THE ARCHIVES OF TRANSPORT

VOL. XXII NO 1 2010

10.2478/v10174-010-0006-1

Algebraic Multigrid Preconditioning for Iterative Eigensolvers

Matthias Krüger

Received November 2009

Abstract

The paper presents a comparative study of iterative solvers for eigen- problems, which arise e.g. in solid mechanics or structural analysis. We consider problems obtained by discretization of elliptic and self-adjoint partial differential operators. Typically, only a few of the smallest eigen- values of these problems are to be computed. We discuss various gradient based preconditioned eigensolvers which make use of algebraic multigrid preconditioning. We present algorithms together with numerical results. Performance characteristics are derived by a comparison with the solution of test problems. We show that known advantages of algebraic multigrid preconditioning (e.g. for boundary-value problems with large jumps in the coefficients) transfer to the eigensolvers considered here.

1. Introduction

We consider the generalized eigenvalue problem

$$Av = \lambda Mv, A, M \in \mathbf{R}^{n\times n}, \quad \lambda \in R, v \in \mathbf{R}^n \tag{1}$$

arising from a finite element discretization of an elliptic and self-adjoint partial differential operator. The stiffness matrix A and the mass matrix M are symmetric positive definite matrices. Many applications demand a high accuracy, therefore the discretization results in a large matrix eigenproblem. Here, the task is to compute a few of the smallest eigenvalues and their corresponding eigenvectors of the pencil (A, M). For this reason we consider a class of preconditioned gradient type iterations. These methods are known to work very well for this type of eigenproblems, [1, 2, 3, 8]. Frequently, geometric multigrid preconditioning is applied to such problems. For these problems such methods can have nearly optimal computational complexity.

However, recent publications suggest algebraic multigrid (AMG) as a convenient choice for this class of problems [3, 6, 10]. AMG works on matrix information only and can be regarded as a gridfree process. This turns out to be an advantage if we consider complex domains or unstructured meshes because the geometric approach highly depends on the grid structure. The paper is organized as follows: Section 2 contains some remarks on the AMG method and its application as a preconditioner. Further each of the algorithms is presented. Then, Section 3 show results and performance characteristics for test problems, in particular for problems with varying coefficients.

2. Overview of Algorithms

In this section we examine the essential tools needed for the construction of the algorithms. Basically, we obtain them by applying a Rayleigh-Ritz procedure to a certain subspace. The choice of this subspace is crucial, and the precon- ditioned residuals of a eigenpair (θ, v) provides important update information.

The preconditioned residual d for a given eigenpair and matrix pair (A, M) reads

$$d = B - 1(Av - Mv\theta) \tag{2}$$

Formally, $\boldsymbol{B}^{-1}$ denotes the action of an abstract preconditioner. Here this action will be carried out by AMG. In the case of subspace iteration with a set of approximations a blockwise formulation reads

$$D = \boldsymbol{B}^{-1}(AV - MV\Theta) \tag{3}$$

with V being a matrix whose columns contain the approximated eigenvectors and Θ being a matrix with the eigenvalue approximations on the diagonal.

The next subsection gives a rough overview of AMG by explaining the structural principles. Afterwards the way of using the obtained preconditioned residuals within the Rayleigh-Ritz procedure is explained and some algorithms are dis- cussed and numerically tested.

2.1. Algebraic Multigrid

All of the eigensolvers use the algebraic multigrid (AMG) routine as a black-box operation to compute the preconditioned residuals, i.e. Equation (2). Never- theless we will recollect general working principles. Unlike most geometric approaches a main property of AMG is the independence of an underlying geometry. As a grid free process it becomes very attractive for unstructured meshes or if no grid information is available. The AMG method divides sharply into two parts, namely the *setup phase* and *the solution* phase. The setup phase serves to construct the hierarchy

and the neccessary operators. The solution phase then provides the approximated solution of the system of equations.

