

Redução do erro de iteração e aceleração do método *Multigrid* com o uso de extrapoladores

Márcio A. M. de Anunciação

Programa de Pós-Graduação em Métodos Numéricos em Engenharia (PPGMNE),
Universidade Federal do Paraná, UFPR
81531-990, Curitiba, PR; E-mail: marcioalexandro@ufpr.br

Marcio A. V. Pinto, Luciano K. Araki, Carlos H. Marchi

Departamento de Engenharia Mecânica, Universidade Federal do Paraná, UFPR, Curitiba, PR
E-mails: marcio_villela@ufpr.br; lucaraki@ufpr.br; marchi@ufpr.br

Márcio A. Martins

Depto. de Matemática, Universidade Estadual do Centro Oeste, UNICENTRO, Guarapuava, PR
E-mail: mandre@unicentro.br

Resumo: Neste trabalho foi resolvido numericamente o problema de condução de calor linear bidimensional, governado pela equação de Poisson, com condições de contorno de Dirichlet, empregando o método *Multigrid* geométrico associado aos seguintes métodos de extrapolação: Aitken, Empírico, Mitin, Épsilon (escalar e topológico), Rho (escalar e topológico) e múltiplas extrapolações de Aitken e Mitin. O objetivo deste trabalho foi a redução do erro de iteração, tempo de CPU e fatores de convergência. De acordo com os resultados, verificou-se a redução da magnitude do erro de iteração, redução do resíduo adimensionalizado com base na estimativa inicial e redução do fator de convergência, em um tempo praticamente equivalente ao da aplicação do método *Multigrid* puro.

Palavras-chave: métodos de extrapolação, *Multigrid*, aceleração de convergência, erro de iteração.

1. Introdução

No processo de discretização de um modelo matemático em problemas de Dinâmica dos Fluidos Computacional (CFD) é comum a obtenção de sistemas de equações algébricas do tipo $Au=f$, onde A é a matriz dos coeficientes, f é o vetor independente e u é o vetor incógnita. A solução desses sistemas, em geral, é feita através de métodos iterativos, que embora mais rápidos que os métodos diretos, geralmente apresentam convergência lenta para problemas de grande porte. A aceleração de convergência de processos iterativos pode se dar de duas formas: modificando o processo iterativo, ou transformando a sequência que converge lentamente em outra, com melhores propriedades de convergência [2].

Nas últimas décadas, uma alternativa que se tem mostrado muito eficiente na aceleração de processos iterativos é o método *Multigrid*. Sua filosofia está baseada no emprego de várias malhas com diferentes graus de refinamento, as quais são percorridas durante o processo iterativo. Outra abordagem que visa acelerar a convergência dos métodos iterativos é associá-los a métodos de extrapolação, cuja finalidade é transformar uma sequência de vetores em uma nova sequência que converge mais rapidamente do que a inicial.

No presente trabalho foram associados métodos de extrapolação ao método *Multigrid*, com o objetivo de acelerar o processo iterativo e reduzir o erro de iteração. Tal formulação mostrou-se atraente, em especial por não se ter na literatura citações sobre tal procedimento.

2. Métodos numéricos

2.1 Método *Multigrid*

O método *Multigrid* se vale das características de suavização do erro por parte dos métodos iterativos clássicos: ela ocorre rapidamente (nas iterações iniciais) para componentes oscilatórias,

enquanto para componentes suaves, há uma grande perda de desempenho, sendo que para um número grande de iterações tais métodos perdem sua eficiência. Assim, o método *Multigrid* trabalha com um esquema de malhas auxiliares mais grosseiras (com menor número de pontos) nas quais as componentes do erro são rapidamente suavizadas, para então retornar-se à malha original. A informação é transferida entre malhas através de operadores, chamados de operadores de restrição (informações de uma malha fina para a grossa seguinte) ou de prolongação (informações da malha grossa para a fina). O operador de restrição utilizado neste trabalho foi o operador de ponderação completa e o operador de prolongação foi a interpolação bilinear [3]. Neste trabalho utilizou-se ainda o ciclo V, por ser bastante referenciado e computacionalmente eficiente [3]. Além disso, como o problema resolvido é linear, utilizou-se o Esquema de Correção (CS – *Correction Scheme*) que transfere apenas o resíduo para as malhas mais grossas [3].

