



CILAMCE2014

XXXV IBERIAN LATIN AMERICAN CONGRESS
ON COMPUTATIONAL METHODS IN ENGINEERING
FORTALEZA - CEARÁ - BRAZIL | NOVEMBER 23-26, 2014

OTIMIZAÇÃO DE PARÂMETROS DO MULTIGRID ALGÉBRICO PARA A EQUAÇÃO DE ADVECCÃO-DIFUSÃO BIDIMENSIONAL

Roberta Suero

roberta.suero@ifpr.edu.br

Instituto Federal de Educação, Ciência e Tecnologia do Paraná – Campus Campo Largo.

Rua Engenheiro Tourinho, 829, 83.607-140, Campo Largo, Paraná, Brasil.

Marcio Augusto Villela Pinto

Carlos Henrique Marchi

marcio_villela@ufpr.br

marchi@ufpr.br

Universidade Federal do Paraná.

Departamento de Engenharia Mecânica (DEMEC), 81.531-980, Curitiba, Paraná, Brasil.

Resumo. *Este trabalho apresenta comparações de parâmetros entre os métodos multigrid algébrico (AMG) e multigrid geométrico (GMG) para o problema de advecção-difusão bidimensional, considerando velocidades constantes e variáveis. O efeito causado pela anisotropia física sobre o tempo de CPU também é um dos objetivos deste estudo. O método de diferenças finitas com malhas uniformes e esquemas numéricos de primeira e segunda ordens de acurácia, respectivamente para os termos de advecção e difusão, são empregados. Os resultados são obtidos a partir de uma adaptação do código computacional AMG1R6 de Ruge e Stüben. Para o AMG são usadas as seguintes componentes: restrição por engrossamento padrão, prolongação padrão, esquema de correção (CS), solver Gauss-Seidel lexicográfico e ciclo V. São feitos estudos comparativos entre os tempos de CPU do AMG e singlegrid (método de malha única), e estudos comparativos entre o AMG e GMG, no que diz respeito ao número de iterações internas no solver e ao número de malhas. Para o AMG, são estudados os efeitos do fator de redução de malha (θ) e do fator de forte dependência na malha grossa (ϵ) sobre o tempo de CPU necessário para obter a solução numérica. Os resultados encontrados para o AMG são comparados aos conhecidos para a equação de Laplace (difusão pura) e os resultados obtidos para a anisotropia física também são confrontados aos alcançados pelo GMG.*

Palavras-chave: *Multigrid algébrico, equação de advecção-difusão, otimização de parâmetros.*

1 INTRODUÇÃO

Para resolver os sistemas lineares que surgem da discretização das equações diferenciais tem-se o método multigrid (Briggs et al., 2000 e Trottenberg et al., 2001). Este método proporciona taxa de convergência independente do tamanho da malha e é muito eficaz para resolver problemas que exigem grande esforço computacional (Gupta et al., 1997). De acordo com Stüben (2001), a eficiência do método multigrid não tem sido totalmente alcançada em aplicações realísticas de CFD (*Computational Fluid Dynamics*).

O método multigrid possui duas abordagens, relacionadas à forma de entrada dos dados e à construção das malhas auxiliares: método multigrid geométrico (GMG) (Trottenberg et al., 2001 e Briggs et al., 2000), em que são necessárias informações a respeito das malhas auxiliares; e método multigrid algébrico (AMG) (Ruge e Stüben, 1986; Brandt, 1986; Falgout, 2006 e Haase e Langer, 2002), em que é preciso conhecer apenas a matriz de coeficientes, proveniente da discretização do problema.

De acordo com Trottenberg et al. (2001), uma simples modificação no algoritmo pode resultar em uma redução significativa do tempo de CPU. Conforme Langer e Pusch (2006), para que o método multigrid seja realmente eficiente, é necessário adaptar suas componentes de acordo com o delineamento físico do problema e a formulação variacional.

Chang et al. (1996), ressaltam que o GMG aplica-se a problemas contínuos, em que a estrutura geométrica do problema é conhecida, com um algoritmo preparado para cada problema e eficiência muito boa. Já o AMG aplica-se a problemas em que os sistemas lineares de equações algébricas são conhecidos, utilizando as entradas da matriz, com um algoritmo único para todos os problemas e com boa eficiência.

Vários trabalhos trazem comparações entre o AMG e o GMG. Watanabe et al. (2005) observam o crescimento do tempo de CPU conforme o número de incógnitas aumenta, tanto para o AMG quanto para o GMG. Langer e Pusch (2006) comparam o número de ciclos gastos pelo AMG e GMG; mostram também os tempos para a geração das malhas auxiliares. O trabalho de Wu e Elman (2006) compara o número de iterações (ciclos) que o AMG e o GMG levam para atingir a tolerância estipulada em 10^{-6} .

O trabalho de Campos et al. (2006) compara a performance do AMG com GMG, pré-condicionados e com algoritmos paralelos para um sistema não linear de equações diferenciais. Para o GMG, há a variação do número de níveis. Para o AMG, são variados o número de níveis e o fator de correção de malha, quando são utilizados números diferentes de processadores.

Suero et al. (2012) apresentam uma comparação dos parâmetros do AMG com GMG para as equações de Laplace e Poisson, discretizadas em malhas quadrangulares. Também indicam os parâmetros ótimos para o AMG. Em Suero et al. (2008), a equação de advecção-difusão é resolvida com o GMG, onde são apresentados o número ótimo de iterações internas, número ótimo de malhas e efeito do número de incógnitas no tempo de CPU.

