

Proceeding Series of the Brazilian Society of Computational and Applied Mathematics

Precondicionador *multigrid* algébrico para métodos iterativos não estacionários na solução de sistemas lineares de grande porte

Henrique Gomes de Jesus¹
 Marcelo Torres Pereira Carrion²
 Maria Claudia Silva Boeres³
 Lucia Catabriga⁴
 Universidade Federal do Espírito Santo

Resumo. Este trabalho se destina à análise empírica do método *multigrid* algébrico como preconditionador do método do Resíduo Mínimo Generalizado (GMRES). Os algoritmos são executados em um conjunto de matrizes do repositório de matrizes esparsas *SuiteSparse Matrix Collection*⁵, relacionadas a aplicações variadas e armazenadas no formato *Compressed Sparse Row* (CSR). Os resultados numéricos são comparados aos do preconditionador Gauss-Seidel e aos do método GMRES sem preconditionador, mostrando que o uso do *multigrid* algébrico como preconditionador permite a convergência para um número maior de matrizes em tempo computacional expressivamente menor.

Palavras-chave. Métodos *multigrid*. *Multigrid* algébrico. Métodos iterativos. Precondicionadores.

1 Introdução

Uma das desvantagens dos métodos iterativos em relação aos diretos é a possibilidade de não convergência, i.e., menor robustez. Porém, métodos iterativos não estacionários podem ter tanto a eficiência quanto a robustez melhoradas com o uso de preconditionamento. Neste trabalho, utilizamos o método do resíduo mínimo generalizado (GMRES, do inglês): método de solução de sistemas lineares assimétricos que aproxima a solução através do vetor de resíduo mínimo em um subespaço de Krylov.

O método GMRES executa, a cada iteração i , 3 operações principais: i produtos escalares, i combinações lineares e 1 produto matriz-vetor. O preconditionamento acelera a convergência do método transformando o sistema:

$$Au = f \tag{1}$$

¹henrique.g.jesus@aluno.ufes.br

²mcarrion@inf.ufes.br

³boeres@inf.ufes.br

⁴luciac@inf.ufes.br

⁵<https://sparse.tamu.edu/>

no equivalente, representado por

$$M^{-1}Au = M^{-1}f \quad (2)$$

Este novo sistema possui a mesma solução, porém, é melhor condicionado. Neste sistema, a matriz M deve ser suficientemente simples de se construir e semelhante à original A . Deste modo, a ação do preconditionador no método iterativo ocorre no produto matriz vetor

$$v = M^{-1}Az \quad (3)$$

sendo calculado em dois passos. No primeiro passo o produto matriz-vetor original é calculado ($w = Az$) e no segundo passo o sistema linear trivial $Mv = w$ é resolvido por algum método de baixa complexibilidade.

A seguir apresentamos os preconditionadores implementados: preconditionador Gauss-Seidel [8] e o preconditionador *multigrid* algébrico [10].

2 Precondicionador Gauss-Seidel

Os métodos iterativos estacionários são baseados na fatoração simples da matriz dos coeficientes definida por:

$$A = L + D + U \quad (4)$$

onde D é a diagonal de A e L e U representam, respectivamente, as partes estritamente inferior e estritamente superior de A . O preconditionador Gauss-Seidel é definido por:

$$M = (L + D)D^{-1}(D + U). \quad (5)$$

Uma vantagem do preconditionador Gauss-Seidel é que não há necessidade de armazenamento extra, pois os coeficientes de M são os próprios coeficientes da matriz A .

3 Precondicionador *multigrid* algébrico

Métodos de relaxação clássicos sofrem com convergência lenta, devido à incapacidade de eliminar as componentes de baixa frequência do vetor erro [1]. Sua eficácia, porém, pode ser significativamente aumentada empregando a estratégia *multigrid*, que corrige este problema solucionando o sistema em discretizações sucessivamente mais grossas, onde a oscilação do erro é aumentada. Após iterar no domínio grosseiro, a solução do sistema é interpolada de volta ao domínio original. Realizando-se este processo recursivamente, temos o esquema chamado de ciclo-V, ilustrado na Fig. 1.