2.2 Métodos de extrapolação

Os métodos de extrapolação têm por finalidade transformar uma sequência que converge lentamente em outra, com melhores propriedades de convergência. Tais extrapoladores podem ser classificados em escalares e vetoriais, de acordo com a forma com que suas informações são manipuladas. Os extrapoladores (ou métodos de extrapolação) escalares utilizados nesse trabalho são: Aitken [1] e Empírico [7], que aceleram a convergência de sequências que convergem linearmente; Mitin [8], que é recomendado para sequências oscilatórias; Épsilon escalar [1], recomendado na literatura como o melhor método para fins de aceleração de sequências que convergem lentamente [5,6]; e Rho escalar [1], que não acelera sequências com convergência linear, mas é recomendado para sequências logaritmicamente convergentes [5, 6]. Os métodos de extrapolação Épsilon e Rho podem ser generalizados para o caso vetorial, podendo ser aplicados de forma recursiva chamada formulação topológica [1] e que foi utilizada neste trabalho. Além dos extrapoladores citados, extrapolações múltiplas (extrapolação de dados já extrapolados) também foram realizadas com os extrapoladores Aitken e Mitin. Mais detalhes sobre métodos de extrapolação podem ser encontrados em [1].

2.3 Metodologia

Foram estudados nove casos, resultantes da combinação dos valores relativos ao dimensionamento da malha mais fina ($N = 129 \times 129$, $N = 1025 \times 1025$ e $N = 4097 \times 4097$) e ao critério de parada (10^{-6} , 10^{-10} e 10^{-15}). As extrapolações se deram em dois momentos: ao final do processo iterativo e durante o processo iterativo.

O objetivo deste estudo é mostrar como o uso de extrapoladores associados ao método *Multigrid* pode reduzir o erro de iteração e acelerar o processo iterativo. Para isso, escolheu-se um problema simples, com uma geometria simples e solução analítica conhecida a fim de analisar especificamente os efeitos desta metodologia. Assim, neste trabalho, resolveu-se o problema de condução de calor linear bidimensional em regime permanente descrito pela equação de Poisson, em um domínio quadrado de lado unitário, com termo fonte e solução analítica obtida pelo método de soluções fabricadas e dadas em [9].

O modelo numérico foi obtido discretizando-se a equação de Poisson com o método das diferenças finitas (MDF) [10], com aproximações por diferença central (CDS – *Central Difference Scheme*) em malhas quadradas uniformes.

3. Resultados

3.1 Uso de extrapoladores no final do *Multigrid*

Para esta metodologia, cada caso foi resolvido de três maneiras distintas:

- usando apenas o método *Multigrid* até atingir o critério de parada (MG);
- usando o método *Multigrid* com um ciclo V além daqueles necessários para se atingir o critério (MG + 1 ITE); e

(c) usando o método *Multigrid* até atingir o critério de parada e extrapolando as soluções obtidas nas últimas iterações (MG + Extrapolador).

Os parâmetros analisados foram: tempo de CPU (t_{CPU}) em segundos (s), pico de memória de armazenamento $M(MB)$, norma adimensionalizada do resíduo com base na estimativa inicial ($\|R\|_2$), fator de convergência empírico $q^{(k)}$ [11], fator de convergência média empírico $\hat{q}^{(k)}$ [11], norma infinito do erro de iteração ($\|E_n\|_\infty$) e norma euclidiana do erro de iteração ($\|E_n\|_2$). A Tabela 1 apresenta os resultados para $N = 1025 \times 1025$ e a tolerância 10^{-15} .