O objetivo do presente trabalho é minimizar o tempo de CPU para a resolução da equação de advecção-difusão bidimensional, em malhas estruturadas quadrangulares, com o uso do AMG. Tal objetivo pretende ser alcançado através do estudo das componentes do algoritmo. Este trabalho apresenta comparações do número ótimo de iterações internas e do número ótimo de níveis de malha para o AMG e GMG (Suero et al., 2008). São investigados ainda dois parâmetros que influenciam a geração das malhas auxiliares do AMG: fator de redução de malha e fator de forte dependência na malha grossa. Estes parâmetros são comparados aos obtidos para as equações de Laplace e Poisson, resolvidas em Suero et al. (2012).

Este artigo está dividido da seguinte forma: na seção 2 é apresentada a teoria do método multigrid algébrico; na seção 3, os modelos matemático e numérico; na seção 4 são apresentados os experimentos numéricos e seus resultados; e na seção 5, a conclusão do trabalho.

2 MÉTODO MULTIGRID ALGÉBRICO

O método multigrid algébrico é dividido em duas etapas. A fase inicial, chamada fase *setup*, é responsável pela geração das malhas e pela construção dos operadores de transferência entre as malhas (restrição e prolongação). A fase de solução emprega os operadores definidos na fase *setup* para resolver o problema. Nesta fase, as malhas auxiliares mais grossas (com menos pontos que a malha original) são visitadas conforme um ciclo previamente determinado. Para a construção das malhas auxiliares, é necessário fazer uma partição dos pontos da malha original, $i \in \Omega^h$, onde Ω^h denota o conjunto de índices $\{1,2,\dots,n\}$. Esta partição resulta em dois subconjuntos disjuntos: o conjunto C, composto pelos pontos que estão na malha grossa e seu complementar, o conjunto F, que contempla os pontos que não estão na malha grossa. Esta partição é feita com base nas fortes conexões algébricas, sendo que, para determiná-la, é necessário definir três conjuntos, conforme Eqs. (1), (2) e (3):

$$N_i = \{j \in \Omega^h: j \neq i, a_{ij} \neq 0\} \quad (1)$$

$$S_i = \left\{ j \in N_i: -a_{ij} \geq \theta \max_{a_{ik} < 0} |a_{ik}| \quad \text{com } \theta \text{ fixo } 0 < \theta < 1 \right\} \quad (2)$$

$$S_i^T = \{j \in \Omega: i \in S_j\} \quad (3)$$

Nas equações acima, N_i é a vizinhança do ponto i , S_i determina o conjunto de pontos que influenciam fortemente i e S_i^T representa o conjunto de pontos que dependem fortemente de i . O parâmetro θ ($0 < \theta < 1$), chamado fator de redução de malha, quantifica o quanto um ponto está fortemente conectado aos outros. De acordo com Ruge e Stüben (1986), Trottenberg et al. (2001), Cleary et al. (2000), Krechel e Stüben (1999) e Chang et al. (1996), o valor a ser empregado é $\theta=0,25$. Briggs et al. (2000) utilizam $\theta=0,20$ e Falgout (2006), $\theta=0,40$. Iwamura et al. (2003) empregam vários valores de θ , que variam conforme a malha que está sendo resolvida. No presente trabalho é apresentado um estudo da influência deste parâmetro sobre o tempo de CPU.

Após a realização da partição, para cada ponto i , tem-se três outros subconjuntos, necessários para se fazer o engrossamento da malha. O conjunto $C_i = C \cap S_i$, que contempla os pontos da vizinhança da malha grossa que influenciam fortemente i . O conjunto $D_i^S = D_i \cap S_i$, onde $D_i = N_i - C_i$, que são os pontos da vizinhança de F que influenciam fortemente i . O conjunto $D_i^W = D_i - S_i$, são os chamados pontos fracamente conectados, que podem estar tanto em C quanto em F. O operador de interpolação (que transfere as informações entre as malhas grossa e fina) é dado pelas Eqs. (4) e (5) (Briggs et al., 2000 e Iwamura et al., 2003):

$$(I_H^h e^H)_i = \begin{cases} e_i^H & \text{se } i \in C \\ \sum_{j \in C_i} w_{ij} e_j^H & \text{se } i \in F \end{cases} \quad (4)$$

onde,

$$w_{ij} = - \frac{a_{ij} + \sum_{m \in D_i^S} \left(\frac{a_{im} a_{mj}}{\sum_{k \in C_i} a_{mk}} \right)}{a_{ii} + \sum_{n \in D_i^W} a_{in}} \quad (5)$$

Outra constante que é objeto de estudo neste trabalho é o fator de forte dependência na malha grossa (ε). Para o ponto $j \in D_i^s$ é necessário decidir quando um ponto está influenciando fortemente um ponto em C , de tal forma que isto interfira na interpolação. Para tanto, é necessário definir um conjunto de dados, conforme Eq. (6) (Iwamura et al., 2003):

$$S_i^D = \left\{ j \in D_i^s : \sum_{l \in C_i} |a_{jl}| > \varepsilon \left(\frac{|a_{ij}|}{\max |a_{ik}|} \right) \max |a_{jl}| \quad \text{com } \varepsilon > 0 \text{ fixo} \right\} \quad (6)$$

O parâmetro $\varepsilon > 0$ tem seu valor fixado em $\varepsilon=0,35$, conforme Ruge e Stüben (1986), Briggs et al. (2000), Cleary et al. (2000), Krechel e Stüben (1999), Chang et al. (1996) e Falgout (2006). Em Trottenberg et al. (2001) é utilizado $\varepsilon=0,20$ e em Iwamura et al. (2003), $\varepsilon=0,35$ e $\varepsilon=0,45$, de acordo com o nível de malha que está sendo resolvido. Neste trabalho é apresentado um estudo da influência deste parâmetro sobre o tempo de CPU.

