

Efeito de componentes do *full multigrid* sobre o tempo de CPU em problemas 2D de CFD

Simone F. T. Gonçalves, Carlos H. Marchi, Marcio A. V. Pinto, Luciano K. Araki

Depto. Engenharia Mecânica, Universidade Federal do Paraná (UFPR), Curitiba, PR
E-mails: simone.tg@ufpr.br; marchi@ufpr.br; marcio_villela@ufpr.br; lucaraki@ufpr.br

Resumo: Neste trabalho analisou-se o efeito de diversos parâmetros algorítmicos no desempenho do Full Multigrid (FMG) para três equações bidimensionais: Laplace, advecção-difusão e Burgers; discretizadas com o Método dos Volumes Finitos (MVF). Foram investigados os efeitos do número de iterações internas do solver, do número de malhas, do número de incógnitas, dos solvers Gauss-Seidel Lexicográfico (GS-Lex) e Gauss-Seidel Red-Black (GS-RB), e seis operadores de prolongação na interpolação da solução entre os níveis do FMG. Verificou-se que o GS-RB é mais eficiente, o operador bilinear foi o que apresentou melhor desempenho, e o emprego de operadores de alta ordem pode não ser a escolha mais vantajosa na redução do tempo de CPU.

Palavras-chave: *multigrid* geométrico, *full multigrid*, otimização do tempo de CPU, operadores de transferência, erro de iteração.

1 Introdução

O método *multigrid* (MG) tem se mostrado muito eficiente para acelerar a taxa de convergência dos *solvers* iterativos empregados na resolução de sistemas lineares [11,14]. A análise espectral dos erros da solução mostra que apenas os erros que possuem comprimento de onda da ordem do espaçamento da malha é que são reduzidos com rapidez nos *solvers* iterativos [3,15]. Neste sentido, o MG consiste no uso de várias malhas com diferentes graus de refinamento, percorridas ao longo do processo iterativo. Assim, a cada nível de refinamento da malha, as diferentes componentes do erro são eficientemente suavizadas, acelerando a convergência do processo iterativo.

Experiências com o MG mostram que uma boa combinação de componentes como *solver*, número de iterações em cada ciclo, assim como o tipo de ciclo, pode ser essencial na redução do tempo de CPU. A qualidade da transferência de informações entre as malhas é apontada como um dos principais fatores que melhoram o desempenho do MG [3, 4, 10,15,16]. Essa transferência é feita mediante operadores de restrição (malha fina para a malha grossa) e prolongação (malha grossa para a malha fina). A sequência com que as malhas são visitadas define diferentes ciclos, designados como V, W e F, entre outros.

No âmbito do MG são distinguidos o geométrico (*Geometrical Multigrid* – GMG) e o algébrico (*Algebraic Multigrid* – AMG). O primeiro é o mais indicado para malhas estruturadas. Para ambos os tipos, há dois modos de se construir o algoritmo MG: o esquema de aproximação completa (*Full Approximation Scheme* – FAS) e o esquema de correção (*Correction Scheme* – CS). No esquema CS, somente o resíduo é transferido para as malhas grossas, onde é resolvida a equação residual. Já no esquema FAS, são transferidos o resíduo e a aproximação da solução para as malhas grossas [13].

Uma boa estimativa inicial para o processo iterativo pode ser obtida interpolando-se uma solução da malha grossa para a malha fina. Esta estratégia dá origem ao FMG, cuja ideia está representada na Figura 1. Inicia-se o processo iterativo na malha mais grossa. Então, a solução é interpolada para a segunda malha mais grossa e um ciclo V é percorrido. Estes dois passos são repetidos recursivamente até que a malha mais fina seja atingida.

No FMG o operador de interpolação não precisa ser o mesmo usado nos ciclos de correção. Além disso, frequentemente sua ordem de aproximação é maior do que a ordem do operador empregado nas correções, sendo o operador bicúbico o mais comum [2,3,15,16].

