidel necessita para atualizar o resíduo e a solução em todo o domínio unidimensional.

Para o problema linear unidimensional estudado na seção anterior, verificou-se  $ITI_{\text{ótimo}} \approx 2r$  para o esquema CS. Aqui, para o mesmo problema, verificou-se que  $ITI_{\text{ótimo}} \approx 3r$  para o esquema FAS. Portanto, o esquema utilizado (CS ou FAS) influencia o número ótimo de iterações internas para o método *multigrid*.

**Níveis de malhas ( $L$ ).** Através da Fig. 2, concluiu-se que o número de níveis influencia o tempo de CPU para o esquema CS. Verificou-se o mesmo resultado com o esquema FAS e que  $L_{\text{ótimo}} \approx L_{\text{máximo}}$  para as diversas razões de engrossamento estudadas, tanto para a equação de advecção-difusão, bem como para a equação de Burgers.

Para ambos os problemas e para todas as razões estudadas, 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  $L_{\text{máximo}}$ . (Pinto et al, 2005b). Portanto recomenda-se utilizar  $L = L_{\text{máximo}}$  para as razões puras. Não se comete erros no tempo de CPU ao adotar-se  $L_{\text{ótimo}} = L_{\text{máximo}}$  tanto para a equação de advecção-difusão como para a equação de Burgers.

Para os problemas unidimensionais estudados na seção anterior, verificou-se que  $L_{\text{ótimo}} \approx L_{\text{máximo}}$  para o esquema CS. Nesta seção, para os mesmos problemas, encontrou-se o mesmo resultado. Portanto o esquema utilizado (CS ou FAS) não influencia o número ótimo de níveis para o método *multigrid*.

**Razão de engrossamento ( $r$ ).** As Figs. 5 e 6 mostram o tempo de CPU para as razões de engrossamento consideradas para o método *multigrid* e também uma comparação entre métodos *Singlegrid* Gauss-Seidel e TDMA em função de diversos  $N_f$  para as equações de advecção-difusão e de Burgers, respectivamente com o esquema FAS.

As Figs. 7 e 8 mostram as Figs. 5 e 6 ampliadas para facilitar a visualização das curvas referentes às razões testadas nos métodos *multigrid*.

Pode-se notar novamente que o método TDMA é o mais eficiente entre os métodos testados para ambas as equações. Ele é seguido pelos métodos *multigrid* e pelo método de Gauss-Seidel.

Para os mesmos  $\varepsilon$ ,  $\vec{v}$  e  $N_f$ , com as razões de engrossamento testadas e o método *multigrid* com o esquema FAS, pode-se perceber que para ambas equações estudadas, para a maioria dos pontos do domínio em estudo, tem-se:

$$t_{CPU}(r=3) < t_{CPU}(r=4) < t_{CPU}(r=2) < t_{CPU}(r=5) \quad (20)$$

Portanto, no caso do método *multigrid* com o uso do esquema FAS, recomenda-se o uso da razão  $r = 3$  para ambas equações em estudo. Esta constatação é inédita e inesperada, além de mostrar que a razão  $r = 2$  perde a hegemonia perante outras razões.