O código computacional empregado para a obtenção dos resultados aqui apresentados teve como base o programa AMG1R6, de Ruge e Stüben (1986). Deste código é utilizada a fase *setup*, que emprega engrossamento padrão (baseado nas fortes conexões negativas) e prolongação padrão (implementada devido ao uso do engrossamento padrão). Para a fase de solução é empregado ciclo V, estimativa inicial nula, *solver* Gauss-Seidel lexicográfico, critério de parada baseado na norma \bar{l}_2 do resíduo, adimensionalizada pelo resíduo da estimativa inicial, e tolerância de 10^{-9} . No código original de Ruge e Stüben foram feitas diversas modificações na fase de solução, para atender aos objetivos do presente trabalho. A forma de entrada dos dados também foi modificada, tendo agora um gerador de matrizes acoplado ao programa principal.

3 MODELOS MATEMÁTICO E NUMÉRICO

Para malhas estruturadas quadrangulares, a equação de advecção-difusão foi discretizada com o uso do método de diferenças finitas (Tannehill et al., 1997), em um domínio de cálculo quadrado unitário. O problema de advecção-difusão 2D, com propriedades constantes, em regime permanente, com condições de contorno de Dirichlet, pode ser representado matematicamente da seguinte forma (Tannehill et al., 1997 e Versteeg e Malalasekera, 2007):

$$u \frac{\partial T}{\partial x} + v \frac{\partial T}{\partial y} = \alpha \frac{\partial^2 T}{\partial x^2} + \alpha \frac{\partial^2 T}{\partial y^2} + S, \quad 0 < x, y < 1 \quad (7)$$

Na Eq. (7), T representa a temperatura, α , a difusividade térmica, u e v as velocidades nas direções x e y , respectivamente. S é o termo fonte dado conforme as velocidades adotadas.

As velocidades, termo fonte, solução analítica e condições de contorno de cada problema estão dispostas na Tabela 1. Pela observação da mesma, pode-se concluir que os dois casos abordados são diferenciados pelo campo de velocidades, que irá levar à resolução de dois problemas distintos. Para o campo de velocidades considerado constante, o vetor velocidades forma um ângulo de 30° . Para velocidades variáveis, o campo de velocidades é obtido de Shih et al. (1989).

Com a aplicação da aproximação UDS (*Upstream Differencing Scheme*) nos termos advectivos e CDS (*Central Differencing Scheme*) nos termos difusivos (Versteeg e Malalasekera, 2007) chega-se a um sistema de equações algébricas. Neste sistema, do tipo $AT = b$, A corresponde à matriz de coeficientes, T representa o vetor de incógnitas e b é o termo fonte.

Tabela 1. Problemas abordados a partir da equação de advecção-difusão

Campo de velocidades	Termo fonte	Solução Analítica	Condições de Contorno
<p>Constante</p> <p>$u = \sqrt{3}/2,$ $v = 1/2$</p>	$S = u^2vy + uv^2x$	$T(x, y) = uvxy$	<p>$T(x, 0) = T(0, y) = 0$</p> <p>$T(1, y) = uvx$</p> <p>$T(x, 1) = uvx$</p>
<p>Variável</p> <p>$u(x, y) = 8(x^4 - 2x^3 + x^2)(4y^3 - 2y),$ $v(x, y) = -8(4x^3 - 6x^2 + 2x)(y^4 - y^2)$</p>	$S = u^2vy + uv^2x$	$T(x, y) = uvxy$	<p>$T(x, 0) = T(0, y) = 0$</p> <p>$T(1, y) = uvx$</p> <p>$T(x, 1) = uvx$</p>

4 RESULTADOS

Cerca de 300 simulações foram realizadas, com as seguintes variações: número de iterações internas no *solver* (ν), número de níveis de malha (L), fator de redução de malha (θ), fator de forte dependência na malha grossa (ϵ) e número de incógnitas (N). O efeito destes parâmetros para a equação de advecção-difusão, discretizada em malhas estruturadas quadrangulares, é o objetivo deste estudo. Para a obtenção dos quatro primeiros parâmetros são considerados: $N = 257 \times 257$, 1025×1025 e 4097×4097 incógnitas. Para avaliar o efeito do número de incógnitas (N) no tempo de CPU são considerados os problemas $N = 5 \times 5$ até 4097×4097 incógnitas. O objetivo dos testes apresentados abaixo é encontrar o valor ótimo para cada um dos parâmetros testados. Entende-se por valor ótimo como sendo aquele que resulta no menor tempo de CPU quando aplicado ao AMG.

4.1 Número de iterações internas (ν)

Para encontrar o valor ótimo para este parâmetro foram feitos testes para os três tamanhos de problema já descritos, sendo o valor de ν variado entre 1 e 20. Nos casos em que as simulações têm tempo de CPU inferior a 10 segundos, estas foram repetidas 3 vezes. Nestes casos foram tomadas as médias aritméticas dos tempos.