A generic two-grid correction step for the equation Au = f is sketched in Procedure 1. Two consecutive levels Ω^h and Ω^H are considered where h denotes the fine grid and H the coarse one. Moreover, the operators A_h (discrete formof A), the restriction operator R_h^H (to transfer quantities from fine to coarse grid), the prolongation operator P_H^h (to transfer quantities from coarse to fine grid) and a smoother S are assumed to be given. With $A_H := R_h^H A_h P_H^h$ and an approximation u_h on Ω_h a two-grid correction step, cf. Procedure 1 below, can be applied.

Procedure 1 Two-grid correction

Required: A_h, A_H, R_h^H, P_H^h

(1) Smooth $A_h u_h = f_h$ by application of S for a number of ν_1 times

(2) Compute residual $r_h = f_h - A_h u_h$

(3) Transfer r_h to coarser level Ω_H by restriction R_h^H

(4) Solve $A_H e_H = r_H$

(5) Transfer e_H to finer level by prolongation P_H^h

(6) Correct u_h by e_h

(7) Smooth $A_h u_h = f_h$ by application of S for a number of ν_2 times

An extension of Procedure 1 to a multigrid process works in a straightforward way. In order to approximately solve $A_H e_H = r_H$ in step (4) the two grid scheme is called recursively. Algorithmic variants are the V-cycles or W-cycles, see [5].

The construction of the hierarchy, precisely the *coarsening process* plays an important role for the efficiency of AMG. Several suggestions can be found in the literature depending on the type of problem that is considered. In the following we focus on two approaches. The first one makes use of “classical” coarsening suggested by Stüben [10, 11]. This method is provided by an algebraic multigrid toolbox implemented by Jane Cullum, Menno Verbeek and Wayne Joubert. The second approach applies the method of smoothed aggregation based AMG that is used by means of the ML package [4, 12, 13]. Both methods are regarded to work for these class of problems. The effects on the coarsening process are exposed in Section 3.

2.2. Subspaces and Algorithms

With the additional information gained by the computed preconditioned residuals as described in the previous section we can improve the current eigenpair approximations. This is done by the Rayleigh-Ritz procedure that is outlined in Procedure 2, see [9]. This procedure computes the (in some sense) optimal approximations for the smallest eigenpairs $(\theta_i, v_i), i = 1, \ldots, s$ in a subspace Z spanned by the columns of the input matrix $Z \in \mathrm{R}^{n \times s}$. Hence the choice of Z is significant and furthermore it will classify the algorithms.

Procedure 2 Rayleigh-Ritz procedure $RR(Z, s)$

Required: Z, V, A, M

(1) Orthonormalize the columns of Z

(2) Form the projections $\tilde{A} = Z^T AZ$ and $\tilde{M} = Z^T MZ$

(3) Compute the eigenpairs $(\Theta, \tilde{V})$ of the pencil $(\tilde{A}, \tilde{M})$, Θ contains the Ritz values

(4) Calculate the Ritz vectors V by $Z\tilde{V}$

(5) Sort Ritz pairs in a nondecreasing order

With these tools, the AMG routine and the Rayleigh-Ritz procedure, we can form the algorithms. As already mentioned we consider gradient based itera- tions, i.e. gradient based iterations for the Rayleigh quotient $\rho(x)$. The latter reads

$$\rho(x) = \frac{(x, Ax)}{(x, Mx)}$$

and therefore the gradient with respect to x is given by

$$\nabla\rho(x) = \frac{2}{(x, Mx)} \cdot (Ax - \rho(x) M$$
$$x)$$

where $(\cdot, \cdot)$ denotes the euclidean inner product. If we reconsider Equation (2) we realize that the preconditioned residuals are strongly related to the gradient of the Rayleigh quotient. They can be regarded as *preconditioned gradient vectors*.

The first algorithm uses a simple correction of the Ritz vectors in the negative direction of the preconditioned residuals. Formally, this reads

$$V_{k+1} = V_k - B^{-1}(AV_k - \Theta M V_k)$$

where k denotes the iteration index. Afterwards the new Ritz pairs are com- puted by the Rayleigh-Ritz procedure applied to $Z = V_{k+1}$. This iteration is known as *preconditioned inverse iteration* (PINVIT). It is sketched in Alg. 1.