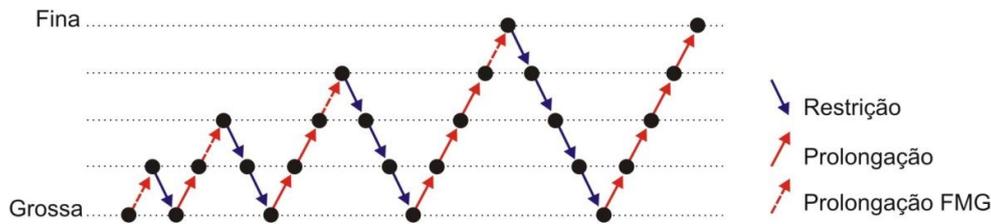


Figura 1: Ilustração do FMG com cinco malhas

2 Modelos matemáticos e numéricos

A Tabela 1 apresenta os três problemas investigados neste trabalho. Em todos os casos adotou-se $\Omega = [0,1] \times [0,1]$ e condições de contorno de Dirichlet, aplicadas mediante a técnica dos volumes fictícios. Para se obter as equações algébricas, adotou-se o MVF e malhas uniformes. Neste procedimento, os termos difusivos e advectivos são aproximados com o esquema CDS de segunda ordem.

Problema	Equações e condições de contorno	Solução analítica
(I) Laplace	$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0$ $\begin{cases} T(x,0) = T(1,y) = T(0,y) = 0 \\ T(x,1) = \text{sen}(\pi x) \end{cases}$	$T(x,y) = \text{sen}(\pi x) \frac{\text{senh}(\pi y)}{\text{senh}(\pi)}$
(II) Advecção-difusão	$Pe_x \frac{\partial T}{\partial x} + Pe_y \frac{\partial T}{\partial y} = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + S$ $\begin{cases} T(0,y) = T(x,0) = T(1,y) = 0 \\ T(x,1) = \text{sen}(\pi x) \frac{(e^{Pe_x} - 1)}{(e^{Pe_x} - 1)} \end{cases}$	$T(x,y) = \text{sen}(\pi x) \frac{(e^{Pe_x} - 1)(e^{Pe_y} - 1)}{(e^{Pe_x} - 1)(e^{Pe_y} - 1)}$ <p>S dado em [12] e Pe_x, Pe_y dados em [7]</p>
(III) Burgers	$\frac{\partial u^2}{\partial x} + \frac{\partial(uv)}{\partial y} = -\frac{\partial p}{\partial x} + \frac{1}{\text{Re}} \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right)$ $\frac{\partial(uv)}{\partial x} + \frac{\partial v^2}{\partial y} = -\frac{\partial p}{\partial y} + \frac{1}{\text{Re}} \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) - S$ $\begin{cases} u(x,0) = u(0,y) = u(1,y) = v(x,0) = v(0,y) = 0 \\ v(1,y) = v(x,1) = 0, u(x,1) = 16(x^4 - 2x^3 + x^2) \end{cases}$	$u(x,y) = 8(x^4 - 2x^3 + x^2)(4y^3 - 2y)$ $v(x,y) = -8(4x^3 - 6x^2 + 2x)(y^4 - y^2)$ <p>p e S dados em [8]</p>

Tabela 1: Definição dos modelos matemáticos

Todos os sistemas de equações algébricas foram resolvidos com o GMG, esquema FAS-FMG e ciclo V. O número ótimo de iterações internas foi definido a partir de testes que resultaram em $v_{\text{ótimo}} = 1$ ou 2 (I e II) e $v_{\text{ótimo}} = 2$ ou 3 (III). A seleção das malhas é baseada no tamanho do problema e na razão de engrossamento adotada. Considerando-se um problema com razão de engrossamento $r = 2$ com $N = 64 \times 64$ incógnitas, o maior número possível de malhas ($L_{\text{máx}}$) é 6. São elas: $2 \times 2, 4 \times 4, 8 \times 8, 16 \times 16, 32 \times 32$ e 64×64 . Em todos os casos estudados verificou-se que $L_{\text{ótimo}} = L_{\text{máx}} - (0 \text{ ou } 1)$. Assim, em todos os casos foi empregado $L_{\text{máx}}$ para cada problema. A restrição da solução da malha fina para a

