Available online at www.sciencedirect.com
ScienceDirect
Applied Mathematics and Computation 179 (2006) 344–351
APPLIED MATHEMATICS AND COMPUTATION
www.elsevier.com/locate/amc

A fast algebraic multigrid preconditioned conjugate gradient solver

Fábio Henrique Pereira, Sérgio Luís Lopes Verardi, Silvio Ikuyo Nabeta *

LMAG-PEA-EPUSP, Av. Prof. Luciano Gualberto, Travessa 3, n. 158, 05508-900 São Paulo/SP, Brazil

Abstract

This work presents a new approach for selecting the coarse grids allowing a faster algebraic multigrid (AMG) preconditioned conjugate gradient solver. This approach is based on an appropriate choice of the parameter α considering the matrix density during the coarsening process which implies in a significant reduction in the matrix dimension at all AMG levels.

Keywords: Algebraic multigrid; Coarsening process; Strength threshold; Preconditioner; Iterative methods; Linear systems; Sparse matrices

1. Introduction

The multigrid method (MG) is a well-established numerical technique for solving linear systems. Several works have explored the use of MG as a preconditioner for the conjugate gradient method (CG) [1–3]. In [2] the MG is used with the CG in the resolution of the two-dimensional Poisson equation in a regular domain, showing the superiority of this method in relation to the incomplete Cholesky conjugate gradient method (ICCG).

In contrast to MG, where a mesh hierarchy is explicitly required, AMG constructs the matrix hierarchy and transfer operators just using information from the original matrix. Hence, when the problem involves an irregular domain and unstructured meshes or when one is interested on black-box solvers the AMG is well suited as a preconditioner for iterative solvers.

The standard AMG normally presents a high computational cost, thus it is not generally used in small and medium size problems [4]. In [1] a modification in the conventional AMG is proposed to reduce its construction and solving time. In that work the original matrix is approximated to a symmetric M-matrix. However, when the original matrix is very different of a M-matrix that approach can increase prohibitively the convergence factor.

* Corresponding author.
E-mail address: nabeta@pea.usp.br (S.I. Nabeta).

0096-3003/$ - see front matter
doi:10.1016/j.amc.2005.11.115

In this work, we present a new approach to select the coarse grids that allows the use of the AMG as a preconditioner for the CG method for small and medium problems. This approach leads to a significant reduction of the matrix dimension at all levels without losing the robustness.

The paper is organized as follows. In Section 2 we give a brief overview of the basic AMG algorithm, detailing the coarsening process in Section 3. In Sections 4 and 5, the test cases and the numerical results are presented, comparing the performance of the AMG with that of incomplete Cholesky method in the preconditioning of the ill-conditioned matrices. Finally, some conclusions are formulated in Section 6.

2. The algebraic multigrid

In this section we give an outline of the basic principles of AMG, and define some terminology and notation. Detailed explanations may be found in [3]. Consider a problem of the form

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

where A is an $n \times n$ matrix with entries a_{ij}. For AMG, a ''grid'' is simply a set of indices of the variables, so the original grid is denoted by $\omega_k = \{1, 2, \ldots, n\}$ with $k = 1$.

In any multigrid method, the central idea is that error e, that is not eliminated by relaxation, must be removed by coarse-grid correction. This is done by solving the residual equation $Ae = r$ on a coarser grid, then interpolating the error back to fine grid and using it to correct the fine-grid approximation by $u \leftarrow u + e$. The coarse-grid problem itself is solved by a recursive application of this method [7–11].

The AMG explores the use of many levels to eliminate the error components that are not efficiently removed by relaxation [2,3]. In this way, the coarsest grid problem becomes sufficiently small to be solved by a direct solver.

Using subscripts to indicate level number, where M denotes the number of levels and 1 is the finest level so that $A_1 = A$, the AMG algorithm consists of the following components [3,5]:

- *Coarsening*: define the splitting $\omega_k = \omega_{\mathrm{C}}^k \cup \omega_{\mathrm{F}}^k$ of ω_k into sets of coarse and fine grid nodes ω_{C}^k and ω_{F}^k, respectively.
- *Transfer operators*:

$$\begin{aligned} &\text{prolongation} \quad P_k : V_{k+1} \to V_k, \\ &\text{restriction} \quad R_k = P_k^{\mathrm{T}}. \end{aligned} \tag{2}$$

- *Definition of the matrix hierarchy*:

$$A_{k+1} = R_k A_k P_k. \tag{3}$$

- *Appropriate smoother*: basic iterative method.