Algorithm 1 PINVIT

Initialization:

(1) Select a random $\tilde{V}_0 \in \mathrm{R}^{n \times s}$, where $1 \leq s \leq n$ is the block size;

Compute $(\ , \Theta\ ) := RR(\tilde{V}, s)$; $k := 0$;

V_0 0 0

Iteration:

(2) Compute $R_k := AV_k - MV_k\Theta_k$

(3) Solve $D_k = B^{-1}R_k$

(4) Set $Z_k = V_k - D_k$

(5) Compute $(V_{k+1}, \Theta_{k+1}) = RR(Z_k, s)$

(6) Mark converged Ritz pairs $(v_{k+1}^{(i)}, \theta_{k+1}^{(i)})$, $i = 1, \ldots, nev$

(7) Stop if number of converged pairs is equal or greater nev, else $k = k + 1$ and goto (2).

Step PINVIT-(3) applies the AMG method that computes the preconditioned residuals. For each Ritz pair $(\theta_k^{(i)}, v_k^{(i)})$ that does not meet the convergence condition a single V-cycle is performed. Step PINVIT-(6) marks Ritz pairs that fulfill the convergence condition $// Av^{k}\,(\mathrm{i}) - \theta^{k}\,(\mathrm{i})\, v^{k}\,(\mathrm{i}) //2 \leq tol$. Already converged Ritz pairs are used further for the orthonormalization process but are not taken into account for a new computation of residuals in step PINVIT-(3). The algorithm stops if the number of requested eigenpairs is gained.

Basically this algorithm, as well as the following, consists of an inner and outer iteration. The inner one is formed by the computation of the preconditioned residuals done by the AMG routine, cf. Equation (2). The outer iteration is the determination of new Ritz pairs by the Rayleigh-Ritz procedure. Hence, AMG is used only to

obtain new information (preconditioned residuals) for improvement of the current Ritz approximations.

Algorithm 2 PSD

Initialization:

(1) Select a random $\tilde{V}_0 \in \mathrm{R}^{n\times s}$, where $1 \le s \le n$ is the blocksize;

Compute $(V_0, \Theta_0) := RR(\tilde{V}_0, s)$; $k := 0$;

Iteration:

(2) Compute $R_k := AV_k - MV_k\Theta_k$

(3) Solve $D_k = B^{-1}R_k$

(4) Set $Z_k = [V_k, D_k]$

(5) Compute $(\tilde{V}_k, \Theta_k) = RR(Z_k, 2s)$

(6) Set $V_{k+1} = Z_k\tilde{V}_k$

(7) Mark converged Ritz pairs $(v_{k+1}^{(i)}, \theta_{k+1}^{(i)})$, $i = 1, \ldots, nev$

(8) Stop if number of converged pairs is equal or greater than nev, else $k = k+1$ and goto (2).

The second algorithm implements the method of preconditioned steepest descent (PSD), see Alg. 2. In contrast to Alg. 1 the matrix Z is formed in such a way that the columns contain the current Ritz vectors and the preconditioned residuals. Therefore $Z(\in \mathbf{R}^{n\times 2s})$ grows in size and the computational costs for solving the projected eigenproblem do grow as well. Nevertheless, a larger subspace results in better approximations for the Ritz pairs. And this, finally, in a faster convergence as the examples will show.

A further extension of the subspace spanned by Z leads to the locally optimal *blockwise preconditioned conjugate gradient* method (LOBPCG), introduced by Knyazev [7], which is sketched in Alg. 3. Here we enlarge the subspace to which the Rayleigh-Ritz method is applied by the previous iterates V_{k-1}.

Algorithm 3 LOBPCG

Initialization:

(1) Select a random $\tilde{V}^0 \in \mathrm{R}^{n \times s}$, where $1 \leq s \leq n$ is the blocksize;

Compute $(\ , \Theta\ ) := RR(\tilde{V}, s)$; $k := 1$;

V_0 0 0

Iteration:

