

## **OTIMIZAÇÃO DO MÉTODO *MULTIGRID* ALGÉBRICO PARA AS EQUAÇÕES BIDIMENSIONAIS DE LAPLACE E POISSON**

Roberta Suero, [roberta.suero@ifpr.edu.br](mailto:roberta.suero@ifpr.edu.br)<sup>1,2</sup>  
Marcio Augusto Villela Pinto, [marcio\\_villela@yahoo.com.br](mailto:marcio_villela@yahoo.com.br)<sup>3,4</sup>  
Carlos Henrique Marchi, [marchi@ufpr.br](mailto:marchi@ufpr.br)<sup>4</sup>  
Luciano Kiyoshi Araki, [lucaraki@ufpr.br](mailto:lucaraki@ufpr.br)<sup>4</sup>  
Arlaide Cristina Alves, [aalves@up.edu.br](mailto:aalves@up.edu.br)<sup>5</sup>

- <sup>1</sup> Programa de Pós-Graduação em Métodos Numéricos em Engenharia, Universidade Federal do Paraná, Curitiba, PR  
<sup>2</sup> Instituto Federal de Educação, Ciência e Tecnologia do Paraná, Campus Paranaguá, PR  
<sup>3</sup> Departamento de Matemática e Estatística, Universidade Estadual de Ponta Grossa, Ponta Grossa, PR  
<sup>4</sup> Departamento de Engenharia Mecânica, Universidade Federal do Paraná, Curitiba, PR  
<sup>5</sup> Núcleo de Ciências Exatas e Tecnológicas, Universidade Positivo, Curitiba, PR

**Resumo.** Este trabalho apresenta comparações de parâmetros entre os métodos *multigrid* algébrico (AMG) e *multigrid* geométrico (GMG) para as equações bidimensionais de Laplace e Poisson, em malhas estruturadas quadrangulares e triangulares. Os parâmetros analisados são: número de iterações internas no solver, número de malhas e número de incógnitas. Para o AMG, também são estudados os efeitos do fator de redução de malha e do fator de forte dependência na malha grossa sobre o tempo de CPU necessário para obter a solução numérica. Para malhas quadrangulares é empregado o método de diferenças finitas, e para malhas triangulares, o de volumes finitos. Os resultados são obtidos com uma adaptação do código computacional AMGIR6 de Ruge e Stüben. Para o AMG são usadas as seguintes componentes *multigrid*: 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 método *multigrid* geométrico, *multigrid* algébrico e *singlegrid* (método de malha única). Verificou-se que: 1) o número ótimo de iterações internas obtidas para o AMG e GMG, em malhas quadrangulares, é o mesmo, porém diferente para malhas triangulares; 2) o número ótimo de malhas é o número máximo, tanto para malhas quadrangulares quanto para malhas triangulares; 3) o AMG mostrou-se sensível à variação do fator de redução de malha e do fator de forte dependência na malha grossa, tanto com relação às equações abordadas, quanto aos tipos de malha; e 4) para malhas quadrangulares, o GMG resolve o problema em 20% do tempo gasto pelo AMG.

**Palavras-chave:** diferenças finitas, volumes finitos, otimização.

### **1. INTRODUÇÃO**

O método *multigrid* tem sido amplamente aplicado devido à sua eficiência na resolução de problemas de engenharia. Esta eficiência está ligada à rápida convergência obtida na resolução de problemas com muitas incógnitas. O método *multigrid* tem duas abordagens, que estão relacionadas à forma de entrada dos dados e na construção das malhas auxiliares: o método *multigrid* geométrico (GMG) (Trottenberg et al, 2001 e Briggs et al, 2000), em que são necessárias informações a respeito das malhas auxiliares; e o método *multigrid* algébrico (AMG) (Ruge e Stüben, 1986; Brandt, 1986; Falgout, 2006 e Haase e Langer, 2002), em que é necessária a matriz de coeficientes. Desta matriz são retiradas informações a respeito da malha inicial, convencionada como a malha com mais incógnitas.

O que os dois métodos têm em comum é o fato de o problema ser resolvido empregando uma hierarquia de malhas que, para o GMG tem que ser conhecida previamente. Para o AMG esta hierarquia é gerada na fase *setup* (uma fase considerada inicial) partindo da matriz de coeficientes. Nas duas abordagens é necessário: gerar as malhas auxiliares, transferir informações entre as malhas (operadores de restrição e prolongação), resolver os sistemas lineares em cada malha com o uso de um método iterativo (*solver*) e optar por um dos tipos de ciclos (sequência com que as diversas malhas são visitadas) disponíveis na literatura (Briggs et al, 2000 e Trottenberg et al, 2001). 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 as componentes do método de acordo com o delineamento físico do problema e a formulação variacional.

De acordo com Ruge e Stuben (1986), o método *multigrid* algébrico pode ser aplicado a problemas onde a aplicação do *multigrid* geométrico é difícil ou inviável. Exemplos destas situações são dados por problemas discretizados em malhas não-estruturadas, quando as informações são dadas apenas por uma matriz, quando o domínio

de cálculo é muito complexo, entre outros. Uma comparação entre os métodos *multigrid* algébrico e geométrico pode ser observada na Tab. (1), adaptada de Chang et al (1996).