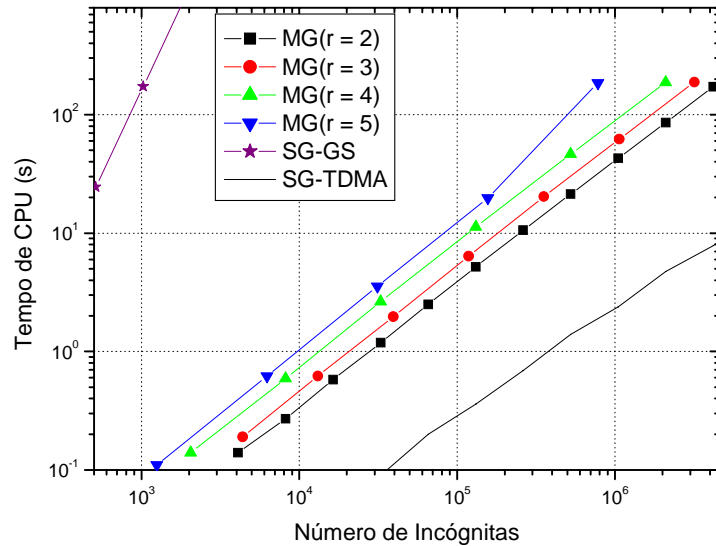


Figura 5: Tempo de CPU *versus*  $N$  para esquema FAS e equação de advecção–difusão.

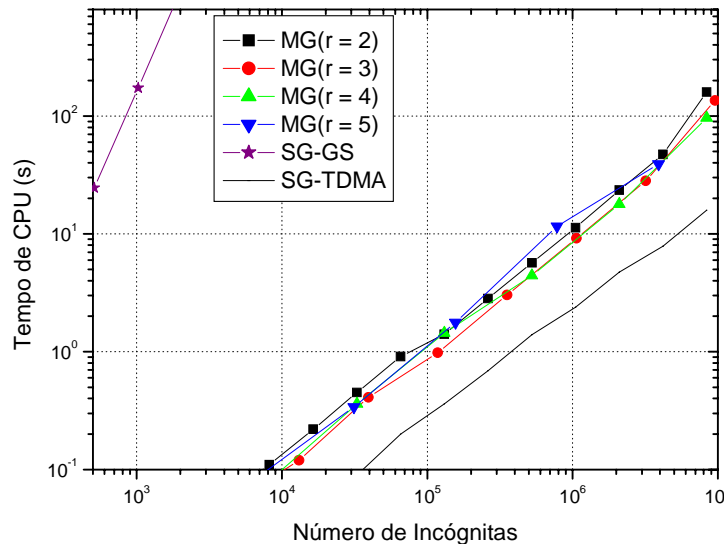


Figura 6: Tempo de CPU *versus*  $N$  para esquema FAS e equação de Burgers.

Seja o problema de minimização do tempo de CPU em função das diversas razões de engrossamento e com o uso do esquema FAS. Para estes problemas, verificou-se que a razão ótima tem uma forte influência dos seguintes aspectos: a razão de queda do número de pontos de uma malha para a outra malha imediatamente mais grossa, a qualidade das informações transferidas pela restrição e pela prolongação e, principalmente, o uso do esquema FAS, que transfere informações a respeito do resíduo e da solução para as demais malhas mais grossas.

Para os problemas unidimensionais estudados na seção anterior, verificou-se que  $r = 2$  é a melhor razão de engrossamento para o esquema CS. Aqui, para os mesmos problemas, verificou-se que  $r = 3$  é a melhor para o esquema FAS. Pode-se concluir que o esquema utilizado (CS ou FAS) influencia a razão ótima para o método *multigrid*.

**Tolerância e estimativa inicial.** Neste trabalho estudam-se também casos com variações tanto da tolerância  $\varepsilon$  como da estimativa inicial  $\vec{v}$  como na secção anterior.

Para a variação da tolerância, os resultados dos estudos do número de iterações internas ótimo, para os problemas estudados, sofreram variação substancial sobre o tempo de CPU ao

se adotar  $ITI = 3r$ . Cerca de 6.87% a 24.53% para a equação de advecção-difusão e 10.81% para a equação de Burgers.