(2) Compute $R_k := AV_k - MV_k\Theta_k$

(3) Solve $D_k = B^{-1}R_k$

(4a) If k=1: Set $Z_k = [V_k, D_k]$

(4b) else : Set $Z_k = [V_k, D_k, V_{k-1}]$

(5a) If k=1: Compute $(\tilde{V}, \Theta\ ) = RR(Z\ , 2s)$

k k k

(5b) else : Compute $(\tilde{V}, \Theta\ ) = RR(Z\ , 3s)$

k k k

(6) Set $V_k = Z_k\tilde{V}^1$ and $V_{k+1} = [0, D_k, V_{k\ 1}]\tilde{V}^1$ with $\tilde{V}\ = [\tilde{V}^1, \tilde{V}^2, \tilde{V}^3], \tilde{V}^p \in$

k $-$ k k k k k k

$\mathrm{R}^{n \times s}$, $p = 1, 2, 3$

(7) Mark converged Ritz pairs $(v_{k+1}^{(i)}, \theta_{k+1}^{(i)})$, $i = 1, \ldots, nev$

(8) Stop if number of converged pairs is equal or greater than nev, else $k = k+1$ and goto (2).

The extension necessitates a modification in step LOBPCG-(4) and step LOBPCG-(5). If $k = 1$ no previous iterate exists. We obtain them by inserting one step of the PSD method. Step LOBPCG-(6) then computes the sets of new and previous Ritz vectors. Again, this increases the number of operations per step. A further extension of the subspace spanned by Z with older iterates leads to "higher order" schemes. Similiar to LOBPCG, the sequence of older iterates has to be computed by proper algorithms of "lower order".

3. Results

In order to validate the algorithms we consider the following two dimensional test problem

$$\begin{aligned} -\nabla \cdot (\varepsilon \nabla u) &= \lambda u \\ u &= 0 \quad on \quad \Gamma_1 \\ \frac{\partial u}{\partial n} &\quad on \quad \Gamma_2 \end{aligned} \tag{4}$$

on a missing wedge unit circle $\Omega = \{(t, \varphi); t \in [0,1], \varphi \in [0,2\pi - \alpha]\}$ for $\alpha > 0$.

The boundaries are

$\Gamma 1 = \{(t, \varphi) : t \in [0,1], \varphi = 0\}$ and $\{t = 1, \varphi \in [0,2\pi - \alpha]\}$
$\Gamma 2 = \{(t, \varphi) : t \in [0, 1], \varphi = 2\pi - \alpha\}$

The domain Ω for the angle $\alpha = \pi/12$ is shown in Figure 1 on the left.

3.1. Test Problem

The analytical solution of equation (1) with $\varepsilon \equiv 1$ is given by

$$u_{j,l}(t, \varphi) = c \cdot \sin(\nu(j, \alpha)\, \varphi) \cdot J_{\nu(j,\alpha)}(\omega_{j,l} t), \; j, l = 0,1, \ldots$$

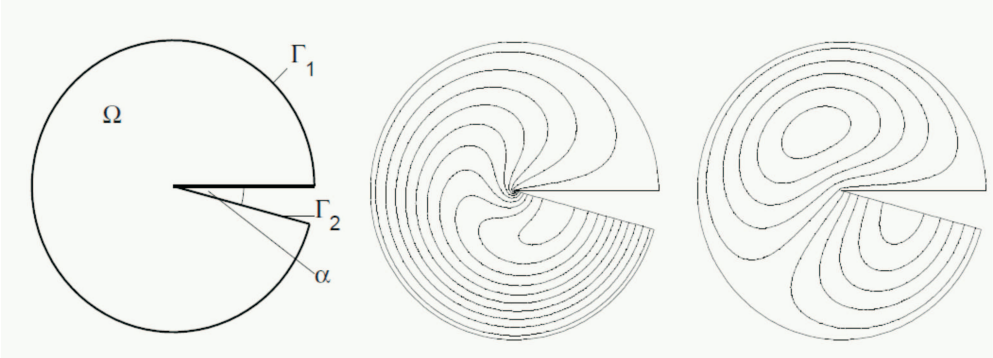

Fig. 1. Domain and two smallest eigenmodes

with

$$\upsilon(j, \alpha) = \frac{j + \frac{1}{2}}{2 - \frac{\alpha}{\pi}}$$

and $J_{\nu(j,\alpha)}$ being the Bessel functions of first kind and fractional order $\nu(j, \alpha)$. The exact eigenvalues are the squares of the positive zeros $(\omega_{j,l})^2, (j, l = 0, 1, \ldots)$ of

$J_{\nu(j,\alpha)}$. The smallest eigenvalues are given in the first column of Table 1. A surface plot of the eigenmodes corresponding to the two smallest eigenvalues are depicted in Figure 1.