**Tabela 1. Comparações entre *multigrid* geométrico e *multigrid* algébrico (Adaptada de Chang et al, 1996).**

Característica	<i>Multigrid</i> Geométrico	<i>Multigrid</i> Algébrico
Problema a ser resolvido	Problemas contínuos	Sistemas lineares de equações algébricas
Informação usada	Estrutura geométrica do problema	Somente as entradas da matriz
Algoritmo (Programa)	Preparado para cada problema	Único para todos os problemas
Eficiência	Muito boa	Boa

O método *multigrid* algébrico pode ser aplicado a problemas em que as localizações dos pontos da malha são conhecidas, mas são não-estruturadas ou irregulares. Stüben (2001) ressalta que este método pode ser aplicado quando não se tem nenhuma informação a respeito da geometria do problema. De acordo com Trottenberg et al (2001), as vantagens deste método são robustez, aplicabilidade em situações com geometrias complexas e resolução de problemas que estão fora do alcance do *multigrid* geométrico.

O objetivo deste trabalho é minimizar o tempo de CPU para a resolução do AMG através do estudo das componentes do algoritmo do método. Este trabalho apresenta comparações do número ótimo de iterações internas e do número ótimo de níveis de malhas para o AMG e GMG. Estas comparações consideram os resultados obtidos com malhas estruturadas quadrangulares, para as equações de Laplace e Poisson, e com malhas estruturadas triangulares, para a equação de Laplace. Este trabalho também investiga o comportamento de dois parâmetros que influenciam diretamente a geração das malhas auxiliares do AMG: fator de redução de malha e fator de forte dependência na malha grossa.

Vários trabalhos presentes na literatura trazem comparações entre o AMG e o GMG. Estas comparações limitam-se a: tempo de CPU, número de ciclos e eficiência do método *multigrid* pelo estudo do *speedup*. Watanabe et al (2005) observaram o crescimento do tempo de CPU conforme o número de incógnitas é aumentado, tanto para o AMG quanto para o GMG. Langer e Pusch (2006) trazem comparações do 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) também 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) traz a comparação da performance do AMG com GMG, ambos pré-condicionados e com algoritmos paralelos para um sistema não-linear de equações diferenciais. São variados número de níveis do GMG e número de níveis e fator de redução de malha para o AMG, quando são utilizados diferentes números de processadores. O trabalho de Gaspar et al (2009) aplica o GMG a malhas triangulares. Trazem resultados para número de iterações internas e tipos de ciclos para o GMG.

Durante a revisão bibliográfica não foram encontrados trabalhos que tratem da otimização do algoritmo do AMG e nem trabalhos que tragam resultados comparativos para os parâmetros estudados no presente trabalho. Para o GMG tem-se resultados, em malhas quadrangulares, para o número de iterações internas e número de malhas em Oliveira et al (2008). Para malhas triangulares tem-se o trabalho de Gaspar et al (2009) que traz resultados para o número de iterações internas para o GMG.

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 (AMG)

O método *multigrid* algébrico é dividido em duas etapas. A fase inicial, chamada fase *setup*, responsável pela geração das malhas e pela construção dos operadores de transferência entre as malhas (restrição e prolongação). A fase de solução é direta e emprega os operadores definidos na fase *setup* para se 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  $\Omega^h$  denota o conjunto de índices  $\{1, 2, \dots, n\}$ . Esta partição resultará em dois subconjuntos disjuntos: o conjunto  $C$ , que é composto pelos pontos que estão na malha grossa e seu complementar, 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 alguns conjuntos, conforme Eqs. (1), (2) e (3):

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

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

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

onde  $N_i$  é a vizinhança do ponto  $i$ ,  $S_i$  determina o conjunto de pontos que influenciam fortemente  $i$  e  $S_i^T$  é o conjunto de pontos que dependem fortemente de  $i$ . O parâmetro  $\theta$  ( $0 \leq \theta \leq 1$ ), que aqui será chamado de fator de redução de malha, é uma constante que 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  $\theta$ , 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.

Depois de feita a 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  $D_i^w = D_i - S_i$ , onde estão os pontos que podem estar tanto em  $C$  quanto em  $F$ , que são chamados de pontos fracamente conectados. O operador de interpolação (que transfere as informações entre as malhas) é dado pelas Eqs. (4) e (5) (Briggs et al, 2000):

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

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 também é objeto de estudo neste trabalho é o fator de forte dependência na malha grossa ( $\varepsilon$ ). 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 a interpolação seja influenciada. Para tanto, é necessário definir um conjunto de dados, conforme Eq. (6) (Iwamura et al, 2003):

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

O parâmetro  $\varepsilon$  também é uma constante e seu valor, que dever ser positivo, está 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. Mais detalhes a respeito do algoritmo empregado para a interpolação podem ser encontrados em Iwamura et al (2003).

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 com base na estimativa inicial e tolerância de  $10^{-8}$ . 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, de forma que, para cada tipo de malha (quadrangular ou triangular) existe um gerador de matrizes específico, acoplado ao programa principal.