Os métodos *multigrid*, normalmente, podem ser divididos em dois tipos: Geométrico (GMG, do inglês), que se baseia no pressuposto de que o sistema é oriundo de uma discretização em malha, o que restringe o escopo de aplicação do *multigrid*; e o algébrico (AMG, do inglês), que é uma generalização do GMG onde o conjunto de variáveis das malhas grossas é definido de forma exclusivamente algébrica através da força das conexões entre as variáveis [4, 9]. Atualmente, o AMG é considerado um dos métodos mais rápidos para

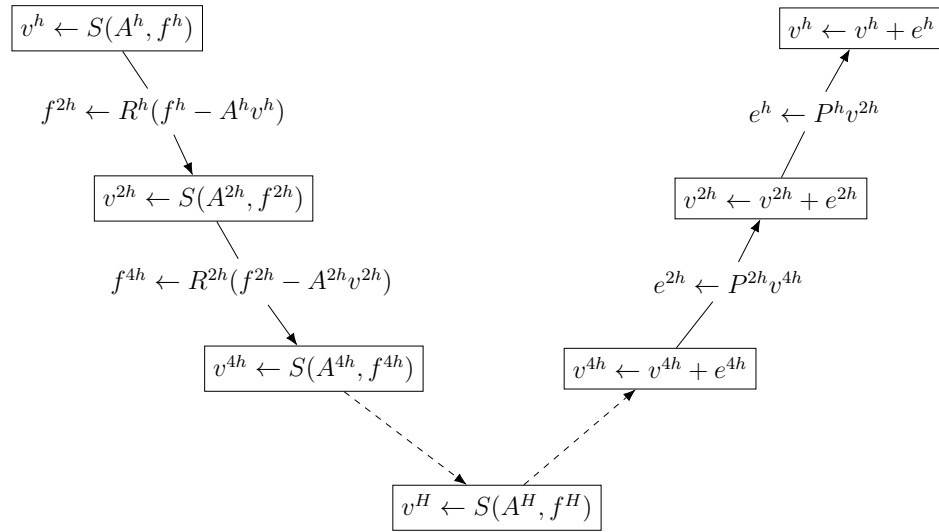


Figura 1: Ciclo-V do método *multigrid*.

várias classes de equações [8]. As técnicas *multigrid* tornaram-se bastante populares porque, em teoria, têm complexidade computacional linear, além de excelente escalabilidade e possibilidade de paralelização. Como consequência, têm sido empregadas em sistemas de grande porte [7].

A construção da versão grosseira, representada por

$$A^{2h} u^{2h} = f^{2h} \tag{6}$$

do sistema original, representado por

$$A^h u^h = f^h \tag{7}$$

onde o valor sobrescrito após cada variável indica o índice do domínio ao qual pertence na hierarquia de discretizações é obtida através de uma operação de Galerkin [5], definida por

$$A^{2h} = R^h A^h P^h \tag{8}$$

onde \mathbf{P} e \mathbf{R} são os operadores de interpolação e restrição, respectivamente, sendo que

$$R^h = (P^h)^T. \tag{9}$$

Existem dois componentes do AMG necessários para a construção da malha grosseira:

- Um esquema de restrição, que determina o conjunto \mathbf{C} de variáveis que fará parte do domínio grosseiro e seu complemento \mathbf{F} em relação ao conjunto de variáveis do domínio original. Este esquema também é chamado de divisão \mathbf{C}/\mathbf{F} ;
- Um operador de interpolação, para fazer o mapeamento e gerar a nova matriz A^{2h} .

No processo de restrição, apenas conexões suficientemente fortes devem ser consideradas. Uma conexão forte é definida por

$$-a_{ij} \geq \epsilon_{str} \max_{k \neq i} (-a_{ik}) \tag{10}$$

onde $0 < \epsilon_{str} < 1$ é o limite de força [10].

Ao utilizar o AMG como preconditionador, ao invés de uma única matriz de preconditionamento M , utilizam-se as matrizes mais grossas produzidas pelo próprio *setup* do AMG (A^h, A^{2h}, \dots, A^H). Estes novos sistemas lineares são solucionados através de uma relaxação clássica – neste trabalho utilizamos a relaxação baseada no método iterativo SOR. Assim, o sistema trivial $Mv = w$ é solucionado por meio um ciclo-V do AMG [6].

4 Resultados Experimentais e Conclusões

Os algoritmos foram implementados em C, compilados com *gcc* versão 5.4.0 e executados em uma máquina com Intel® Xeon® E5410 2.33GHz \times 4 64 bits com 32 GB de memória RAM e Ubuntu 16.04.10. Foram utilizadas matrizes do repositório de matrizes esparsas *SuiteSparse Matrix Collection*⁶, relacionadas a aplicações diversas.

Tabela 1: Ordem e número de elementos não nulos das matrizes A^0 e A^1 .