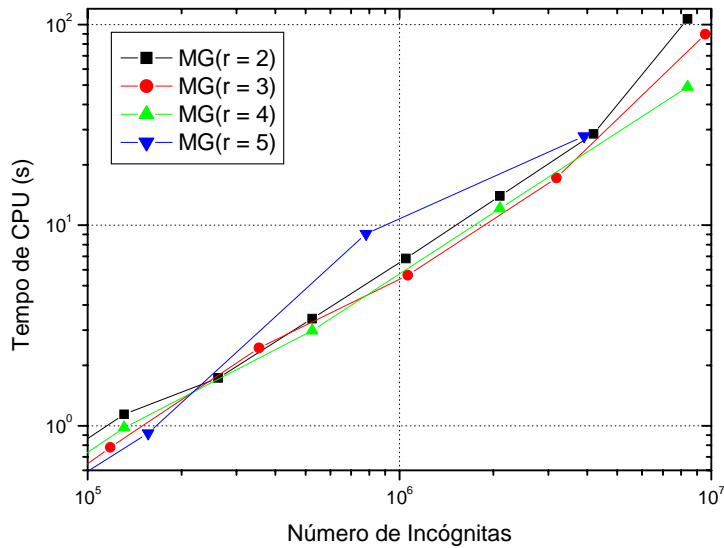


Figura 7: Ampliação da Fig. 5.

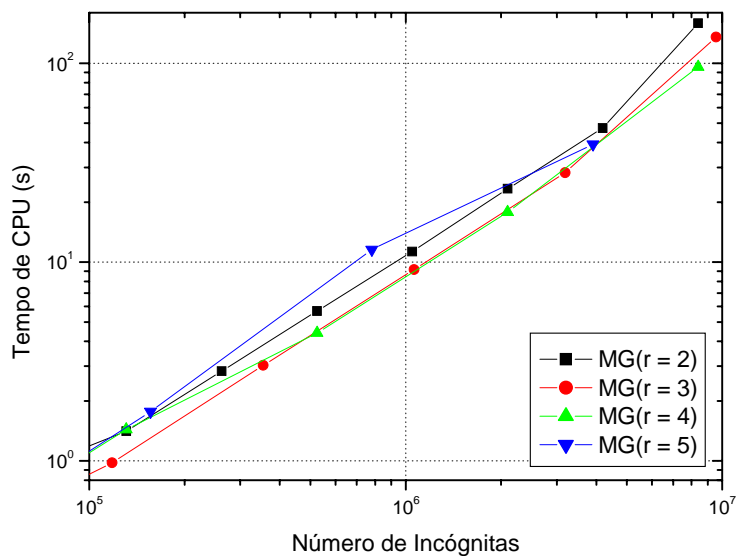


Figura 8: Ampliação da Fig. 6.

Para a variação da estimativa inicial, os resultados dos estudos do número de iterações internas ótimo tanto para a equação de advecção-difusão quanto para a equação de Burgers sofreram uma variação significativa sobre o tempo de CPU: de 11.93% a 32.65% para a equação de advecção-difusão e de 0 a 27.05% para a equação de Burgers ao se adotar  $ITI = 3r$ .

Em relação ao número de níveis, ao adotar-se  $L_{\text{ótimo}} = L_{\text{máximo}}$ , a maior variação encontrada foi de 24.78% para a equação de advecção-difusão e 0.65% para a equação de Burgers, independente de estar variando a tolerância ou a estimativa inicial.

Isto mostra que, como na seção anterior, o problema de minimização do tempo de CPU é fracamente dependente da tolerância e fortemente dependente da estimativa inicial quando se estuda o número de iterações internas; e fracamente dependente da tolerância e estimativa

inicial quando se estuda o número de níveis. Conclui-se que, pode-se usar  $ITI_{\text{ótimo}} = 3r$  e, como na seção anterior  $L_{\text{ótimo}} = L_{\text{máximo}}$  para ambos os problemas.