Na Fig. 1, pode-se observar os resultados obtidos para o número de iterações internas para malhas quadrangulares com $N = 4097 \times 4097 = 16785409$ incógnitas, considerando os dois casos estudados (velocidades constantes e variáveis). Os resultados obtidos para outros valores de N são qualitativamente iguais. O número ótimo de iterações internas, para velocidades constantes é 2, e para velocidades variáveis é 1 e 2. Neste caso, considera-se $\nu_{\text{ótimo}} = 2$, para os dois casos analisados. Este é o valor a ser utilizado no restante das simulações. Observa-se também que quanto maior o número de iterações internas, maior é o tempo de CPU gasto para se resolver o problema.

Em Ruge e Stüben (1986), Krechel e Stüben (1999), Briggs et al. (2000), Trottenberg et al. (2001), Iwamura et al. (2003) e Falgout (2006) é empregado $\nu = 1$ para a resolução dos problemas, para qualquer tipo de malha, com o uso do AMG. Suero et al. (2008) resolvem o mesmo problema aqui tratado, utilizando GMG e recomendam o uso de $\nu = 3$ para velocidades constantes e variáveis.

Um estudo a respeito do efeito do número de iterações internas no tempo de CPU para o AMG pode ser encontrado em Suero et al. (2012), onde são abordadas as equações de Laplace e Poisson. Neste trabalho é recomendado empregar $\nu = 2$ para as equações resolvidas.

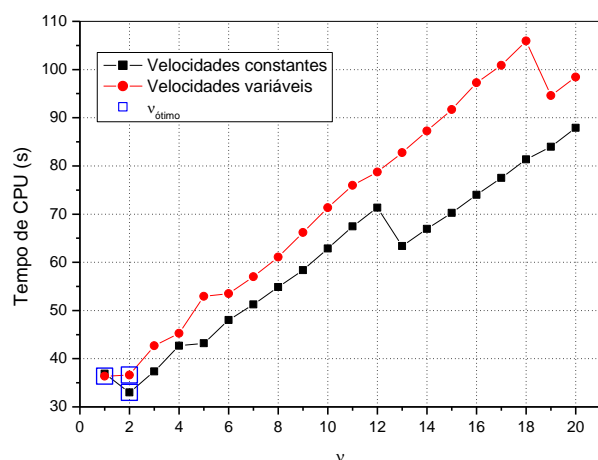


Figura 1. Número de iterações internas versus Tempo de CPU

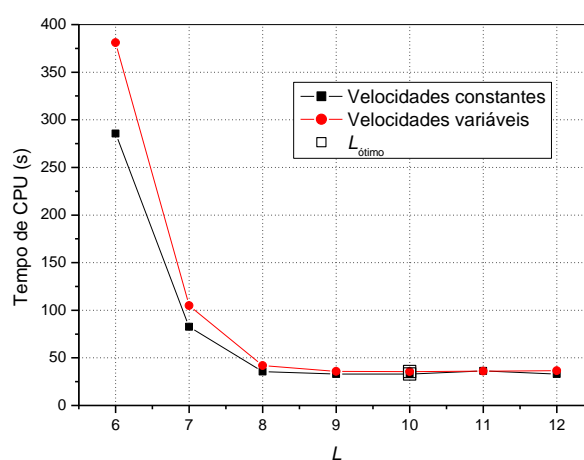


Figura 2. Número de níveis versus Tempo de CPU

4.2 Número de níveis (L)

Para o estudo da influência do número de níveis no tempo de CPU foi utilizado o valor ótimo para o número de iterações internas obtido anteriormente, $\nu_{\text{ótimo}} = 2$. Os valores empregados para o número de níveis foram variados de seu valor máximo (utilizando todos os níveis possíveis na resolução) até um determinado número de níveis, que varia conforme o tamanho do problema. Para a determinação do valor mínimo para este parâmetro, deve-se observar o crescimento do tempo de CPU, conforme o número de níveis está diminuindo. Este valor foi determinado depois da realização de vários testes computacionais. O valor tomado como mínimo é aquele em que o tempo de CPU torna-se muito grande. O mínimo para este trabalho é aquele em que a resolução do problema leva em torno de uma hora. Deve-se observar que, com este valor adotado, já é possível notar o crescimento do tempo de CPU, comportamento para L que se quer estudar.

Na Fig. 2 observa-se o comportamento do tempo de CPU com a variação do número de níveis. Desta análise, conclui-se que o número de níveis a ser empregado na resolução do AMG influencia no tempo de CPU, para os dois problemas abordados. Nota-se que, quanto menor o número de níveis de malha empregado, maior é o tempo de CPU para o AMG. O número de níveis que apresenta o menor tempo de CPU é $L_{\text{ótimo}} = L_{\text{máximo}} - 2$, que é o valor a ser recomendado para estes problemas.

Estes resultados concordam parcialmente com os obtidos para o AMG em Suero et al. (2012), onde o número ótimo de níveis é o valor máximo a ser empregado. Nos trabalhos de Ruge e Stüben (1986), Falgout (2006) e Langer e Pusch (2006) não é observada a preocupação com o número de malhas a ser empregado. Nos trabalhos em que este dado está explícito, observa-se a utilização de todos os níveis de malha, como por exemplo em Krechel e Stüben (1999), Wu e Elman (2006) e Iwamura et al. (2003). Suero et al. (2008) recomendam o uso de todos os níveis de malha para a resolução da equação de advecção-difusão com GMG.

Neste caso, pode-se concluir que o tipo de problema a ser resolvido não exerce influência no valor ótimo encontrado para o número de malhas. Os resultados aqui apresentados são para

as malhas mais refinadas, sendo que, para outros valores de N , os resultados obtidos são qualitativamente iguais.