For the computation we consider a finite element discretization of (1) resulting in matrices A and M of size n= 297078. We apply the presented algorithms PINVIT, PSD and LOBPCG as well as two schemes of higher order. The subspace dimension is s= 20. We request at least 15 eigenpairs to be stationary, in other words the quantity of the error for the i-th eigenpair in the k-th iteration step should satisfy

$$err_i = \|A\upsilon_i^{(k)} - \theta_i^{(k)} M\upsilon_i^{(k)}\|_2 \leq 10^{-10}, \quad i = 1, ..., 15$$

The construction of the hierarchy is based on the "classical" coarsening strategy for all calculations. This method is provided by the algebraic multigrid toolbox (CTB) by Jane Cullum et. al. The action of the preconditioner B is performed by executing a single V-cycle. The computed eigenvalues and respective final residuals for PINVIT, PSD and LOBPCG are given in Table 1. In no case any eigenpair was missed. Additionally, the number of needed iterations for each algorithm is stated at the top. The results of higher order schemes are similar to the ones obtained by LOBPCG, presented in the last column of Table 1.

Figure 2 shows the normalized computational times for the solution phase. The costs for the setup phase are excluded because they are for all algorithms identically. We can conclude that LOBPCG appears to be the most efficient algorithm. This coincides with an observation made in [3]. As consequence of this fact we focus on LOBPCG for the next computations.

Table 2 shows results for the 15-th eigenvalue obtained by LOBPCG for different meshes. Moreover, we now utilize both the "classical" coarsening strategy (provided by CTB) and smoothed aggregation (provided by ML). L denotes

Table 1

Exact and computed eigenvalues

	PINVIT, Ni_{ter} = 65		PSD, Ni_{ter} = 28		LOBPCG, Ni_{ter} = 18	
λ_{ex}	$\lambda_{k,l}$	$\lvert err\rvert$	$\lambda_{k,l}$	$\lvert err\rvert$	$\lambda_{k,l}$	$\lvert err\rvert$
7.822386	7.96429	8.05E-11	7.96429	1.44E-11	7.96429	1.80E-11
12.502574	12.50297	2.93E-11	12.50297	2.36E-11	12.50297	3.26E-11
17.953688	17.95399	6.76E-11	17.95399	4.49E-11	17.95399	5.86E-11
24.148219	24.14876	9.24E-11	24.14876	2.98E-11	24.14876	1.77E-11
31.065919	31.06684	3.89E-11	31.06684	6.45E-11	31.06684	2.93E-11
35.078513	35.53111	3.40E-11	35.53111	2.02E-11	35.53111	5.60E-11
38.691135	38.69259	5.41E-11	38.69259	3.02E-11	38.69259	7.27E-11
44.894358	44.89748	5.34E-11	44.89748	7.93E-11	44.89748	2.25E-11
47.011332	47.01348	8.37E-11	47.01348	2.13E-11	47.01348	2.22E-11
55.500948	55.50382	4.46E-11	55.50382	4.69E-11	55.50382	5.10E-11
56.016195	56.01922	4.63E-11	56.01922	3.52E-11	56.01922	5.28E-11
65.697054	65.70124	8.23E-11	65.70124	3.92E-11	65.70124	2.10E-11
66.884838	66.88902	7.19E-11	66.88902	4.80E-11	66.88902	1.79E-11
76.046502	76.05212	5.40E-11	76.05212	3.22E-11	76.05212	4.45E-11
79.033483	79.03937	9.34E-11	79.03937	6.36E-11	79.03937	5.01E-11