The most important issue to be discussed is the construction of the matrix hierarchy and the prolongation operator, i.e., the setup phase [3,4]. This phase is explained as follows:

AMG setup phase:

1. Set $k = 1$ and $A_1 = A$.
2. Partition ω_k into disjoint sets ω_{C}^k and ω_{F}^k
   (a) Set $\omega_{k+1} = \omega_{\mathrm{C}}^k$.
   (b) Define interpolation P_k.
3. Set $R_k = P_k^{\mathrm{T}}$ and $A_{k+1} = R_k A_k P_k$.
4. If ω_{k+1} is small enough, set $M = k + 1$ and *stop*. Otherwise, set $k = k + 1$ and go to step 2.

Step 2 is the core of the AMG setup process and is detailed in the following section. Once the setup phase is completed and all these components are defined, the recursively defined multigrid $V(\mu_1, \mu_2)$-cycle, which is used here as a preconditioner for the CG method, can be performed as follows:

Algorithm. $MGV_k(u_k,f_k)$. The $V(\mu_1,\mu_2)$-cycle.

If $k = M$, solve $A_M u_M = f_M$ with a direct solver.
Otherwise:
  Apply the smoother μ_1 times to $A_k u_k = f_k$.
  Perform coarse-grid correction:
    Set $e_{k+1} = 0$ and $r_k = f_k - A_k u_k$.
    Restrict $r_{k+1} = R_k r_k$.
    Apply $MGV_{k+1}(e_{k+1}, r_{k+1})$.
      Prolong $e_k = P_k e_{k+1}$.
      Correct the solution by $u_k \leftarrow u_k + e_k$.
  Apply the smoother μ_2 times to $A_k u_k = f_k$.

3. The coarsening process

The AMG efficiency is improved by reducing the number of nonzero entries of the coarse matrices by using the following sets of connections between grid points [1,3–5,7]:

$$N_i = \{j : |a_{ij}| \neq 0, i \neq j\},$$
$$S_i = \left\{ j \in N_i : |a_{ij}| \geqslant \alpha \max_{k \neq i} |a_{ik}| \right\},$$
$$S_i^{\mathrm{T}} = \{j : i \in S_j\},$$

where N_i is the direct neighborhood of a point i, S_i is the set of the points that strongly influence the point i and S_i^{T} is the points that are strongly influenced by point i. Here the influence concept is defined in terms of the absolute value of the nonzero entries. For any set P, $|P|$ denotes the number of elements in P.

The coarsening process is defined as follows [7]:

AMG coarsening process:

1. Input: the $n \times n$ matrix A_k (level k).
2. Initialize: $\omega_{\mathrm{C}}^k = 0$, $\omega_{\mathrm{F}}^k = 0$, $U = \omega_k$ and $\lambda_i = |S_i^{\mathrm{T}}|$.
3. Loop until $|\omega_{\mathrm{C}}^k| + |\omega_{\mathrm{F}}^k| = n$:
   (a) get $i \in U$ with maximal λ_i and set $\omega_{\mathrm{C}}^k = \omega_{\mathrm{C}}^k \cup \{i\}$, $U = U - \{i\}$,
   (b) for all $j \in S_i^{\mathrm{T}} \cap U$, make:
       set $\omega_{\mathrm{F}}^k = \omega_{\mathrm{F}}^k \cup \{j\}$, $U = U - \{j\}$,
       for all $l \in S_j \cap U$, set $\lambda_l = \lambda_l + 1$,
   (c) for all $j \in S_i \cap U$, set $\lambda_j = \lambda_j - 1$.

In this coarsening process the *strength threshold* α, which is a positive constant smaller than one, is modified in order to increase $|S_i|$ and $|S_i^{\mathrm{T}}|$ that ensures a reduction of $|\omega_{\mathrm{C}}^k|$ at each level. In contrast to standard AMG which uses a fixed value, here α is expressed as a function of the matrix density d, defined as the ratio between the nonzero number and the square of the number of rows of the matrix.

$$\alpha(d) = \frac{\mathrm{e}^{-(3d-1)^2} + 3d}{3 + d}. \tag{4}$$

This function is illustrated in Fig. 1.