Matriz	A^0		A^1		Redução %
	Ordem	Não Nulos	Ordem	Não Nulos	
rail_5177	5.177	35.185	1.821	24.631	64.82%
aft01	8.205	125.567	1.108	29.670	86.49%
FEM_3D_thermal1	17.880	430.740	4.737	288.833	73.50%
Dubcova2	65.025	1.030.225	20.190	1.815.606	68.95%
H2O	67.024	2.216.736	33.512	6.797.532	50.00%
FEM_3D_thermal2	147.900	3.489.300	35.359	2.503.795	76.09%
parabolic_fem	525.825	3.674.625	262.913	3.928.323	50.00%
atmosmodd	1.270.432	8.814.880	635.216	16.534.896	50.00%
atmosmodj	1.270.432	8.814.880	635.216	16.534.896	50.00%
Serena	1.391.349	64.131.971	288.874	86.065.994	79.23%
Geo_1438	1.437.960	60.236.322	308.563	84.269.143	78.54%
atmosmodl	1.489.752	10.319.760	744.876	13.828.000	50.00%
af_shell10	1.508.065	26.883.975	585.351	38.315.661	61.18%
G3_circuit	1.585.478	7.660.826	742.720	7.191.332	53.15%
Transport	1.602.111	23.487.281	775.218	30.642.524	51.61%

O método otimizado de armazenamento considerado foi o formato *Compressed sparse row* (CSR). Quanto aos parâmetros, a tolerância utilizada para o resíduo foi de 10^{-8} , o

⁶<https://sparse.tamu.edu/>

limite de iterações do método GMRES foi 1000, o número de vetores da base de Krylov para a reinicialização foi 20, o parâmetro de relaxação do método SOR foi $\omega = 1.5$, a profundidade do ciclo-V foi 1, o valor utilizado para ϵ_{str} foi 0.25 e o número de relaxações aplicadas em cada nível do *multigrid* foi 2. A Tab. 1 apresenta a ordem e o número de elementos não nulos de cada matriz nos níveis A^0 e A^1 , além da porcentagem de redução do ciclo-V com apenas um nível de descida. Observa-se pelos resultados que, para todas as matrizes a redução em um único nível foi igual ou superior a 50%.

Tabela 2: Número de iterações e tempo em segundos do método.

Matriz	GMRES		GMRES+Seidel		GMRES+AMG	
	Iter.	t (s)	Iter.	t (s)	Iter.	t (s)
rail_5177	†	†	†	†	174	0,32
aft01	†	†	†	†	753	3,13
FEM_3D_thermal1	269	0,63	249	1,83	12	0,23
Dubcova2	†	†	817	18,97	48	4,51
H2O	†	†	427	19,23	26	6,01
FEM_3D_thermal2	759	21,33	392	31,00	11	2,39
parabolic_fem	†	†	†	†	585	264,69
atmosmodd	†	†	222	73,44	44	43,81
atmosmodj	†	†	340	113,05	43	44,73
Serena	†	†	†	†	139	612,89
Geo_1438	†	†	†	†	197	838,55
af_shell10	445	110,61	111	43,12	23	26,04
af_shell10	109	33,18	49	31,46	7	16,47
G3_circuit	†	†	†	†	326	284,71
Transport	†	†	†	†	561	1147,0

A Tab. 2 apresenta o número de iterações (Iter) e o tempo (t(s)), em segundos, utilizados por cada método – GMRES sem preconditionador, GMRES com preconditionador Gauss-Seidel e GMRES com preconditionador AMG. O símbolo † indica a não convergência do método. O método GMRES com o preconditionador AMG foi o único que convergiu para todas as matrizes, tendo obtido o menor número de iterações e menor tempo de CPU. Para todas as matrizes que convergiram com o preconditionador Gauss-Seidel, o número de iterações foi menor que o método sem preconditionador, mas o tempo de CPU foi maior para as matrizes fem_3d_thermal1 e fem_3d_thermal2, enquanto que para o preconditionador AMG o tempo de processamento também foi menor. Como o *setup* do AMG é muito caro computacionalmente, tal vantagem ocorreu pois o número de iterações diminuiu drasticamente.

A Fig. 2 apresenta as medidas do tempo de processamento para:

- **Setup:** Processo de geração da matriz A^1 ;
- **Solver:** Método GMRES, exceto pelo produto matriz-vetor, que foi computado em

Precond. e Relax;

- **Precond.:** Ciclo-V, exceto pelo tempo de relaxação, que foi computado em **Relax**, ou seja, engloba o tempo dos produtos matriz-vetor responsáveis por interpolar os vetores solução entre a hierarquia do Ciclo-V;
- **Relax:** SOR operado pelo AMG em cada nível. O número de relaxações necessárias para executar um ciclo-V é sempre igual a 8, uma vez que em cada nível são executadas 2 relaxações e no nível mais baixo são executadas 4.