### 3. MODELOS MATEMÁTICO E NUMÉRICO

Para malhas estruturadas quadrangulares, as equações de Laplace e Poisson são discretizadas com o uso do método de diferenças finitas (Tannehill et al, 1997), em um domínio de cálculo quadrado unitário, sendo que as condições de contorno são do tipo Dirichlet. As aproximações empregadas para as derivadas da Eq. (7) são do tipo CDS-2. Para as duas equações são conhecidas as soluções analíticas, conforme dado pela Eq. (7) e pela Tab. (2):

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = S \quad (7)$$

**Tabela 2. Problemas resolvidos.**

Termo Fonte	Solução Analítica	Condições de Contorno
$S = 0$	$T(x, y) = xy$	$T(0, y) = T(x, 0) = 0, T(x, 1) = x,$ $T(1, y) = y$
$S = 0$	$T(x, y) = \frac{\sinh(\pi y)}{\sinh(\pi)}$	$T(0, y) = T(x, 0) = T(1, y) = 0,$ $T(x, 1) = \sinh(\pi x)$
$S = -2[(1-6x^2)y^2(1-y^2) + (1-6y^2)x^2(1-x^2)]$	$T(x, y) = (x^2 - x^4)(y^4 - y^2)$	$T(0, y) = T(x, 0) = 0,$ $T(1, y) = T(x, 1) = 0$

Para malhas estruturadas triangulares, a equação de Laplace é discretizada em um domínio de cálculo quadrado unitário, com o uso do método de volumes finitos (Maliska, 2004 e Ferziger e Peric, 2002). A aplicação das condições de contorno é feita com o uso de volumes fictícios. Nesta forma de discretização, a equação diferencial parcial na forma conservativa é integrada. As equações de conservação, aproximadas para os volumes de controle através de diferença central de segunda ordem de acurácia, são obtidas através da montagem elemento por elemento. Nos dois métodos de discretização chega-se a um sistema de equações algébricas, do tipo  $AT=b$ , onde  $A$  corresponde à matriz de coeficientes,  $T$  representa o vetor de incógnitas e  $b$  é o termo fonte.

### 4. RESULTADOS

Cerca de 700 simulações foram realizadas, sendo variados: número de iterações internas no *solver* ( $v$ ), número de níveis de malha ( $L$ ), fator de redução de malha ( $\theta$ ), fator de forte dependência na malha grossa ( $\epsilon$ ) e número de incógnitas ( $N$ ). Estes parâmetros são estudados para a Equação de Laplace, discretizada em malhas estruturadas quadrangulares e triangulares e para a equação de Poisson discretizada em malhas estruturadas quadrangulares. Para a obtenção dos resultados são considerados três tamanhos de problema para os dois tipos de malha. Para os testes dos quatro primeiros parâmetros, para malhas quadrangulares, são considerados: problema pequeno ( $257 \times 257$  incógnitas), problema médio ( $1025 \times 1025$  incógnitas) e problema grande ( $4097 \times 4097$  incógnitas). Para malhas estruturadas triangulares, são considerados: problema pequeno ( $2^{18}=262144$  incógnitas), problema médio ( $2^{20}=1048576$  incógnitas) e problema grande ( $2^{24}=16777216$  incógnitas). Para avaliar o efeito do número de incógnitas ( $N$ ) no tempo de CPU são considerados todos os tamanhos de problema possíveis até ser atingido o tamanho máximo (malhas quadrangulares:  $5 \times 5$  até  $4097 \times 4097$  incógnitas e malhas triangulares:  $2^2=4$  até  $2^{24}=16777216$  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 ( $v$ )

Para se encontrar o valor ótimo para este parâmetro são feitos testes para os três tamanhos de problema descritos acima, tanto para malhas quadrangulares quanto para malhas triangulares. O valor de  $v$  é variado de 1 até 20. Para os problemas considerados pequeno e médio, para os quais as simulações tem tempo de CPU inferior a 10 segundos, as simulações são repetidas 3 vezes, sendo que é tomada a média aritmética destes tempos.

Na Fig. (1), podem ser observados os resultados obtidos para cada uma das equações estudadas. Esta figura traz resultados para o número de iterações internas para malhas quadrangulares com  $N=4097 \times 4097=16785409$  incógnitas e triangulares com  $N=2^{24}=16777216$  incógnitas. Os resultados obtidos para os outros tamanhos de problema são qualitativamente iguais. O número ótimo de iterações internas para malhas quadrangulares, independente do problema a ser resolvido, é 2; ou seja,  $v_{\text{ótimo}}=2$ . Para malhas triangulares,  $v_{\text{ótimo}}=1$ . Para os dois tipos de malha, observa-se que, quanto maior o número de iterações internas, maior é o tempo de CPU gasto para 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  $v=1$  para a resolução dos problemas, para qualquer tipo de malha. Para malhas quadrangulares, observou-se que o valor de  $v_{\text{ótimo}}$  coincide com o encontrado para o GMG, em Oliveira et al (2008). O

trabalho de Gaspar et al (2009), que traz o GMG aplicado a malhas triangulares, também conclui que o valor ótimo para  $v$  é 1.