We can say that if the matrix density d is low then a point i will have few direct neighbors, in other words, $|N_i|$ will be small. In this way, the value of α should also be small to allow that more direct neighbors of that point i make part of the set S_i. Hence, the number of elements in the set S_i^{T} will also increase, reducing $|\omega_{\mathrm{C}}|$. On the other hand, when the matrix density d is high the value of α should increase to ensure an enough number of points in $|\omega_{\mathrm{C}}|$ to achieve an accurate interpolation. This approach preserves the robustness of the method and

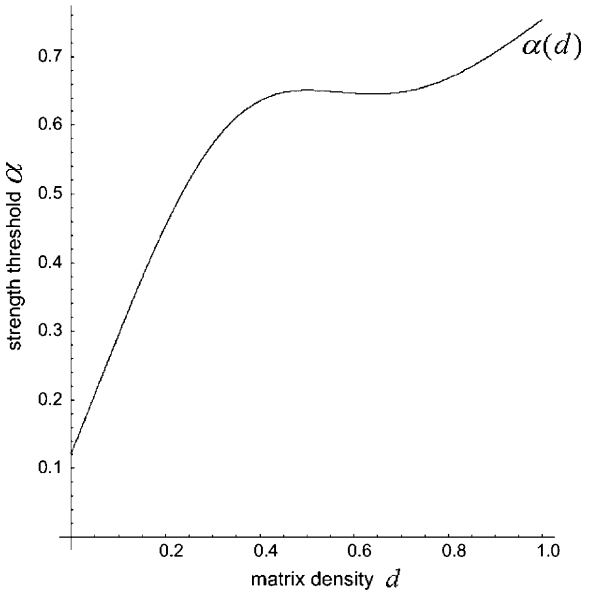

Fig. 1. α as a function of the matrix density d.

allows a reduction in the AMG setup and resolution time. Expression (4) is a possible choice for $\alpha(d)$ that incorporates the above analysis.

The storage space required by the operators A_k and for the right hand sides and approximation vectors over all grids are also significantly reduced. Just as in the geometric case, the work in the solver phase of AMG is dominated by relaxation and residual computations, which are directly proportional to the number of nonzero entries in the operator [3]. Hence, the work of a V-cycle, which is used here in the preconditioning, turns out to be proportional to the *operator complexity* that is defined as the total number of nonzero entries, in all matrices A_k, divided by the number of nonzero entries in the fine-grid operator $A_1 = A$. The measure of the storage space required for the vectors is given by the *grid complexity*, which is defined as the total number of grid points, on all grids, divided by the number of grid points on the finest grid.

4. Test cases

The proposed AMG preconditioning method was applied in the resolution of linear systems with ill-conditioned matrices. These matrices were obtained from the Matrix Market [6] and Davis [12] sparse matrix collections and have the characteristics summarized in Table 1. In that table, the fourth column indicates the average nonzero per column and row in the matrix. All the matrices are symmetric and positive definite, hence the incomplete Cholesky conjugate gradient method (ICCG) was used for compare the convergence results.

This paper pays special attention to the use of the AMG as a preconditioner for the CG method (AMGCG) for small and medium problems. However, the method can be also used with great advantage for large problems. In this preconditioning a V-cycle scheme was used.

Table 1
Matrices characteristics

Matrix name	Size	Nonzero number	Nonzero per row	Condition number
bcsstk14	1806×1806	63,454	35	1.3e+10
sts4098	4098×4098	72,356	18	2.1e+08
bcsstk16	4884×4884	290,378	59	6.5e+01
bcsstk17	$10{,}974 \times 10{,}974$	428,650	39	6.5e+01

In the tests we used a $V(1,1)$-cycle of AMG as preconditioner for matrix bcsstk14 and a $V(2,2)$-cycle for others three matrices. We used the Gauss–Seidel method as smoother and the right hand side vector **f** was chosen as that the solution **u** of the problem (1) be a vector with all the elements equal to 1.0.

5. Results

The convergence of iterative methods like CG can be understood in terms of the eigenvalue analysis of the preconditioned matrix [2]. Thus, we analyze the efficiency of preconditioner AMG through of eigenvalue distribution analysis of the matrix after the preconditioning.