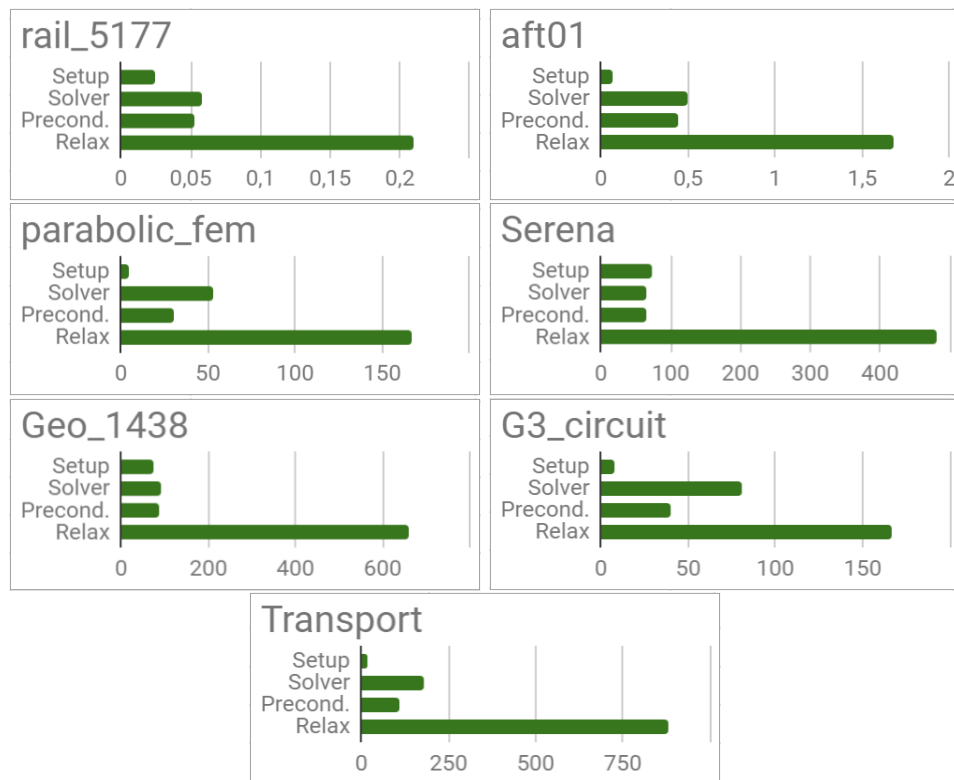


Figura 2: Medida dos tempos de processamento, por componente do método iterativo, para cada matriz de difícil convergência.

Os gráficos apresentam resultados somente para as matrizes de convergência difícil, isto é, aquelas para as quais só houve convergência utilizando o preconditionador AMG. É possível perceber que o tempo de **Setup** não aumentou em relação ao tempo de **Relax** para as matrizes maiores. Como o cálculo da matriz reduzida envolve a definição das matrizes de restrição e interpolação, além de produtos matriz-matriz (Eq.(8)), o algoritmo desenvolvido é escalável, e portanto, adequado para sistemas lineares de grande porte.

Matrizes de ordem acima de 1.000.000 foram solucionadas em menos de um minuto. Estes resultados mostram que, muito embora o AMG tenha a desvantagem de necessitar de

um processo de *setup* consideravelmente oneroso, seu uso pode ser muito recompensador em matrizes de difícil convergência.

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Referências

- [1] W. L. Briggs, V. E. Henson e S. F. McCormick. *A Multigrid Tutorial*. SIAM, 2000.
- [2] M. T. P. Carrion, An Experimental Study of the Geometric and Algebraic Multigrid Strategies, Dissertação de Mestrado, UFES, 2016.
- [3] M. T. P. Carrion, M. C. S. Boeres e L. Catrabriga, An Experimental Study of the Geometric and Algebraic Multigrid Strategies, *XXXVIII IberianLatin American Congress on Computational Methods in Engineering*, ABMEC, 2017.
- [4] R. D. Falgout, An Introduction to Algebraic Multigrid, *Computing Science and Engineering*, 24–33, 2006.
- [5] P. W. Hemker, A note on defect correction processes with an approximate inverse of deficient rank, *Journal of Computational and Applied Mathematics*, pages 137–139, 1982.
- [6] A. McAdams, A. Sifakis e E. Teran, A parallel multigrid Poisson solver for fluids simulation on large grids, *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, pages 65-74, 2010.
- [7] J. Park, M. Smelyanskiy, U. M. Yang, D. Mudigere e P. Dubey, High-performance algebraic multigrid solver optimized for multi-core based distributed parallel systems., *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, ACM, No. 54, 2015.
- [8] Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, Minneapolis, 2003.
- [9] K. Stuben, Algebraic multigrid (AMG): an introduction with applications, *Academic Press*, New York, 2000.
- [10] U. M. Yang, Parallel Algebraic Multigrid Methods - High Performance Preconditioners, *Springer Berlin Heidelberg*, Berlin, Heidelberg, pages 209–236, 2006.