4.3 Fator de redução de malha (θ)

Para o estudo deste parâmetro são considerados os valores ótimos obtidos para o número de iterações internas e número de malhas, respectivamente $\nu = 2$ e $L = L_{m\acute{a}ximo} - 2$. O fator de redução de malha é uma constante que quantifica o quanto um ponto está fortemente conectado a outro. Dependendo desta relação, um ponto pode ser considerado da malha grossa ou não. Existe uma discordância com relação aos valores empregados pelos autores que resolvem problemas empregando AMG. A maioria dos trabalhos, como Ruge e Stüben (1986), Krechel e Stüben (1999), Trottenberg et al. (2001) e Iwamura et al. (2003) emprega $\theta=0,25$. Briggs et al. (2000) empregam $\theta=0,20$ e Falgout (2006) usa $\theta=0,40$. Nenhum dos trabalhos pesquisados justifica a escolha do valor para θ , apenas ressaltam que $\theta=0,25$ é um valor padrão. Suero et al. (2012) trazem um estudo deste parâmetro para o AMG. Neste trabalho, observa-se que o valor ótimo encontrado varia com o tipo de problema (equações de Laplace, $\theta=0,25$ e Poisson, $\theta=0,20$) e com o tipo de malha empregada (quadrangular descrita conforme as equações e triangular, $\theta=0,0625$). Por analogia ao GMG, observou-se que $\theta=0,25$ corresponde à interpolação de quatro pontos para a obtenção do ponto da malha imediatamente mais fina, que corresponde à razão de engrossamento $r=2$ para o GMG. Os outros valores aqui testados para este parâmetro têm correspondência com o GMG: $\theta=0,11$ corresponde a $r=3$ e $\theta=0,0625$ corresponde a $r=4$. Estes valores são os mesmos explorados em Suero et al. (2012).

Na Fig. 3 observa-se o comportamento de θ para os problemas aqui resolvidos. Os resultados ilustrados nesta figura são os obtidos para as malhas mais refinadas, sendo que para os outros valores de N , os resultados são qualitativamente iguais. Conforme pode ser observado, o fator de redução de malha apresenta o mesmo valor ótimo para os dois problemas abordados, sendo o mesmo considerado padrão na literatura ($\theta=0,25$).

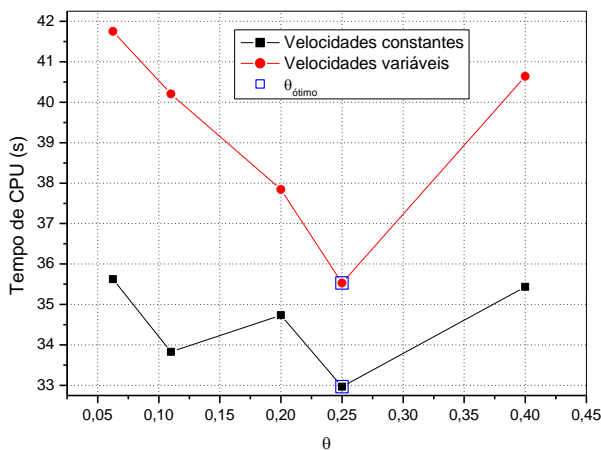


Figura 3. Efeito do fator de redução de malha no tempo de CPU

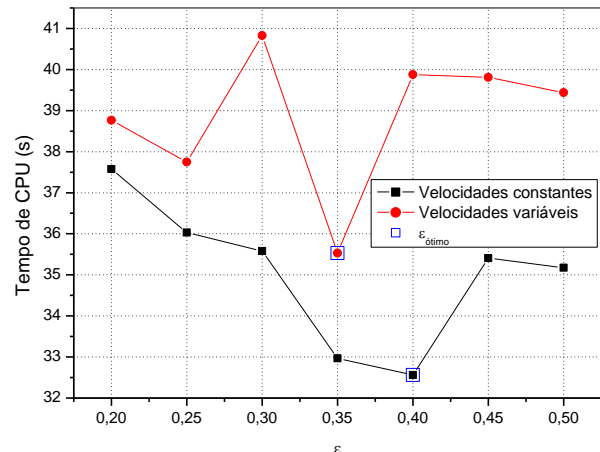


Figura 4. Efeito do fator de forte dependência na malha grossa no tempo de CPU

4.4 Fator de forte dependência na malha grossa (ϵ)

Para se encontrar o valor ótimo para ϵ , são utilizados os parâmetros ótimos encontrados acima, para cada um dos problemas abordados. O fator de forte dependência na malha grossa é uma constante que define a forte dependência no conjunto de pontos que estão na malha grossa. Esta constante influencia na construção das malhas auxiliares na aplicação do AMG. Este valor

não é divulgado na grande maioria dos trabalhos que tratam do AMG, como em Krechel e Stüben (1999), Watanabe et al. (2005), Campos et al. (2006), Falgout (2006) e em Langer e Pusch (2006). Em Ruge e Stüben (1986) é utilizado $\varepsilon=0,35$. Em Iwamura et al. (2003) são empregados valores diferentes conforme o nível de malha que está sendo construído. Para este trabalho tem-se $\varepsilon=0,35$ para o número de níveis menor ou igual a 3 e, $\varepsilon=0,45$ quando o número de níveis for maior que 3. Em Trottenberg et al. (2001) é utilizado $\varepsilon=0,20$ na resolução da equação de Poisson. No presente trabalho são testados diversos valores para ε . Estes valores são os mesmos explorados em Suero et al. (2012), que trazem um estudo a respeito da variação deste parâmetro na resolução das equações de Laplace ($\varepsilon_{\text{ótimo}} = 0,35$) e Poisson ($\varepsilon_{\text{ótimo}} = 0,40$).