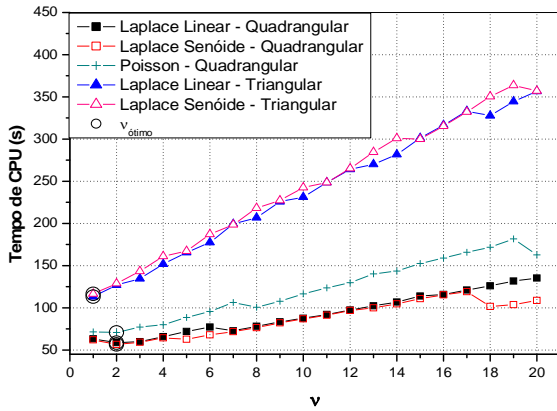


Figura 1. Número de iterações internas ( $v$ ) para malhas quadrangulares e triangulares.

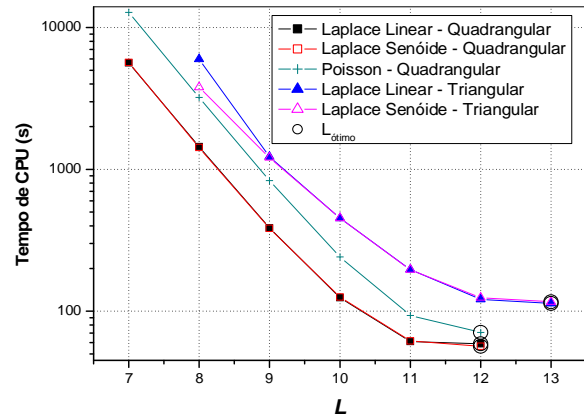


Figura 2. Número de níveis ( $L$ ) para malhas quadrangulares e triangulares.

Com base nos resultados obtidos, pode-se concluir também que o tipo de malha empregado, quadrangular ou triangular, influencia no número ótimo de iterações internas. Com esta otimização para malhas quadrangulares, obteve-se um ganho de 7,5%, 9,9% e 1,1% para as equações de Laplace (Linear e Senóide) e Poisson, respectivamente. Este ganho é com relação ao valor amplamente empregado pela literatura,  $v=1$ .

#### 4.2. Número de níveis ( $L$ )

Para o estudo da influência do número de níveis no tempo de CPU é utilizado o valor ótimo para o número de iterações internas para cada caso analisado. Para malhas quadrangulares,  $v=2$  e para malhas triangulares,  $v=1$ . Na Fig. (2) pode-se observar o comportamento do tempo de CPU quando é variado o número de níveis para malhas quadrangulares e triangulares. Desta análise, pode-se concluir que o número de malhas a ser empregado na resolução do AMG influencia no tempo de CPU, para os dois tipos de discretização aqui empregados. Observa-se que, quanto menor o número de níveis de malha empregados, maior é o tempo de CPU para o AMG.

Estes resultados concordam com os obtidos para o GMG em Oliveira et al (2008), onde o número ótimo de níveis também é o máximo. Durante a revisão bibliográfica para o AMG, observou-se que alguns autores, como Ruge e Stüben (1986), Falgout (2006) e Langer e Pusch (2006) não se preocupam com o número de malhas a ser empregado. Nos trabalhos em que este dado está explícito, nota-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). O trabalho que aplica o GMG a malhas triangulares, Gaspar et al (2009), também utiliza todos os níveis de malha na resolução do problema.

Não são encontrados estudos a respeito do efeito do número de malhas no tempo de CPU para o AMG. Pode-se concluir que o tipo de malha aqui empregada (quadrangular ou triangular) não exerce influência no valor ótimo encontrado para o número de malhas. Os resultados apresentados acima são para as malhas mais refinadas, sendo que, para os outros tamanhos de problema, os resultados obtidos são qualitativamente iguais.

#### 4.3. Fator de redução de malha ( $\theta$ )

Para o estudo deste parâmetro são considerados os valores ótimos obtidos para o número de iterações internas, ( $v=2$  em malhas quadrangulares e  $v=1$  em malhas triangulares). Para o número de malhas é considerado  $L=L_{máximo}$ . 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  $\theta$ , apenas ressaltam que  $\theta=0,25$  é um valor padrão. Não se conhece na literatura um estudo qualitativo deste parâmetro para o AMG. Por analogia ao GMG, observou-se que  $\theta=0,25$  ou  $\theta=1/4$  corresponde à interpolação de quatro pontos para a obtenção do ponto da malha imediatamente mais fina, o 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 à razão de engrossamento  $r=3$  e  $\theta=0,0625$  corresponde à razão  $r=4$ .

Na Fig. (3) pode-se observar o comportamento do fator de redução de malha para os problemas resolvidos em malhas estruturadas quadrangulares e triangulares. Os resultados ilustrados nesta figura são os obtidos para as malhas mais refinadas, sendo que para os outros tamanhos de problema, os resultados são qualitativamente iguais. Conforme

pode ser observado na Fig. (3), o fator de redução de malha apresenta valores ótimos diferentes para os problemas abordados, que em geral é diferente do valor considerado padrão na literatura ( $\theta=0,25$ ).