Fig. 2 shows the eigenvalue distribution for matrix bcsstk14 after the applications of the AMG and the incomplete Cholesky method (IC). For AMG the eigenvalues are clustered around 1 and a few eigenvalues are scattered between 1 and 0. This makes the AMG preconditioner more effective than IC for CG in this case. The eigenvalues are calculated directly from the preconditioned matrix which was explicitly created.

Table 2 shows the results of application of the AMGCG for matrix bcsstk14. The Table shows the values of grid and operator complexity and presents the total time for the AMGCG setup and the solver phase for the linear system with matrix bcsstk14. Several cases were analyzed, with fixed values for α and the proposed approach where α is variable and a function of the matrix density d.

The results in Table 2 suggest the use of small values for α because it produces a great reduction in the grid and operator complexity and, consequently, decrease in the AMG setup and solver phase times. However, the use of very small values for α can increase prohibitively the convergence factor, reducing the robustness of the method. Moreover, it is very difficult to determine the best fixed value for a given problem. On the other hand, Table 2 shows also that the use of a *strength threshold* α as a function of the matrix density d produces very similar results to the best results produced using a fixed α.

In Table 3 are presented the results of the application of AMGCG for matrix bcsstk16, for different values of α. These results show that the use of $\alpha = 0.15$ produces smaller setup time for that problem, but it is not enough to solve it. In this case, the best computational performance was obtained with use of $\alpha = \alpha(d)$.

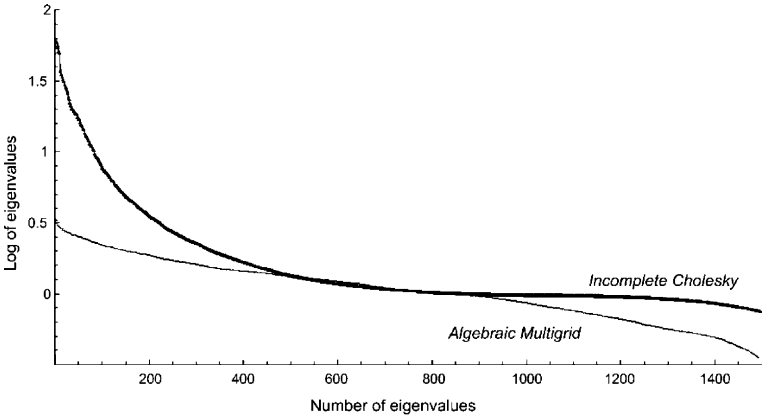

Fig. 2. Eigenvalues distribution after preconditioning with AMG and IC.

Table 2
AMG grid and operator complexity for matrix bcsstk14

Matrix	Method	Grid complexity	Operator complexity	Number of iteration	Total time (s)
bcsstk14	AMGCG(0.15)	1.39	1.30	61	5.34
	AMGCG (0.25)	1.50	1.43	61	6.01
	AMGCG (0.50)	1.61	1.52	61	6.05
	AMGCG (0.75)	1.72	1.52	61	5.90
	AMGCG ($\alpha = \alpha(d)$)	1.43	1.30	61	5.36
	ICCG	2.00	2.00	142	6.99

The statistics on coarsening for AMG with *strength threshold* α as a function of the matrix density d are shown in Table 4.

In Table 5, the results show the performance of AMGCG for matrix sts4098. Again, the use of AMG with *strength threshold* α as a function of the matrix density presented the best results.

Table 3
AMGCG for matrix bcsstk16

Value of α	Grid complex	Operator complex	Number of iteration	Total time (s)
0.15	1.22	1.13	–	–
0.25	1.28	1.23	47	61.48
0.50	1.47	1.57	45	79.75
$\alpha = \alpha(d)$	1.21	1.12	45	46.15

Table 4
Statistics on coarsening for AMG applied to matrix bcsstk16

Level	Value of α	Number of rows	Number of nonzeros	Density d
1	0.14	4884	290,378	0.012
2	0.25	670	33,886	0.075
3	0.22	170	1588	0.055
4	0.17	100	288	0.029
5	0.15	79	87	0.014