Na Fig. 4 observa-se o comportamento deste parâmetro para os dois casos aqui abordados. Os resultados ilustrados são os obtidos para os problemas com mais incógnitas, sendo que os resultados para outros valores de N são qualitativamente iguais. Para a equação de advecção-difusão, com velocidades variáveis, o valor ótimo para o fator de forte dependência na malha grossa é o mesmo, ou seja, $\varepsilon=0,35$. Este resultado concorda com a literatura. Para velocidades constantes este valor é diferente, $\varepsilon=0,40$. Conclui-se assim, que este valor é sensível quanto ao problema que está sendo resolvido.

4.5 Efeito do número de incógnitas (N)

Para avaliar o efeito do número de incógnitas no tempo de CPU na resolução dos problemas em questão, são tomados os parâmetros ótimos destacados anteriormente. Para velocidades constantes $\nu = 2, L = L_{\text{máximo}} - 2, \theta = 0,25$ e $\varepsilon = 0,40$. Para velocidades variáveis, $\nu = 2, L = L_{\text{máximo}} - 2, \theta = 0,25$ e $\varepsilon = 0,35$. Para os dois problemas abordados com o AMG (velocidades constantes e variáveis), foram feitos estudos a respeito da inclinação das curvas apresentadas na Fig. 5. Tal estudo é feito com o ajuste de curvas pelo método dos mínimos quadrados, utilizando uma curva geométrica, conforme a equação abaixo:

$$t = cN^p \quad (8)$$

Na Eq. (8), p representa a ordem do *solver* associado ao método empregado, e c é um coeficiente associado ao método e ao *solver*. Para cada método, quanto mais próximo da unidade estiver o expoente p , melhor é seu desempenho, pois o método multigrid ideal é aquele cujo $p = 1$, ou seja, o tempo de CPU cresce linearmente conforme o tamanho do problema. Foram utilizados os três pontos de cada curva com maior N para se fazer o ajuste, pois, para os outros tamanhos de problema, tem-se como maior influência o tempo para a geração das malhas no tempo de CPU. O efeito do método multigrid (princípio que se quer estudar aqui) é melhor observado nas malhas mais refinadas. Na Tabela 2 são observados os valores para o expoente p , que manteve-se razoavelmente próximo da unidade, conforme o esperado. Para o GMG, os dados são obtidos de Suero et al. (2008). Para o método singlegrid (SG - método de malha única), este valor manteve-se próximo de dois.

Na Fig. 5, observa-se que o AMG é mais rápido (leva menos tempo de CPU) que o GMG. Isso ocorre porque os tempos de CPU para o GMG (Suero et al., 2008) foram obtidos em um computador com capacidade de processamento inferior ao utilizado para a obtenção dos resultados aqui apresentados (computador dos testes atuais possui processador aproximadamente 35% mais eficiente). Tal fato não deve ocorrer, pois o AMG, em malhas estruturadas, é mais custoso computacionalmente que o GMG. O que se quer enfatizar com esta figura é a inclinação para as curvas do AMG e GMG. Observa-se que as inclinações são praticamente as mesmas, para os dois problemas analisados. Este fato pode ser comprovado pela Tabela 2, que mostra os valores de p próximos para estes casos. Nota-se também que estas

curvas, para o AMG e GMG, estão um pouco afastadas, devido ao coeficiente c . Na Fig. 5 observa-se ainda que, a curva para o singlegrid, além de estar mais afastada das demais, possui inclinação diferente (fato que também pode ser confirmado pela Tabela 2). Esta inclinação, para os problemas menores, é praticamente nula (paralela ao eixo das abcissas), enquanto que para problemas maiores ela cresce, se distanciando das demais. Isso ocorre pois os resultados apresentados na Fig. 5, para o SG, foram obtidos com o mesmo código do AMG, portanto, tendo um tempo adicional para a geração da matriz de coeficientes. O comportamento dos tempos de CPU para o AMG e GMG concordam com os apresentados por Watanabe et al. (2005) e Suero et al. (2012), que trazem gráficos onde são comparados os tempos de CPU para o AMG e GMG.

Tabela 2. Valor de p em $t=cN^p$.

Problema resolvido	AMG	GMG	SG
Velocidades constantes	0,92	1,08	1,77
Velocidades variáveis	0,94	1,10	1,82