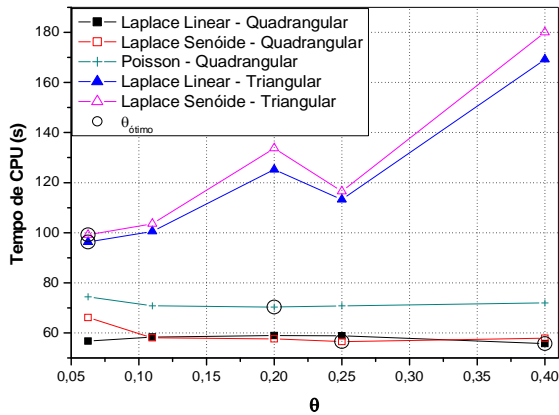


Figura 3. Fator de redução de malha ( $\theta$ ) para malhas quadrangulares e triangulares.

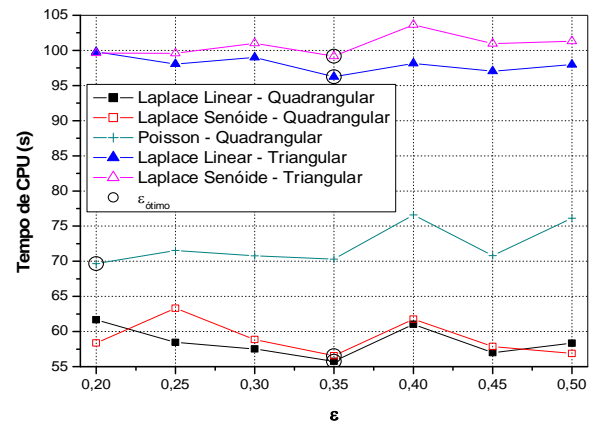


Figura 4. Fator de forte dependência na malha grossa ( $\epsilon$ ) para malhas quadrangulares e triangulares.

Para malhas quadrangulares, o valor empregado para  $\theta$  não influencia na construção das malhas auxiliares. Isto ocorre pois todos os vizinhos estão fortemente conectados entre si, independente do valor empregado para  $\theta$ , ou seja, todos os pontos sempre satisfazem a Eq. (1). Esta conclusão pode ser extraída da Fig. (3), onde nota-se a pequena variação entre os tempos de CPU obtidos para os valores de  $\theta$  em malhas quadrangulares.

Para malhas triangulares, a situação descrita acima não se aplica, pois na vizinhança de um ponto tem-se outros três (que podem ou não satisfazer a Eq. (1)). Assim, o valor do fator de redução de malha faz diferença para a construção das malhas triangulares, ou seja, quanto menor, melhor para os casos analisados, conforme pode ser observado na Fig. (3). Isto se deve ao fato de que, quanto menor o valor de  $\theta$ , mais pontos serão usados para a construção dos níveis de malha auxiliares.

#### 4.4. Fator de forte dependência na malha grossa ( $\epsilon$ )

Para se encontrar o valor ótimo para  $\epsilon$ , 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 irá influenciar 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  $\epsilon=0,35$ . Em Iwamura et al (2003), são empregados valores diferentes conforme o nível de malha que está sendo construído. Para este trabalho tem-se  $\epsilon=0,35$  para o número de níveis menor ou igual a 3 e,  $\epsilon=0,45$  quando o número de níveis for maior que 3. Em Trottenberg et al (2001) é utilizado  $\epsilon=0,20$  na resolução da equação de Poisson. No presente trabalho são testados vários valores para  $\epsilon$ .

Na Fig. (4), pode-se observar o comportamento deste parâmetro para malhas quadrangulares e triangulares. Os resultados ilustrados são os obtidos para os problemas resolvidos com mais incógnitas, sendo que os resultados para os problemas menores são qualitativamente iguais. Para a equação de Laplace, tanto para malhas quadrangulares, quanto para malhas triangulares, o valor ótimo para o fator de forte dependência na malha grossa é o mesmo, ou seja,  $\epsilon=0,35$ . Este resultado concorda com a literatura. Para a equação de Poisson este valor é diferente,  $\epsilon=0,20$ . Este último resultado concorda com Trottenberg et al (2001) que também resolve a equação de Poisson com o uso do AMG.

#### 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, foram tomados os parâmetros ótimos obtidos anteriormente. Para o método *multigrid*, tanto algébrico quanto geométrico, em todos os problemas abordados, o expoente  $p$  (da equação  $t=cN^p$ ), manteve-se próximo da unidade conforme o esperado. Para o método *singlegrid* (SG - método de malha única), este valor manteve-se próximo de dois. Os valores para  $p$  mostrados na Tab. (3) são obtidos por meio do ajuste de curvas pelo método dos mínimos quadrados, onde são utilizados os 4 pontos de cada curva com maior  $N$ , para se fazer o ajuste.

Nas Figs. (5) e (6), pode-se observar o comportamento do tempo de CPU conforme é variado o número de incógnitas para o AMG, GMG e SG em malhas quadrangulares e para o AMG e SG, em malhas triangulares, respectivamente. Os resultados apresentados nas Figs. (5) e (6) são complementares aos apresentados na Tab. (3). Os

valores encontrados para  $p$  estão próximos para o AMG e GMG. Isto reflete o fato destas curvas apresentarem praticamente a mesma inclinação. A inclinação para a curva que representa o SG, quando comparado ao AMG e GMG, é diferente, fato que também pode ser confirmado pela Tab. (3), onde é observada a grande diferença entre os valores de  $p$  para estes métodos.