### 4.3 Esquema CS versus FAS

Nas Figs. 9 e 10 se faz uma comparação entre os esquemas *multigrid* CS e FAS em função de diversos valores de  $N_f$ . Pode-se notar que o esquema FAS é mais rápido do que o CS em qualquer  $N_f$ . Por exemplo, a equação de Burgers para a malha de 4194304 elementos o tempo de CPU do MG-FAS( $r = 2$ ) e MG-CS( $r = 2$ ) é de 47.25 s e 142.47 s, respectivamente, ou seja, o MG-FAS( $r = 2$ ) é cerca e 3 vezes mais rápido do que o MG-CS( $r = 2$ ). Outro exemplo, para a malha de 3188646 elementos o tempo de CPU do MG-FAS( $r = 3$ ) e MG-CS( $r = 3$ ) é de 28.19 s e 156.74 s, respectivamente, ou seja, o MG-FAS( $r = 3$ ) é cerca e 5 vezes mais rápido do que o MG-CS( $r = 3$ ).

Portanto, recomenda-se o uso do esquema FAS com a razão  $r = 3$  para as equações de advecção-difusão e Burgers. Esta constatação é inédita e inesperada, pois o esquema CS é indicado para resolver equações lineares e o esquema FAS, para equações não-lineares. Aqui vemos que o esquema CS perde a hegemonia perante o esquema FAS, mesmo que para um problema linear.

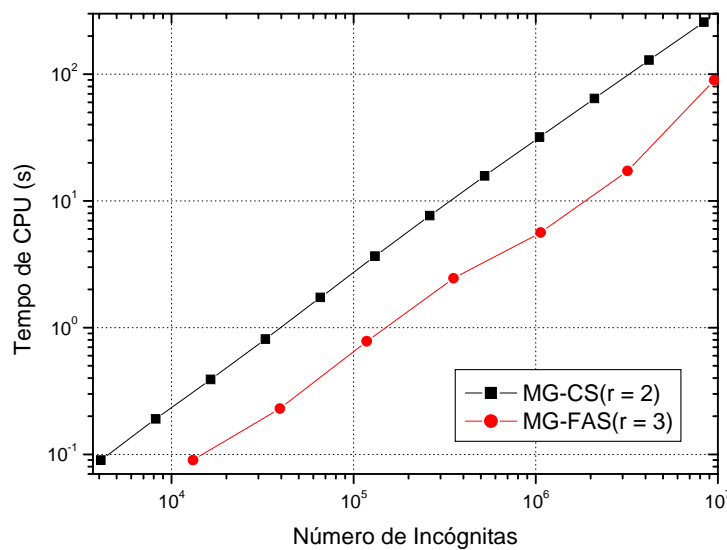


Figura 9 - Esquemas CS e FAS para a equação de advecção-difusão.

A conclusão do presente trabalho difere daquela de Brandt (1977). Ele fez análises teóricas e experimentais (numéricas) entre as razões de engrossamento  $r = 2, 3$  e  $3/2$  para diversos problemas, mas não se referiu às Eqs. (9) e (14) exatamente. Brandt mostra preferência pelo esquema CS em relação ao esquema FAS no caso de problemas lineares. Segundo Brandt, cada ciclo iterativo do esquema FAS é mais caro computacionalmente se comparado ao esquema CS devido à transferência de informações a respeito do resíduo e da solução para as malhas mais grossas.

Seja o problema de minimização do tempo de CPU em função do uso dos diferentes esquemas CS e FAS. Para este problema, verificou-se que o esquema ótimo tem uma forte influência da qualidade das informações transferidas pela restrição e pela prolongação e, principalmente, pela transferência de informações a respeito do resíduo e da solução para as

malhas mais grossas. Entretanto, no presente trabalho verificou-se que o número de ciclos ou iterações externas do esquema FAS é significativamente menor do que o esquema CS. Por exemplo, para a equação de advecção-difusão e para a malha de 8388608 elementos, o número de ciclos ou iterações externas (*ITE*) do MG-FAS( $r = 2$ ) e MG-CS( $r = 2$ ) é de 3 e 25, respectivamente.