Metodologia	t_{CPU} (s)	$M(MB)$	$\ R\ _2$	$q^{(k)}$	$\hat{q}^{(k)}$	$\ E_n\ _\infty$	$\ E_n\ _2$
MG	21,871	119,640	6,272E-16	4,238E-02	4,149E-02	3,672E-17	1,816E-17
MG + 1 ITE	23,775	119,636	2,663E-17	4,246E-02	4,157E-02	1,553E-18	7,749E-19
MG + Aitken	119,076	218,468	3,145E-14	5,014E+01	7,495E-02	4,118E-18	7,301E-20
MG + Empírico	119,076	218,468	3,145E-14	5,014E+01	7,495E-02	4,118E-18	7,301E-20
MG + Mitin	119,029	251,360	8,701E-14	1,387E+02	8,158E-02	9,630E-18	3,349E-19
MG + Épsilon escalar	29,874	251,556	3,044E-11	4,854E+04	1,329E-01	5,152E-15	7,111E-18
MG + Rho escalar	29,921	251,564	4,511E-12	7,193E+03	1,133E-01	2,066E-14	1,000E-14
MG + Épsilon topológico	121,525	399,444	1,648E-19	2,628E-04	2,721E-02	1,213E-21	5,165E-22
MG + Rho topológico	31,169	399,656	3,498E-13	5,577E+02	9,161E-02	2,066E-14	9,998E-15
MG + Múlt. ext. Aitken	29,952	251,576	4,716E-12	7,519E+03	1,137E-01	7,045E-16	1,856E-18
MG + Múlt. ext. Mitin	30,405	317,368	4,702E-10	7,497E+05	1,669E-01	6,402E-14	1,098E-16

Tabela 1: Resultados dos parâmetros estudados (Extrapolações ao final do *Multigrid*)

Percebe-se, para este caso, que o uso do extrapolador Épsilon topológico foi a metodologia que apresentou os melhores resultados para a norma do resíduo, para os fatores de convergência e para as normas do erro de iteração. Os resultados para os outros casos (outros valores de N) foram análogos. Desta forma, conclui-se que o extrapolador que se mostrou mais eficiente na maioria dos parâmetros foi o Épsilon topológico. Em relação ao tempo de CPU e a memória, os valores para esses dois parâmetros mostraram-se maiores para todos os problemas resolvidos com algum extrapolador. Isso ocorre porque os vetores com as soluções usadas nos cálculos das extrapolações precisam ser armazenados e os cálculos das extrapolações demandam um tempo excedente em comparação com o simples uso do método *Multigrid*.

3.2 Uso de extrapoladores durante o *Multigrid*

Nesta abordagem utilizou-se o extrapolador Épsilon topológico durante os ciclos do *Multigrid*, ou seja, a cada cinco soluções, estas são combinadas com o extrapolador e geram uma nova solução. Esta solução serve de estimativa inicial para o próximo ciclo *Multigrid* e o processo iterativo prossegue. Ao se obter outras cinco soluções, uma nova extrapolação é realizada e assim por diante, até se atingir o critério de parada estabelecido.

A Tabela 2 apresenta os resultados para $N = 4097 \times 4097$ e tolerância 10^{-15} . Nesta tabela apenas as metodologias *Multigrid* (MG) e extrapolador Épsilon topológico durante o *Multigrid* (MG + Épsilon topológico) são comparadas.