Tabela 3. Valor de  $p$  em  $t=cN^p$ .

Problema resolvido	AMG	GMG	SG
Laplace Linear Quadrangular	1,05	1,09	1,92
Laplace Senóide Quadrangular	1,00	1,07	1,91
Poisson Quadrangular	1,04	1,04	2,04
Laplace Linear Triangular	1,05	-----	1,88
Laplace Senóide Triangular	1,04	-----	1,88

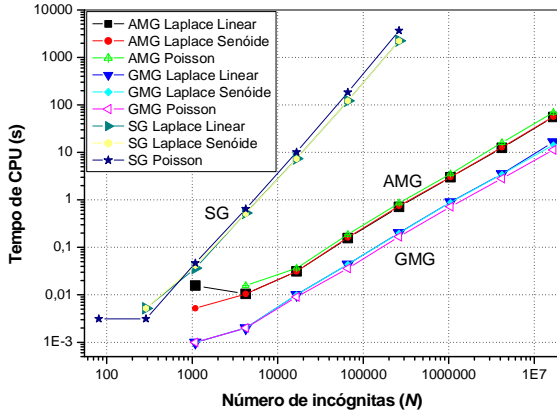


Figura 5. Tempo de CPU para AMG, GMG e SG para malhas quadrangulares.

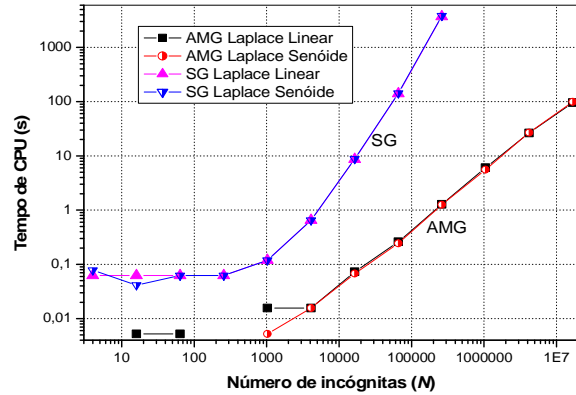


Figura 6. Tempo de CPU para AMG e SG para malhas triangulares.

Na Fig. (5), que ilustra os resultados em malhas quadrangulares, pode-se observar que as curvas para o GMG e para o AMG possuem a mesma inclinação, para os três problemas analisados. Nota-se ainda que estas curvas, para o AMG e GMG, estão um pouco afastadas. Este comportamento é esperado, já que o GMG é mais rápido para malhas estruturadas quadrangulares. A razão entre os tempos de CPU obtidos para o AMG e GMG é de aproximadamente 5, ou seja, o GMG resolve o problema em 20% do tempo gasto pelo AMG. Na Fig. (5) nota-se ainda que a curva para o *singlegrid*, além de estar mais afastada das demais, possui inclinação diferente. Estes resultados concordam com os obtidos por Watanabe et al (2005), que trazem um gráfico onde são comparados os tempos de CPU para o AMG e GMG.

Para malhas triangulares, na Fig. (6), é observada a inclinação diferente para as curvas que correspondem ao AMG e ao SG, além destas curvas estarem bem afastadas. Este comportamento também é o esperado para malhas triangulares.

#### 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 analisados. Com este conjunto de otimizações, são obtidos ganhos em tempo de CPU 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  $\epsilon=0,35$ ) e o algoritmo otimizado cresce conforme aumenta o número de incógnitas. O algoritmo ótimo contempla os parâmetros ótimos obtidos para cada problema acima estudado. Na Tab. (4) pode ser observada a razão entre os tempos de CPU para o programa otimizado e o padrão. Nesta tabela também podem ser observados os tempos de CPU para os problemas resolvidos com mais incógnitas.

Conforme a Tab. (4), pode-se concluir que são alcançados ganhos em termos de tempo de CPU para todos os problemas resolvidos. A redução obtida do algoritmo otimizado em relação ao padrão é de 8% a 15%. O problema que obteve o menor ganho é o de Poisson, cuja resolução com o algoritmo otimizado leva 92% do tempo obtido com o algoritmo padrão. Os problemas que obtiveram melhores ganhos são os resolvidos em malhas triangulares. Para a equação de Laplace, cuja solução analítica é dada por uma função linear, o algoritmo otimizado leva 85% do tempo de CPU obtido com o algoritmo padrão.

Na Tab. (5), podem ser observados os valores obtidos para o valor de  $p$  (em  $t=cN^p$ ) tanto para o AMG otimizado, quanto para o AMG considerado padrão. Os valores do algoritmo otimizado são menores (estão mais próximos de um) que os valores obtidos quando é considerado o algoritmo padrão. Nesta tabela, pode-se observar ainda o número de

ciclos (iterações externas) que são realizados para o AMG atingir a tolerância estipulada. Nota-se uma redução neste número para o algoritmo otimizado. Para a equação de Laplace resolvida em malhas quadrangulares, esta redução é de quase metade.