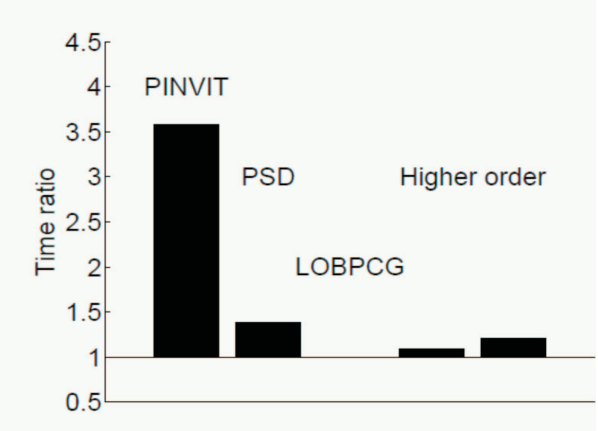

Fig. 2. Ratio of (normalized) computational times

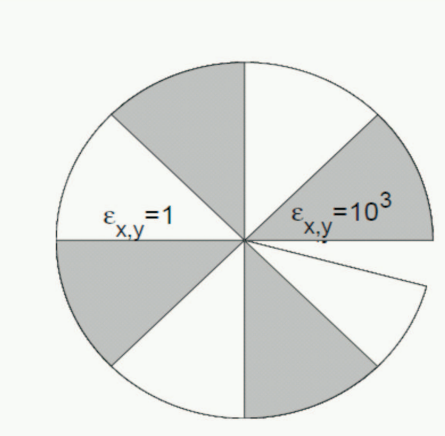

Fig. 3. Localisation of ε

Table 2

Performance characteristics of LOBPCG for two different preconditioners

n	CTB			ML		
	L	N_{iter}	λ_{15}	L	N_{iter}	λ_{15}
97429	8	17	79.05168	4	49	79.05168
191257	9	17	79.04268	4	60	79.04268
297078	9	17	79.03937	4	66	79.03937
403688	10	17	79.03792	4	70	79.03792
615636	10	17	79.03634	4	74	79.03634
857811	10	17	79.03557	4	83	79.03557

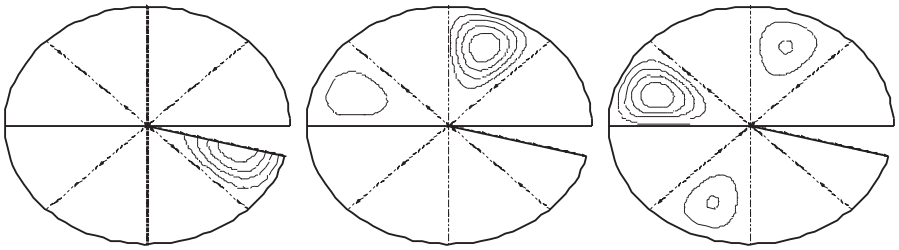

Fig. 4. Three smallest eigenmodes for the test problem with ε according to equation (5)

the number of levels constructed in the setup phase. A single V-cycle is executed to compute the preconditioned residuals. For n= 97429 CTB generates L= 8 levels with, respectively, 97429, 44051, 15465, 5637, 2046, 743, 270 and 96 unknowns. In contrast, ML construtcs a hierarchy consisting of L= 4 levels with 97429,11863,798 and 45 unknowns. The smoothed aggregation approach leads to a "faster" coarsening reflected in the smaller number of levels. But this results in less exact preconditioned residuals and in a higher number of itera- tion steps, too. Nevertheless, both approaches compute the same eigenvalue, cf. Table 2.

3.2. Test problem with varying coefficients

In this section we want to underline the capability of AMG to deal with eigenproblems with jumps in the coefficients. For such problems it is known that solvers which make use of geometric multigrid methods loses its potential con- vergence properties.

For this purpose we consider equation (1) with ε defined by

$$\varepsilon = \begin{cases} 1 & \text{for} \quad \varphi \in \left[(2m+1)\frac{\pi}{4}, (2m+2)\frac{\pi}{4}, t \in [0,1]\right. \\ 10^3 & \text{for} \quad \varphi \in \left[(2m)\frac{\pi}{4}, (2m+1)\frac{\pi}{4}, t \in [0,1]\right. \end{cases} \tag{5}$$

for m= 0,1,2,3. The distribution of ε is depicted in Figure. 3. LOBPCG (with adapted matrices $A$and $M$of size n= 297078) is applied to compute the 15-th smallest eigenpairs. Again, the first variant utilizes CTB as precondi- tioner. We obtain the three smallest eigenvalues, θ_1= 40.91474, θ_2= 57.87177 and θ_3= 57.96446. The corresponding eigenmodes are depicted in Figure 4. The inhomogeneity of ε leads to a strong localization of the gradients of the eigenfunctions within the areas with small conductivity.