Metodologia	t_{CPU} (s)	$M(MB)$	$\ R\ _2$	$q^{(k)}$	$\hat{q}^{(k)}$	$\ E_n\ _\infty$	$\ E_n\ _2$
MG	351,579	1,844	6,270E-16	4,238E-02	4,148E-02	3,673E-17	1,818E-17
MG + Épsilon topológico	364,699	3,946	2,320E-16	2,417E-02	1,831E-02	1,912E-18	8,251E-19

Tabela 2: Resultados dos parâmetros estudados (Extrapolações durante o *Multigrid*)

Analisando este caso, pode-se perceber que o tempo de CPU do uso do extrapolador foi levemente maior quando comparado ao uso do *Multigrid* puro. Quanto à norma adimensionalizada do resíduo, o uso do extrapolador mostrou-se mais eficiente do que o *Multigrid* puro. O mesmo pode ser percebido para os fatores de convergência. Quanto ao erro de iteração, houve uma redução de sua magnitude em relação às duas normas, e esta redução chegou a aproximadamente 95%, enquanto o tempo de CPU teve um aumento de menos de 4%.

Buscando resultados mais precisos em relação ao tempo de CPU, foram realizados testes adicionais, com N variando de 33×33 a 8193×8193 e tolerância de 10^{-20} . Nestes testes, pode-se perceber que o uso do extrapolador mostra-se mais vantajoso com o aumento o tamanho do problema, pois reduz o tempo de CPU em relação ao uso do *Multigrid*. Para confirmar tal melhora, foi calculado o *speed-up* da metodologia MG em relação à metodologia com extrapolador. O *speed-up* (S_p) é definido como a razão entre os tempos de CPU de dois algoritmos, no caso, o *Multigrid puro* e o com extrapolador [4]. Os resultados aparecem na Tabela 3.

N	33x33	65x65	129x129	257x257	513x513	1025x1025	2049x2049	4097x4097	8193x8193
S_p	0,966	0,929	0,908	0,930	0,938	1,013	1,011	1,011	1,011

Tabela 3: *Speed-up* de MG em relação ao MG + Épsilon topológico para $\epsilon = 10^{-20}$

Com base na Tabela 3, percebe-se que a partir de $N = 1025 \times 1025$, o valor do *speed-up* passa a ser maior que 1, indicando que o uso do extrapolador passa a ser mais rápido do que o *Multigrid* puro.

Foi feito ainda um ajuste de curvas por mínimos quadrados para $t_{CPU}(N) = cN^p$, onde p descreve o grau de complexidade do algoritmo. Estes resultados são apresentados na Tabela 4.

Metodologia	c	p
MG	0,0755	1,03258
MG + Épsilon topológico	0,08506	1,02957

Tabela 4: Valores de p e c para os algoritmos das metodologias em estudo

De acordo com os dados da Tabela 4, pode-se afirmar que as duas metodologias são praticamente equivalentes, sendo que o uso do extrapolador Épsilon topológico acelera o método *Multigrid* com o aumento do número de incógnitas do problema.

Pode-se perceber pela Figura 1 que os valores de memória aumentam com o uso dos extrapoladores, mas o aumento percentual tem caráter assintótico, o que é um bom sinal, pois se busca resolver problemas em malhas bem refinadas.

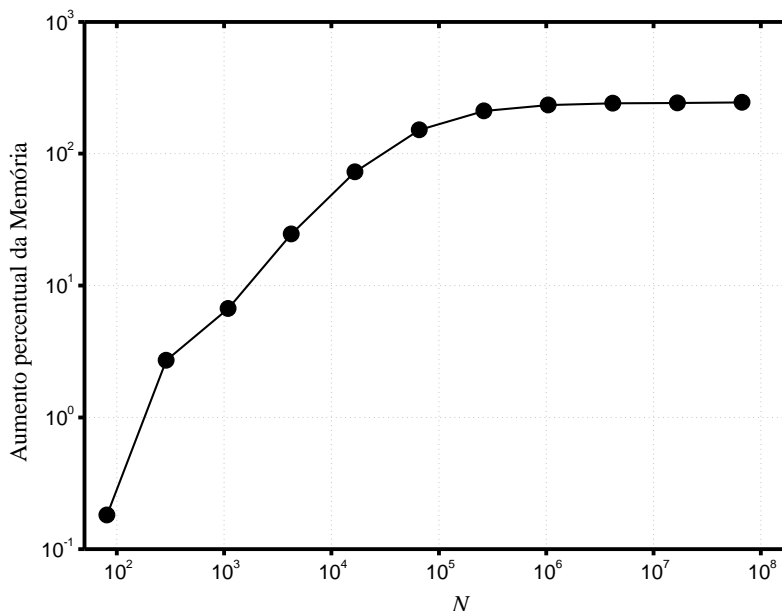


Figura 1 – Aumento percentual da memória em função do número de variáveis do problema