**Tabela 4. Comparações do tempo de CPU entre o algoritmo padrão e o otimizado.**

Problema resolvido	Tempo de CPU (s)		$\frac{t_{\text{CPU}}(\text{otimizado})}{t_{\text{CPU}}(\text{padrão})}$
	Padrão	Otimizado	
Laplace Linear Quadrangular	62,2	55,8	0,90
Laplace Senóide Quadrangular	63,1	56,6	0,90
Poisson Quadrangular	75,6	69,7	0,92
Laplace Linear Triangular	112,8	96,2	0,85
Laplace Senóide Triangular	115,1	99,2	0,86

**Tabela 5. Valor de  $p$  em  $t=cN^p$ .**

Problema resolvido	AMG – Padrão	AMG – Otimizado	Ciclos – Padrão	Ciclos – Otimizado
Laplace Linear Quadrangular	1,06	1,05	10	6
Laplace Senóide Quadrangular	1,05	1,00	10	6
Poisson Quadrangular	1,12	1,04	13	9
Laplace Linear Triangular	1,08	1,05	29	25
Laplace Senóide Triangular	1,07	1,04	30	25

## 5. CONCLUSÃO

Neste trabalho são testados alguns parâmetros do método *multigrid* algébrico na resolução das equações de Laplace e Poisson bidimensionais, sendo que a equação de Laplace é discretizada em malhas estruturadas quadrangulares e triangulares, enquanto que a equação de Poisson é resolvida apenas em malhas quadrangulares. Para malhas quadrangulares emprega-se o método de diferenças finitas para a discretização das equações diferenciais parciais, e para as malhas triangulares, utiliza-se o método de volumes finitos. Foram analisados os efeitos dos seguintes parâmetros sobre o tempo de CPU: número de iterações internas ( $v$ ), número de malhas ( $L$ ), fator de redução de malha ( $\theta$ ), fator de forte dependência na malha grossa ( $\epsilon$ ) e número de incógnitas ( $N$ ). Alguns dos resultados obtidos são comparados aos disponíveis para o GMG.

Com a realização deste trabalho, verificou-se que:

- 1) O número de iterações internas afeta significativamente o tempo de CPU. Para as equações de Laplace e Poisson discretizadas em malhas quadrangulares,  $v_{\text{ótimo}}=2$ . Para malhas triangulares, encontrou-se  $v_{\text{ótimo}}=1$ .
- 2) O número de níveis empregados para a resolução do AMG afeta significativamente o tempo de CPU. Para as equações acima citadas, observou-se que  $L_{\text{ótimo}}=L_{\text{máximo}}$ , para malhas quadrangulares e triangulares. Para o GMG o ótimo encontrado também é o número máximo de malhas.
- 3) O fator de redução de malha varia conforme o problema que está sendo abordado: em malhas quadrangulares, para a equação de Laplace (Linear) tem-se  $\theta=0,20$ ; para Laplace (Senóide), tem-se  $\theta=0,25$ ; e para Poisson, tem-se  $\theta=0,40$ . Para malhas triangulares o valor ótimo é  $\theta=0,0625$ .
- 4) O fator de forte dependência na malha grossa também apresenta variações conforme o problema resolvido. Em malhas quadrangulares, para a equação de Laplace, tanto linear quanto senóide, tem-se  $\epsilon=0,35$ . Para Poisson,  $\epsilon=0,20$ . Para malhas triangulares o valor ótimo é  $\epsilon=0,35$ .
- 5) Com as otimizações feitas foram reduzidos o tempo de CPU, o valor do expoente  $p$  e o número de iterações externas para todos os casos analisados.
- 6) Foram obtidos resultados em malhas mais refinadas que as apresentadas na literatura, tanto para malhas triangulares quanto para malhas quadrangulares.
- 7) Para malhas quadrangulares, o GMG resolve o problema em 20% do tempo gasto pelo AMG.

## 6. AGRADECIMENTOS

A primeira autora agradece ao Laboratório de Experimentação Numérica (LENA), do Departamento de Engenharia Mecânica da UFPR, por disponibilizar sua estrutura, aos amigos do LENA e a CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) pelo suporte financeiro. 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.