In contrast to the homogenous case considered in Section 3.1 a hierarchy with L= 10 levels is constructed. Consequently, we observe a modified behavior in the case of "classical" coarsening. However, for CTB the number of needed iteration steps increases to N_{iter}= 19. Whereas ML works again on a hierar- chy with L= 4 levels. Due to this the number of iterations increases to 191.

Nevertheless, the same eigenpairs are computed.

4. Conclusion

A class of gradient based eigensolvers which make use of algebraic multigrid preconditioning has been presented and tested. We conclude from performance characteristics that LOBPCG is the most efficient method. Furthermore, two variants of AMG preconditioning which differ in underlying coarsening strate- gies were applied. The computation of test problems showed that AMG pre- conditioning can be applied to such problems successfully. In particular, the capability of AMG to deal with problems with large jumps in the coefficients was confirmed. Consequently, preconditioning by AMG is a well-working alter- native to the so far predominant geometric multigrid approach.

References

1. Arbenz P., Hetmaniuk U.L., Lehoucq R.B., Tuminaro R.S.: A comparison of eigensolvers for large-scale 3D modal analysis using AMG- preconditioned iterative methods. Int. J. Numer. Methods Eng., 64(2):204-236, 2005.
2. e. Bai Z., e. Demmel J., e. Dongarra J., e. Ruhe A., e. Van der Vorst H.: Templates for the solution of algebraic eigenvalue problems. A prac- tical guide. Software – Environments – Tools. 11. Philadelphia, PA: SIAM, Society for Industrial and Applied Mathematics. xxix, 410 p. , 2000.
3. Borzì A., Borzì G.: Algebraic multigrid methods for solving generalized eigenvalue problems. Int. J. Numer. Methods Eng., 65(8):1186-1196, 2006.
4. Gee M., Siefert C., Hu J., Tuminaro R., Sala M.: ML 5.0 smoothed ag- gregation user's guide. Technical Report SAND2006-2649, Sandia National Laboratories, 2006.
5. Hackbusch W.: Iterative solution of large sparse systems of equations. Transl. from the German. Applied Mathematical Sciences. 95. New York, NY: Springer-Verlag. xxi, 429 p. DM 108.00; ¨os 842.40; sFr. 108.00 , 1994.
6. Hetmaniuk U.L.: A Rayleigh quotient minimization algorithm based on algebraic multigrid. Numer. Linear Algebra Appl., 14:563-580, 2007.
7. Knyazev A.V.: A preconditioned conjugate gradient method for eigenvalue problems and its implementation in a subspace. Numerical treatment of eigenvalue problems. Vol. 5, Proc. Workshop, Oberwolfach/Germ. 1990, ISNM 96, 143-154, 1991.
8. Knyazev A.V., Neymeyr K.: A geometric theory for preconditioned in- verse iteration. III: A short and sharp convergence estimate for generalized eigenvalue problems. Linear Algebra Appl., 358(1-3):95-114, 2003.
9. Parlett B.N.: The symmetric eigenvalue problem. Prentice-Hall Series in Computational Mathematics. Englewood Cliffs, New Jersey: Prentice-Hall, Inc. XIX, 348 p. $ 33.75 , 1980.
10. Stüben K.: A review of algebraic multigrid. J. Comput. Appl. Math., 128(1-2):281-309, 2001.
11. Stüben K.: Algebraic multigrid (AMG). An introduction with applications. Tech. Report 70, GMD, Sankt Augustin, Germany, November 1999.
12. Vaněk P., Brezina M., Mandel J.: Convergence of algebraic multigrid based on smoothed aggregation. Numer. Math., 88(3):559-579, 2001.
13. Vaněk P., Mandel J., Brezina M.: Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems. Computing, 56(3):179-196, 1996.