malha grossa é a média simples das soluções nos quatro volumes de controle da malha fina que formam o volume de controle da malha grossa. No problema I a restrição dos resíduos é feita desta mesma forma. Já nos problemas II e III é empregada a soma dos resíduos dos quatro volumes de controle da malha fina que formam o volume de controle da malha grossa. Para todos os problemas adotou-se a estimativa inicial com valor nulo para as variáveis dependentes. O critério de parada adotado para as iterações externas é baseado na razão entre a norma l_1 do resíduo em uma dada iteração $\mathbf{r}_{(ite)}$ e a norma l_1 do resíduo da estimativa inicial $\mathbf{r}_{(0)}$ [6].

O *solver* Gauss-Seidel e suas variantes constituem uma importante classe de *solvers* iterativos. Neste trabalho, foram investigados os resultados obtidos com relação ao tempo de CPU, com os *solvers* GS-Lex e GS-RB. No GS-Lex, as incógnitas são atualizadas na ordem lexicográfica, ou seja, de oeste para leste e do sul para o norte, conforme a Figura 2(a). No GS-RB, a malha é organizada conforme a Figura 2(b), onde as incógnitas em vermelho (círculos) e preto (losangos) são atualizadas alternadamente.



Figura 2: Ordenação lexicográfica (a) e ordenação *red-black* (b) em uma malha 2D

Neste trabalho, o operador de prolongação adotado como padrão foi o operador bilinear [3,7,15], aplicado para prolongar as soluções de cada malha para o nível seguinte, no ciclo FMG, e também para prolongar as correções de malha grossa para malha fina no ciclo V. Para o problema III, foram testados outros operadores para interpolar as soluções ao final do ciclo FMG. Os operadores de alta ordem investigados para esta finalidade foram o biquadrático e bicúbico, obtidos de forma análoga ao operador bilinear, ou seja, interpolando-se polinômios quadráticos ou bicúbicos nas direções x e y [7,15]. Também foram testados os operadores de baixa ordem ilustrados na Figura 3: o *upwind* e o *mixed*, implementados conforme [5] e o *licen*, que consiste em uma interpolação linear na direção vertical seguida da atribuição do valor interpolado para seus vizinhos a oeste e a leste na malha fina.

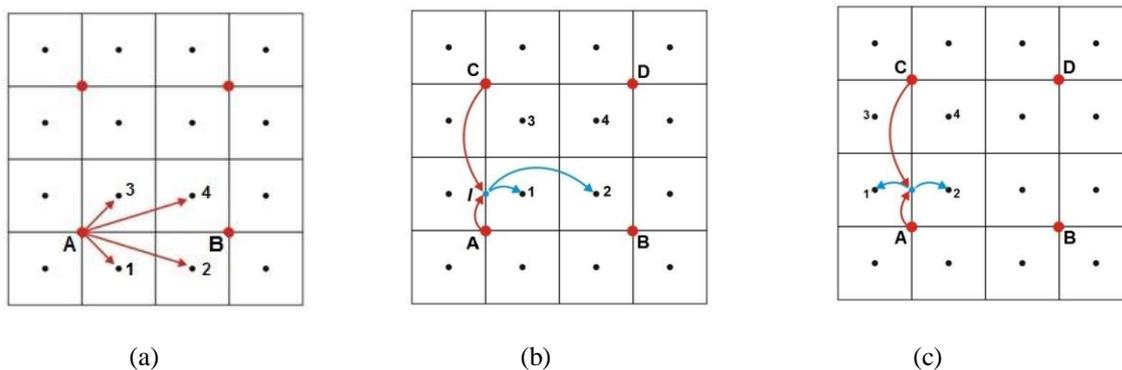


Figura 3: (a) Aplicação do *upwind* para $\phi_A^{2h} > 0$, (b) aplicação do *mixed* para $\phi_I^{2h} > 0$ e (c) aplicação do *licen*