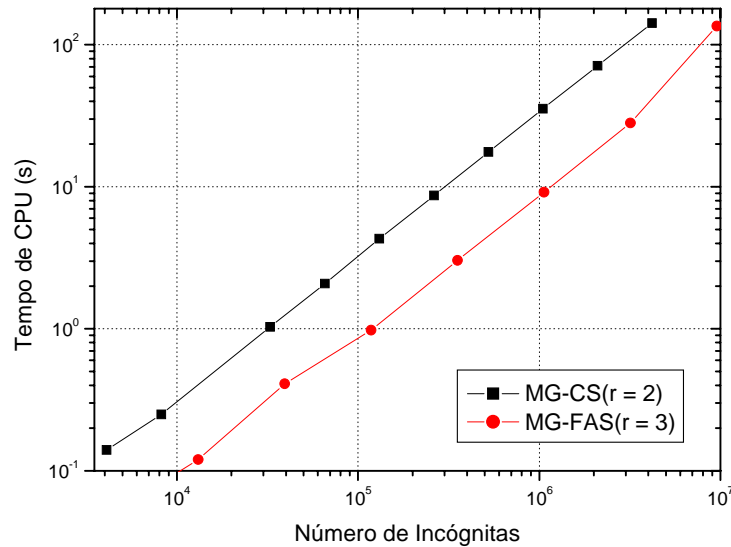


Figura 10 - Esquemas CS e FAS para a equação de Burgers.

## 5. CONCLUSÃO

Para os esquemas CS e FAS do método *multigrid* geométrico, neste trabalho estudou-se o efeito de diversos parâmetros sobre o tempo de CPU necessário para resolver dois problemas: um linear (equação de advecção-difusão) e outro não-linear (equação de Burgers), ambos em regime permanente, unidimensionais, com condições de contorno de Dirichlet. Estas equações foram discretizadas com o método de diferenças finitas.

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

- 1) Para os mesmos dados ( $\varepsilon$ ,  $\vec{v}$  e  $N_f$ ), o esquema FAS é mais rápido do que o esquema CS, para os problemas linear e não-linear.
- 2) O número de iterações internas (*ITI*) e o número de níveis de malhas ( $L$ ) afetam significativamente o tempo de CPU para os esquemas CS e FAS. Para o esquema FAS, recomenda-se utilizar  $ITI = 3r$  para qualquer número de incógnitas ( $N_f$ ); e para o esquema CS,  $ITI = 2r$ . Para ambos os esquemas, recomenda-se usar  $L = L_{máximo}$ .
- 3) Para os mesmos dados ( $\varepsilon$ ,  $\vec{v}$  e  $N_f$ ), com  $L = L_{máximo}$  para cada  $N_f$ , entre as razões de engrossamento de malhas testadas ( $r = 2, 3, 4$  e  $5$ ), o esquema CS é mais rápido com  $r = 2$ ; já o esquema FAS é mais rápido com  $r = 3$ .



## *Agradecimentos*

Agradecemos o apoio do Laboratório de Experimentação Numérica (LENA), do Departamento de Engenharia Mecânica da UFPR, por disponibilizar sua infra-estrutura. Os dois primeiros autores agradecem à Universidade Estadual de Ponta Grossa pelo suporte financeiro.

## **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.
- 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*”, 2ª ed., Johns Hopkins Press, Baltimore, 1989.
- Hestenes, M. and Stiefel, E., “Methods of Conjugate Gradients for Solving Linear Systems”, *Journal of Research of the National Bureau of Standards*, vol. 49, pp. 409-436, 1952.
- Hirsch, C., “*Numerical Computational of Internal and External Flows*”, Vol.1, Wiley, 1998.
- Maliska, C. R., “*Transferência de calor e mecânica dos fluidos computacional*”, LTC, 2ª 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.

- 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 2005a.
- Pinto, M. V., Santiago, C. D., Marchi, C. H. “Efeito de Parâmetros do Método *Multigrid* sobre o tempo de CPU para a Equação de Burgers Unidimensional”, Proceedings of the XXVI Iberian Latin-American Congress on Computational methods in Engineering CILAMCE 2005b.
- 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.