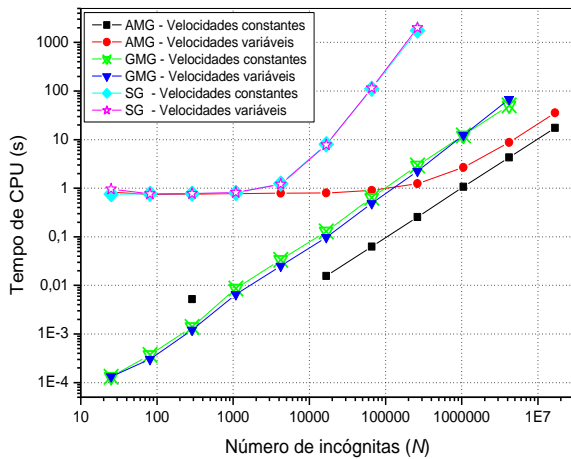


Figura 5. Efeito do número de incógnitas no tempo de CPU

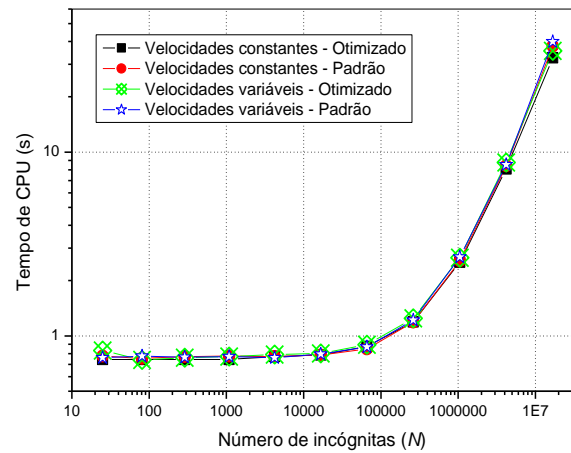


Figura 6. Algoritmo padrão versus otimizado

4.6 Otimização do algoritmo AMG

Neste trabalho, o principal objetivo é a otimização do código computacional, através da análise de parâmetros previamente determinados. Estes parâmetros são testados um a um, de forma a resultar em um conjunto ótimo para cada um dos casos estudados. Com este conjunto de otimizações, são obtidos ganhos em tempo de CPU e em número de ciclos, para a resolução dos problemas abordados. A diferença entre o problema resolvido com o algoritmo padrão ($v=1$, $L = L_{máximo}$, $\theta=0,25$ e $\varepsilon=0,35$) e o algoritmo otimizado, aumenta com o número de incógnitas, como pode-se observar na Fig. 6. O algoritmo ótimo contempla os parâmetros ótimos obtidos para cada problema acima estudado. Na Tabela 3 pode-se observar os fatores de aceleração adimensionalizados (razões entre os tempos de CPU) para o programa otimizado e o padrão. Esta razão é conhecida como *speedup*. Pode-se observar também os tempos de CPU para os problemas resolvidos com $N = 4097 \times 4097$ incógnitas.

Tabela 3. Comparações do tempo de CPU entre o algoritmo padrão e o otimizado.

Problema resolvido	Tempo de CPU (s)	Tempo de CPU (s)	$\frac{t_{CPU}(\text{otimizado})}{t_{CPU}(\text{padrão})}$
	Padrão	Otimizado	
Velocidades constantes	37,67	32,56	0,86
Velocidades variáveis	39,88	35,53	0,89

Conforme a Tabela 3, pode-se concluir que são alcançados ganhos em tempo de CPU para os problemas aqui resolvidos. A redução obtida do algoritmo otimizado em relação ao padrão é de 15,7% para velocidades constantes, e de 12,2% para velocidades variáveis.

Na Tabela 4, podem ser observados os valores obtidos para o valor de p (em $t = cN^p$) tanto para o AMG otimizado, quanto para o AMG padrão. Todos os valores obtidos estão próximos da unidade. Pode-se observar ainda o valor de c , constante que determina o afastamento entre as curvas da Fig. 6. Nesta tabela, pode-se observar também o número de ciclos (iterações externas) realizados para o AMG atingir a tolerância estipulada. Nota-se uma redução neste número para o algoritmo otimizado.

Tabela 4. Valores de c e p em $t=cN^p$

Problema resolvido	AMG – Padrão (c)	AMG – Otimizado (c)	AMG – Padrão (p)	AMG – Otimizado (p)	Ciclos – Padrão	Ciclos – Otimizado
Velocidades constantes	$6,01 \cdot 10^{-6}$	$6,7 \cdot 10^{-6}$	0,93	0,92	11	7
Velocidades variáveis	$3,47 \cdot 10^{-6}$	$6,01 \cdot 10^{-6}$	0,97	0,94	12	8

5 CONCLUSÃO

Neste trabalho, a equação de advecção-difusão bidimensional foi resolvida com o emprego do método AMG. A equação foi discretizada com o método de diferenças finitas, em malhas estruturadas quadrangulares. As condições de contorno são do tipo Dirichlet. Para o vetor velocidades da equação, foram considerados dois campos diferentes: velocidades constantes e velocidades variáveis. Isso levou à resolução de dois problemas diferentes, originados da mesma equação diferencial parcial. O objetivo do trabalho era analisar os efeitos dos seguintes parâmetros sobre o tempo de CPU: número de iterações internas (ν), número de malhas (L), fator de redução de malha (θ), fator de forte dependência na malha grossa (ε) e número de incógnitas (N). Alguns dos resultados obtidos foram comparados aos disponíveis para o GMG e AMG. Com a realização deste trabalho, verificou-se que:

- 1) O número de iterações internas afeta significativamente o tempo de CPU. Tanto para velocidades constantes, como para velocidades variáveis. Recomenda-se empregar $\nu = 2$. Para a resolução do mesmo problema com o GMG, o valor ótimo encontrado foi $\nu = 3$.
- 2) O número de níveis empregados para a resolução do AMG afeta significativamente o tempo de CPU. Para os problemas acima citados, observou-se que $L_{\text{ótimo}} = L_{\text{máximo}} - 2$. Para o GMG o ótimo encontrado é o número máximo de malhas.
- 3) Não houve variação para o fator de redução de malha para os problemas abordados. Tanto para velocidades constantes, quanto para velocidades variáveis tem-se $\theta=0,25$, mesmo valor recomendado pela literatura.
- 4) Houve variações do fator de forte dependência na malha grossa para os problemas resolvidos. Para velocidades constantes, tem-se $\varepsilon=0,40$, para velocidades variáveis tem-se $\varepsilon=0,35$.
- 5) Com o estudo proposto dos parâmetros do AMG foi obtida uma redução de pelo menos 12% no tempo de CPU, na resolução dos problemas.
- 6) Houve redução no número de ciclos necessários para se atingir a convergência estipulada com as otimizações aqui propostas. Esta redução foi de, pelo menos, 36%.