4. Conclusão

Com base nos resultados obtidos neste trabalho, verificou-se que:

- 1) O uso dos extrapoladores ao final do *Multigrid*, para o problema estudado, fez o tempo de CPU aumentar em relação a um ciclo adicional do *Multigrid*.
- 2) A memória de armazenamento é maior quando se usa um extrapolador.

- 3) O extrapolador Épsilon topológico usado tanto ao final, quanto durante o *Multigrid* foi a metodologia que mais reduziu a norma adimensionalizada do resíduo, os fatores de convergência e a magnitude do erro de iteração.
- 4) Os tempos de CPU das extrapolações realizadas durante o *Multigrid* são praticamente iguais aos obtidos apenas com *Multigrid*, sendo que a metodologia com extrapolador passa a ser levemente mais rápida com o aumento do número de incógnitas do problema..

Agradecimentos

Os autores agradecem o apoio financeiro do CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico), AEB (Agência Espacial Brasileira) através do Programa Uniespaço e CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior). O quarto autor é bolsista do CNPq.

Referências

- [1] C. Brezinski, R. M. Zaglia, *Extrapolation Methods - Theory and Practice*. 2 ed., Amsterdam: Elsevier Science Publishers, 2002.
- [2] C. Brezinski, R. M. Zaglia, A Review of Vector Convergence Acceleration Methods, With Applications to Linear Algebra Problems, *International Journal of Quantum Chemistry*, vol. 109, pp. 1631–1639, 2008.
- [3] W. L. Briggs, V. E. Henson, S. F. McCormick, *A Multigrid Tutorial*, 2 ed. Philadelphia: SIAM, 2000.
- [4] G. Galante, *Métodos Multigrid Paralelos em Malhas Não-Estruturadas Aplicados à Simulação de Problemas de Dinâmica de Fluidos Computacional e Transferência de Calor*. Dissertação (Mestrado). Universidade Federal do Rio Grande do Sul, Porto Alegre, RS, 2006.
- [5] Q. Gao, Z. Jiang, T. Liao, K. Song, Application of the vector ε and ρ extrapolation methods in the acceleration of the Richardson–Lucy algorithm, *Optics Communications*, vol. 283, pp. 4224–4229, 2010.
- [6] P. R. Graves-Morris, D. E. Roberts, A. Salameh, The epsilon algorithm and related topics, *Journal of Computational and Applied Mathematics*, vol. 122, pp. 51-80, 2000.
- [7] M. A. Martins, C. H. Marchi, A. F. C. Silva, Estimate of iteration errors in Computational Fluid Dynamics, *Num. Heat Transfer, Part. B*, vol. 53, pp. 234-245, (2008).
- [8] A. V. Mitin, Linear extrapolation in an iterative method for solving systems of equations, *U.S.S.R. Comput. Maths. Math. Phys.*, vol.25, n.2, pp.1-6, 1985.
- [9] F. Oliveira, M. A. V. Pinto, C. H. Marchi, L. K. Araki, Optimized partial semicoarsening multigrid algorithm for heat diffusion problems and anisotropic grids. *Applied Mathematical Modelling*, vol.36, pp. 4665-4676, 2012
- [10] J. C. Tannehill, D. A. Anderson, R. H. Pletcher, *Computational Fluid Mechanics and Heat Transfer*, Washington: Taylor & Francis, 1997.
- [11] U. Trottenberg,; C. Oosterlee,; A. Schüller, *Multigrid*, St Augustin, Germany: Academic Press, 2001.