Table 5
AMGCG for matrix sts4098

Value of α	Grid complexity	Operator complexity	Number of iterations	Total time (s)
0.25	1.30	1.30	6	6.67
0.50	1.44	1.43	14	10.96
0.75	1.63	1.58	2	9.85
$\alpha = \alpha(d)$	1.24	1.26	6	6.02

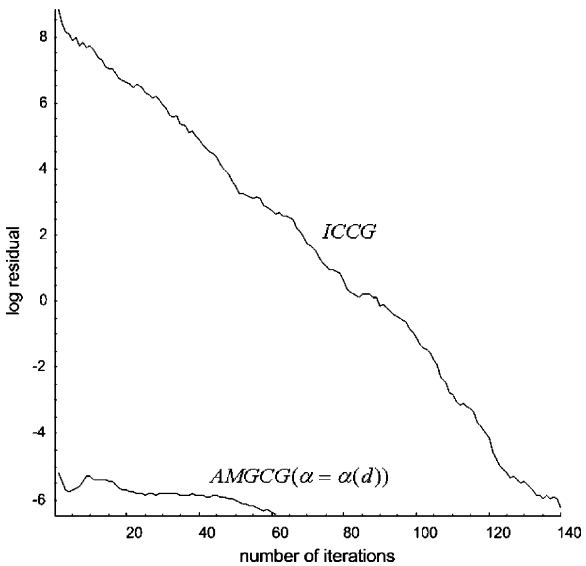

Fig. 3. Convergence results of the CG method for matrix bcsstk14.

The convergence results for matrix bcsstk14 were compared to that of the application of the ICCG for the same problem. These results are shown in Fig. 3.

The results of the application of AMGCG for the last case, the matrix bcsstk17, are presented next. Table 6 shows values related to the matrix in all levels, obtained from the AMG setup phase for different values of α. The first five lines of the table correspond to the use of *strength threshold* α as a function of the matrix density d.

The values presented in Table 6 shows that the use of α as function of d and $\alpha = 0.13$ produces very similar values. However, as shown in Fig. 4, the use of α as function of d, proposed here, is more effective in this case. Moreover, the use of other fixed values for α as 0.15, 0.25, 0.50 and 0.75, do not produce good preconditioning results and do not solve the problem. Is important to remember that, in contrast to basic AMG coarse-point

Table 6
Statistics on coarsening AMG for matrix bcssta17

α	Level	Number of rows	Number of nonzeros	Density (% full)	Operator complexity	Grid complexity
0.129	1	10,974	428,650	0.0036	1.17	1.43
0.144	2	2269	63,991	0.0124		
0.138	3	800	5756	0.0090		
0.129	4	596	1404	0.0039		
0.127	5	543	695	0.0024		
0.13	1	10,974	428,650	0.0036	1.17	1.39
	2	2330	67044	0.0123		
	3	797	5667	0.0089		
	4	581	1143	0.0034		
	5	535	605	0.0021		
0.25	1	10,974	428,650	0.0036	1.24	1.50
	2	2792	89,310	0.0115		
	3	924	8400	0.0098		
	4	645	2213	0.0053		
	5	564	890	0.0028		
0.50	1	10,974	428,650	0.0036	1.42	1.67
	2	4044	146,060	0.0089		
	3	1418	24,844	0.0123		
	4	777	5189	0.0086		
	5	615	1671	0.0044		

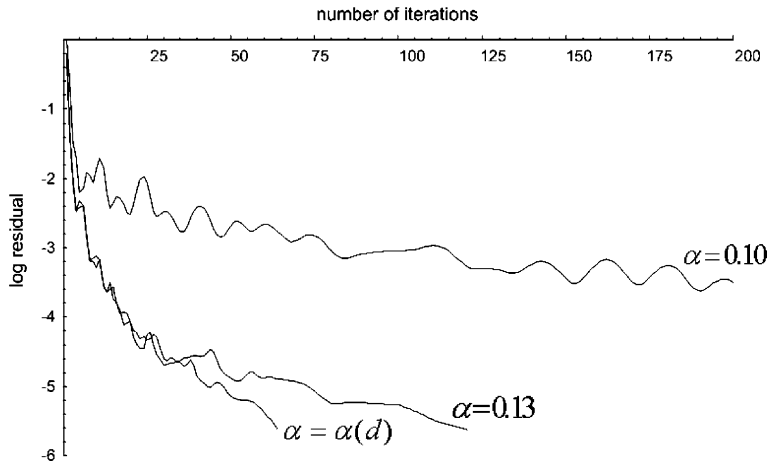

Fig. 4. AMGCG convergence for matrix bcsstk17.