Todos os códigos deste trabalho foram implementados na linguagem FORTRAN/95 com o compilador Compaq Visual Fortran 2005. Todas as simulações foram realizadas no computador CFD-14 do Laboratório de Experimentação Numérica (LENA) da UFPR com processador AMD Athlon

Core 2 Duo com 2.2 GHz e 2 GB RAM, usando aritmética de dupla precisão em sistema operacional Windows XP de 64 bits.

3 Resultados e Conclusão

Com o intuito de avaliar a influência do número de incógnitas no sistema de equações sobre o tempo de CPU, foram analisados os resultados obtidos para as malhas com 16×16 a 2048×2048 volumes de controle, com os métodos FAS-FMG e de malha única, o *singlegrid* (SG), sendo que os resultados deste último são relativos às malhas 16×16 até no máximo 1024×1024 .

Na Figura 4 é possível observar que as curvas do FAS-FMG têm menor inclinação do que as respectivas curvas do SG, o que evidencia a melhora no desempenho do FAS-FMG com o aumento do número de incógnitas. Considerando-se a malha mais fina, o tempo de CPU observado para o problema de Burgers com o FAS-FMG é de aproximadamente 10 e 20 vezes maior que o tempo observado para os problemas de Advecção-Difusão e de Laplace, respectivamente.

Os resultados na Tabela 2 confirmam o excelente desempenho do MG em relação ao algoritmo de malha única, o *singlegrid* (SG). O fator de aceleração (S) é a razão entre o tempo de CPU do SG e o do MG [7].

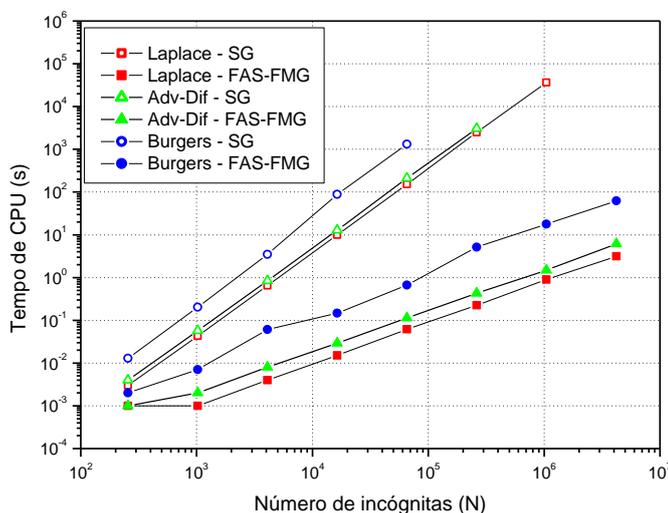


Figura 4: Efeito do número de incógnitas sobre o tempo de CPU

Problema	Malha	----- Tempo de CPU (s) -----			----- S -----	
		SG	FAS	FAS-FMG	FAS	FAS-FMG
Laplace		9,98	0,02	0,02	454	666
Advecção-Difusão	128×128	12,91	0,04	0,03	340	445
Burgers		69,33	0,21	0,15	327	472
Laplace		2478,27	0,38	0,22	6609	11014
Advecção-Difusão	512×512	3049,77	0,70	0,43	4338	7059
Burgers		25281,72	7,63	5,12	3312	4940
Laplace		541355 *	6,19	3,16	87484	171531
Advecção-Difusão	2048×2048	698889 *	10,58	6,12	66070	114104
Burgers		858560 *	128,62	62,62	6675	13709

* valores estimados a partir do ajuste da curva $t_{CPU} = cN^p$

Tabela 2: Fatores de aceleração S do FAS e FAS-FMG em relação ao SG

Tanto na versão FAS quanto na versão FAS-FMG os fatores de aceleração (S) são significativamente altos e melhoram com o refinamento da malha, mas diminuem com o aumento da

complexidade do problema. Conforme esperado, o desempenho do FAS-FMG é aproximadamente duas vezes mais rápido do que o FAS [16].