AGRADECIMENTOS

A primeira autora agradece ao Laboratório de Experimentação Numérica (LENA), do Departamento de Engenharia Mecânica da UFPR, por disponibilizar sua estrutura e ao IFPR. O terceiro autor é bolsista do CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico). Os autores agradecem ao CNPq, à Fundação Araucária (Paraná) e à Agência Espacial Brasileira (AEB), através do Programa Uniespaço, pelo apoio financeiro. Os autores agradecem ao Dr. K. Stüben pela cessão do código AMG1R6 utilizado neste trabalho.

REFERÊNCIAS

- Brandt, A., 1986. Algebraic Multigrid Theory: The Symmetric Case. *Applied Mathematics and Computation*, vol. 19, pp. 23-56.
- Briggs, W. L., Henson, V. E. & McCormick, S. F., 2000. *A Multigrid Tutorial*, 2ª ed. SIAM.
- Campos, F. O., Oliveira, R. S. & Santos, R. W., 2006. Performance Comparison of Parallel Geometric and Algebraic Multigrid Preconditioners for Bidomain Equations. *Springer-Verlag Berlin Heidelberg*, pp. 76-83.
- Chang, Q., Wong, Y. S. & Fu, H., 1996. On the Algebraic Multigrid. *Journal of Computational Physics*, vol. 125, pp. 279-292.
- Cleary, A. J., Falgout, R. D., Henson, V. E., Jones, J. E., Manteuffel, T. A., McCormick, S. F., Miranda, G. N. & Ruge, J. W., 2000. Robustness and Scalability of Algebraic Multigrid. *SIAM Journal on Scientific Computing*, vol. 21, n. 5, pp. 1886-1908.
- Falgout, R. D., 2006. An Introduction to Algebraic Multigrid. *Computing in Science and Engineering*, pp. 1-11.
- Ferziger, J. H. & Peric, M., 2002. *Computational Methods for Fluid Dynamics*. Berlin: Springer-Verlag.
- Gupta, M. M. Kouatchou, J. & Zhang, J., 1997. A compact multigrid solver for convection-diffusion equations. *Journal of Computational Physics*, vol. 132, pp. 123-129.
- Haase, G. & Langer, U., 2002. Multigrid Methods: From Geometric to Algebraic Versions, em Bourlioux A. e Gander, M. J. *Modern Methods in Scientific Computing and Applications*. Kluwer Academic Press, Dordrecht, pp. 103-154.
- Iwamura, C., Costa, F. S., Sbarski, I., Easton, A. & Li, N., 2003. An efficient algebraic multigrid preconditioned conjugate gradient solver. *Computer Methods in Applied Mechanics and Engineering*, vol. 192, pp. 2299-2318.
- Krechel, A. & Stüben, K., 1999. Operator Dependent Interpolation in Algebraic Multigrid, *Proceedings of Fifth European Multigrid Conference*, 24 p.
- Langer, U. & Pusch, D., 2006. Comparison of Geometrical and Algebraic Multigrid Preconditioners for Data-Sparse Boundary Element Matrices. *Springer-Verlag Berlin Heidelberg*, pp. 130-137.
- Ruge, J. & Stüben, K., 1986. Algebraic Multigrid (AMG). *S. F. McCormick, Multigrid Methods, volume 5 de Frontiers in Applied Mathematics, SIAM*.
- Shih, T. M., Tan, C. H., Hwang, B. C., 1989. Effects of grid staggering on numerical scheme. *International Journal for Numerical Methods in Fluids*, v. 9, pp. 193-212.

- Suero, R., Pinto, M. A. V., Marchi, C. H., 2008. Efeito da anisotropia física sobre o método multigrid na solução da equação de advecção-difusão 2D. *XXIX CILAMCE*.
- Suero, R., Pinto, M. A. V., Marchi, C. H., Araki, L. K. & Alves, A. C., 2012. Analysis of algebraic multigrid parameters for two-dimensional steady-state heat diffusion equation. *Applied Mathematical Modeling*, vol. 36, pp. 2996-3006.
- Stüben, K., 2001. A review of algebraic multigrid. *Journal of Computational and Applied Mathematics*, vol. 128, pp. 281-309.
- Tannehill, J. C., Anderson, D. A. & Pletcher, R. H., 1997. *Computational Fluid Mechanics and Heat Transfer*. Taylor e Francis.
- Trottenberg, U., Oosterlee, C. & Schüller, A., 2001. *Multigrid*. Academic Press.
- Versteeg, H. G.; Malalasekera, W., 2007. *An introduction to Computational Fluid Dynamics: The Finite Volume Method*. Harlow, England: Pearson/Prentice Hall.
- Watanabe, K., Igarashi, H. & Honma, T., 2005. Comparison of Geometric and Algebraic Multigrid Methods in Edge-Based Finite-Element Analysis. *IEEE Transactions on Magnetics*, vol. 41, n. 5, pp. 1672-1675.
- Wu, C., T. & Elman, H. C., 2006. Analysis and Comparison of Geometric and Algebraic Multigrid for Convection-Diffusion Equations. *SIAM Journal on Scientific Computing*, vol. 28, n. 6, pp. 2208-2228.