Table 7
AMGCG results for matrix bcsstk17

Strength threshold	Number of iterations	Time for iteration (s)	Total time (s)
$\alpha = \alpha(d)$	59	0.532	80.64
$\alpha = 0.10$	1120	0.830	976.72
$\alpha = 0.13$	114	0.835	144.13

selection algorithm, which proceeds in two steps, here the objective is to create in a single step the best possible set of coarse nodes ω_C. Fig. 4 shows the first 200 AMGCG iterations for matrix bcsstk17 with $\alpha = 0.10$, $\alpha = 0.13$ and $\alpha = \alpha(d)$. More details for this problem can be seen in Table 7.

The results for matrix bcsstk17, presented in Fig. 4 and Table 7, show with clarity the best performance of $\alpha = \alpha(d)$. Besides spending a smaller number of iterations the time spend in each iteration was also smaller, resulting in a reduction about 45% of the total time regarding to the best result from a fixed α. Moreover, the use of other fixed values for α as 0.15, 0.25 do not solve the problem.

6. Conclusions

The paper presented a new approach to the use of the AMG as a preconditioner for the CG method in the resolution of small and medium size ill-conditioned problems. The proposed approach allowed a significant reduction in the AMG setup and solver times and the storage space required by the operators and for the right sides and approximation vectors over all grids of the AMG.

The results presented prove the robustness of the method for the test problems. For matrix bcsstk17, in which the use of "standard" $\alpha = 0.25$ is not effective, the $\alpha = \alpha(d)$ produce a smaller number of iterations reducing about 45% of the total time, regarding the best result from a fixed α. Moreover, for all other tests the results from $\alpha = \alpha(d)$ are always equal or better than to those with fixed α.

The matrices used in the tests possess a great number of nonzero elements per row, allowing small α to produce a great reduction in the values of grid and operator complexity. However, the use of a much small α can commit the performance of the method. In these cases, the new approach proposed here presented very good results.

References

[1] C. Iwamura, F.S. Costa, I. Sbarski, A. Easton, N. Li, An efficient algebraic multigrid preconditioner conjugate gradient solver, Comput. Methods Appl. Mech. Engrg. 192 (2003) 2299–2318.
[2] O. Tatebe, The Multigrid Preconditioned Conjugate Gradient Method, vol. 3224, NASA Conference Publication, 1993, pp. 621–634.
[3] W.L. Briggs, V.E. Henson, S.F. McCormick, A Multigrid Tutorial, second ed., SIAM, California, 2000.
[4] A.J. Cleary, R.D. Falgout, V.E. Henson, J.E. Jones, Coarse-grid Selection for Parallel Algebraic Multigrid, Lawrence Livermore Nat. Laboratory, Livermore, 2000.
[5] G. Haase, M. Kuhn, S. Reitzinger, Parallel algebraic multigrid methods on distributed memory computers, SIAM J. Sci. Comput. 24-2 (2001) 410–427.
[6] National Institute of Standards and Technology, Matrix Market. Available from: <http://math.nist.gov/MatrixMarket/>.
[7] Q. Chang, Y.S. Wong, H. Fu, On the algebraic multigrid method, J. Comput. Phys. 125 (1996) 279–292.
[8] D. Braess, Towards algebraic multigrid for elliptic problems of second order, Computing 55 (1995) 379–393.
[9] J.W. Ruge, K. Stüben, Algebraic multigrid, in: Multigrid Methods, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1987.
[10] A. Brandt, Algebraic multigrid theory: the symmetric case, Appl. Math. Comput. 19 (1986) 23–56.
[11] K. Stüben, A review of algebraic multigrid, J. Comput. Appl. Math. 128 (2001) 281–309.
[12] T. Davis, University of Florida Sparse Matrix Collection. Available from: <http://www.cise.ufl.edu/research/sparse/matrices>, NA Digest, vol. 92, no. 42, October 16, 1994, NA Digest, vol. 96, no. 28, July 23, 1996, and NA Digest, vol. 97, no. 23, June 7, 1997.