O desempenho do GS-Lex foi comparado com o do GS-RB. Como pode ser observado na Tabela 3, os resultados do tempo de CPU com o *solver* GS-RB para os problemas de Laplace e Advecção-Difusão são respectivamente 34 e 29% menores do que os resultados com o GS-Lex na malha mais refinada. O problema de Burgers, por sua vez, é apenas 5,7% mais rápido nesta malha.

Resultados semelhantes aos obtidos neste estudo, acerca do comportamento do GS-RB em relação ao GS-Lex foram encontrados em [9,11]. Por outro lado, em [1,8] a redução no tempo de CPU obtida com o emprego do GS-RB foi pequena, em relação ao GS-Lex ou, ainda, o GS-Lex foi superior.

Nas Tabelas 4 e 5 são apresentados os resultados obtidos para o tempo de CPU no problema de Burgers, mediante o emprego dos diferentes operadores de prolongação.

Malha	Laplace		Advecção-Difusão		Burgers	
	GS-Lex	GS-RB	GS-Lex	GS-RB	GS-Lex	GS-RB
128×128	0,015	0,011	0,029	0,018	0,147	0,143
512×512	0,225	0,195	0,432	0,336	5,117	4,771
2048×2048	3,156	2,359	6,172	4,792	62,625	59,234

Tabela 3: Tempo de CPU (s) com os *solvers* GS-Lex e GS-RB – FAS-FMG

Malha	Operador de prolongação da solução no FMG											
	bilinear		licen		Biquadrático		bicúbico		<i>mixed</i>		<i>Upwind</i>	
	CPU	V	CPU	V	CPU	V	CPU	V	CPU	V	CPU	V
512×512	5,117	7	5,117	7	5,117	7	5,125	7	6,242	10	5,898	8
1024×1024	17,750	6	20,453	7	17,734	6	17,750	6	22,703	9	23,609	8
2048×2048	62,625	5	85,344	7	62,625	5	62,703	5	96,312,	9	87,438	7

Tabela 4: Tempo de CPU (s) e número de ciclos V para cada operador com *solver* GS-Lex

Malha	Operador de prolongação da solução no FMG											
	bilinear		licen		Biquadrático		bicúbico		<i>mixed</i>		<i>upwind</i>	
	CPU	V	CPU	V	CPU	V	CPU	V	CPU	V	CPU	V
512×512	4,771	7	4,760	6	4,635	8	4,766	6	6,180	7	6,297	9
1024×1024	16,656	6	19,016	6	16,391	7	19,031	6	21,391	6	25,219	9
2048×2048	59,234	5	79,812	6	69,438	7	67,578	5	89,562	6	94,047	8

Tabela 5: Tempo de CPU (s) e número de ciclos V para cada operador com *solver* GS-RB

De acordo com os resultados das Tabelas 4 e 5, para os dois *solvers* testados, considerando-se o tempo de CPU, o operador bilinear é o que apresenta o melhor desempenho, em geral.

Considerando-se os resultados para o GS-Lex, os operadores de ordem mais alta, biquadrático e bicúbico, tiveram resultados bem próximos àqueles do operador bilinear, tanto no que diz respeito ao tempo de CPU quanto ao número de ciclos V empregados. Os demais operadores de 1ª ordem *licen*, *mixed* e *upwind* tiveram desempenho muito inferior, sendo de 36 a 54% mais lentos do que o operador bilinear.