## 7. 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. e McCormick, S. F., 2000, "A *Multigrid* Tutorial", 2ª Ed., SIAM, EUA, 193p.
- Falgout, R. D., 2006, "An Introduction to Algebraic *Multigrid*", Computing in Science and Engineering, pp. 1-11.
- Campos, F. O., Oliveira, R. S. e 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. e 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. e Ruge, J. W., 2000, "Robustness and Scalability of Algebraic *Multigrid*", SIAM Journal on Scientific Computing, Vol. 21, Nº 5, pp. 1886-1908.
- Ferziger, J. H. e Peric, M., 2002, "Computational Methods for Fluid Dynamics", Berlin: Springer-Verlag, 423 p.
- Gaspar, F. J., Gracia, J. L., Lisbona, F. J. e Rodrigo, C., 2009, "On geometric *multigrid* methods for triangular grids using three-coarsening strategy", Applied Numerical Mathematics, Article in press.
- Haase, G. e 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. e 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. e Stüben, K., 1999, "Operator Dependent Interpolation in Algebraic *Multigrid*", Proceedings of Fifth European *Multigrid* Conference, 24 p.
- Langer, U. e Pusch, D., 2006, "Comparison of Geometrical and Algebraic *Multigrid* Preconditioners for Data-Sparse Boundary Element Matrices", Springer-Verlag Berlin Heidelberg, pp. 130-137.
- Maliska, C. R., 2004, "Transferência de Calor e Mecânica dos Fluidos Computacional", 2ª Ed., Editora LTC, 453 p.
- Oliveira, F., Pinto, M. A. V. e Marchi, C. H., 2008, "Efeito de roteiros do método *multigrid* sobre o tempo de CPU para a equação de Laplace 2D", Proceedings of the XXIX Congresso Ibero-Latino-Americano de Métodos Computacionais em Engenharia, Maceió, Alagoas, Brasil, 21p.
- Ruge, J. e Stüben, K., 1986, "Algebraic *Multigrid* (AMG)", em S. F. McCormick, *Multigrid* Methods, volume 5 de Frontiers in Applied Mathematics, SIAM, Philadelphia, 57 p.
- 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. e Pletcher, R. H., 1997, "Computational Fluid Mechanics and Heat Transfer", Taylor e Francis.
- Trottenberg, U., Oosterlee, C. e Schüller, A., 2001, "*Multigrid*", Academic Press, 631 p.
- Watanabe, K., Igarashi, H. e Honma, T., 2005, "Comparison of Geometric and Algebraic *Multigrid* Methods in Edge-Based Finite-Element Analysis", IEEE Transactions on Magnetics, Vol. 41, Nº. 5, pp. 1672-1675.
- Wu, C., T. e Elman, H. C., 2006, "Analysis and Comparison of Geometric and Algebraic *Multigrid* for Convection-Diffusion Equations", SIAM Journal on Scientific Computing, Vol. 28, Nº 6, pp. 2208-2228.



VI CONGRESSO NACIONAL DE ENGENHARIA MECÂNICA  
VI NATIONAL CONGRESS OF MECHANICAL ENGINEERING  
18 a 21 de agosto de 2010 – Campina Grande – Paraíba - Brasil  
August 18 – 21, 2010 – Campina Grande – Paraíba – Brazil

## OPTIMIZATION OF THE ALGEBRAIC MULTIGRID METHOD FOR THE TWO-DIMENSIONAL EQUATIONS OF LAPLACE AND POISSON

Roberta Suero, [roberta.suero@ifpr.edu.br](mailto:roberta.suero@ifpr.edu.br)<sup>1,2</sup>  
Marcio Augusto Villela Pinto, [marcio\\_villela@yahoo.com.br](mailto:marcio_villela@yahoo.com.br)<sup>3,4</sup>  
Carlos Henrique Marchi, [marchi@ufpr.br](mailto:marchi@ufpr.br)<sup>4</sup>  
Luciano Kiyoshi Araki, [lucaraki@ufpr.br](mailto:lucaraki@ufpr.br)<sup>4</sup>  
Arleide Cristina Alves, [aalves@up.edu.br](mailto:aalves@up.edu.br)<sup>5</sup>

<sup>1</sup> Post-Graduate Program in Numerical Methods in Engineering, Federal University of Paraná, Curitiba, PR

<sup>2</sup> Federal Institute of Education, Science and Technology of Paraná, Campus Paranaguá, PR

<sup>3</sup> Department of Mathematics and Statistics, State University of Ponta Grossa, Ponta Grossa, PR

<sup>4</sup> Department of Mechanical Engineering, Federal University of Paraná, Curitiba, PR

<sup>5</sup> School of Exact and Technology Sciences, Positivo University, Curitiba, PR

**Abstract.** This work presents comparisons of parameters between the algebraic multigrid (AMG) and geometric multigrid (GMG) methods for Laplace and Poisson two-dimensional equations in square and triangular structured grids. The analyzed parameters are: the number of internal iterations in the solver, the number of grids and number of unknowns. For AMG, the effects of the grid reduction factor and the strong dependence factor in the coarse grid on the necessary CPU time to obtain the numeric solution are studied. For square grids the finite difference method is used, and for the triangular grids, the finite volume one. The results are obtained with the use of an adapted computational code from the original AMGIR6 of Ruge and Stüben. For the AMG the following multigrid components are used: restriction by the standard coarsening, standard interpolation, correction scheme (CS), lexicographic Gauss-Seidel as solver and V cycle. Comparative studies among the CPU time of the geometric and algebraic multigrid methods and singlegrid (method of unique mesh) are made. It was verified that: 1) the optimum number of internal iterations obtained for AMG and GMG, in square grids, is the same, however it has a value different for triangular grids; 2) the optimum number of grids is the maximum number, for both square and triangular grids; 3) AMG was shown to be sensitive to both the variation of the grid reduction factor and the strong dependence factor in the coarse grid, in relation to the approached equations, as like to the mesh types; and 4) in square grids, the GMG solves the problem in 20% of the time spend for AMG.

**Keywords:** finite difference, finite volume, optimization.