Nos testes realizados com o GS-RB, observa-se que o operador bilinear ainda é o operador que apresenta o menor tempo de CPU e os operadores de 1ª ordem permanecem com os piores desempenhos. Já os operadores de alta ordem, biquadrático e bicúbico, têm desempenho inferior ao apresentado com o GS-Lex, sendo aproximadamente 16% mais lentos na malha mais refinada.

Agradecimentos

Os autores agradecem o apoio financeiro do CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico), AEB (Agência Espacial Brasileira) através do Programa Uniespaço,

Fundação Araucária (Paraná) e CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior). O segundo autor é bolsista do CNPq.

Referências

- [1] Borba, A. A., "Métodos iterativos e *multigrid*", Dissertação de Mestrado, PosMec – UFSC, 2004.
- [2] Brandt, A., Livne, O., "Multigrid Techniques: 1984 Guide with Applications to Fluid Dynamics", Revised Edition. Philadelphia: Society for Industrial and Applied Mathematics, 2011.
- [3] Briggs, W. L., Henson, V. E., McCormick, S. F., A multigrid tutorial, 2^a ed. Philadelphia: Society for Industrial and Applied Mathematics, 2000.
- [4] Darbandi, M., Vakili, S., Schneider, G., "Efficient multilevel restriction-prolongation expressions for hybrid finite volume element method", International Journal of Computational Fluid Dynamics, v. 22, n. 1, p. 29-38, 2008.
- [5] Drikakis D., Iliev, O. P., Vassileva, D. P., "A nonlinear multigrid method for the three-dimensional incompressible Navier-Stokes Equations". Journal of Computational Physics, v. 146, p. 301-321, 1998.
- [6] Ferziger, J. H., Peric, M., Computational Methods for Fluid Dynamics, 3^a ed. Berlin: Springer-Verlag, 2002.
- [7] Gonçalves, S. F. T., "Estudo de parâmetros do método *multigrid* geométrico para equações 2D em CFD e volumes finitos", Tese de Doutorado, PG-Mec - UFPR, 2013.
- [8] Kouatchou, J., Zhang, J., "Optimal injection operator in multigrid solution of the three dimensional Poisson equation, International", Journal of Computer Mathematics, v. 76, p. 173-190, 2000.
- [9] Larsson, J., Lien, F. S., YEE, E. "Conditional Semicoarsening Multigrid Algorithm for the Poisson Equation on Anisotropic Grids. Journal of Computational Physics, v. 208, p. 368-383, 2005.
- [10] Liu, Z. "Multigrid method with a new interpolation operator", International Journal of Computer Mathematics, v. 88, n. 5, pp.982-993, 2011.
- [11] Oliveira, F., Pinto, M. A. V., Marchi, C. H., Araki, L. K., "Optimized partial semicoarsening multigrid algorithm for heat diffusion problems and anisotropic grids", Applied Mathematical Modelling, v. 36, p. 4665–4676, 2012.
- [12] Schneider, F. A., "Verificação de soluções numéricas em problemas difusivos e advectivos com malhas não-uniformes". 2007, Tese de Doutorado, PPGMNE – UFPR, 2007.
- [13] Shih, T. M., Tan, C. H., Hwang, B. C., "Effects of grid staggering on numerical scheme". International Journal for Numerical Methods in Fluids, v. 9, p. 193-212, 1989.
- [14] Suero, R., Pinto, M. A. V., Marchi, C. H., Araki, L. K., Alves, A. C., "Analysis of algebraic multigrid parameters for two-dimensional steady-state heat diffusion equations", Applied Mathematical Modelling, v. 36 p. 2996–3006, 2012.
- [15] Trottenberg, U., Oosterlee, C., Schüller, A., Multigrid. San Diego: Academic Press, 2001.
- [16] Zhang, W.; Zhang, C. H.; Xi, G., "An explicit Chebyshev pseudospectral multigrid method for incompressible Navier-Stokes equations". Computers & Fluids, v. 39, pp.178-188, 2010.