Algebraic Multilevel Methods And Sparse Approximate Inverses

*Matthias Bollhöfer and †Volker Mehrmann
Institut für Mathematik, MA 4-5
Technische Universität Berlin
Str. des 17. Juni 136
D–10623 Berlin, Germany

Abstract

In this paper we introduce a new approach to algebraic multilevel methods and their use as preconditioners in iterative methods for the solution of symmetric positive definite linear systems. The multilevel process and in particular the coarsening process is based on the construction of sparse approximate inverses and their augmentation with corrections of smaller size. We present comparisons of the effectiveness of the resulting multilevel technique and numerical results.

Keywords: sparse approximate inverse, large sparse matrices, algebraic multilevel method.

AMS subject classification: 65F05, 65F10, 65F50, 65Y05.

1 Introduction

For the solution of large sparse linear systems of the form

$$Ax = b,\ A \in \mathbb{R}^{n,n}, b \in \mathbb{R}^n, \tag{1}$$

sparse approximate inverses, i.e., sparse matrices that are good approximations of the inverse of a sparse matrix, [26, 25, 12, 18, 7] have become popular as preconditioners for Krylov–subspace [15, 33, 17] techniques. There are several techniques to construct such sparse approximate inverses. One may for example minimize the norm of $\|AB - I\|$ subject to some prescribed pattern [25, 12, 18]. Another technique is to construct upper triangular matrices $Z, W^\top$ such that for a diagonal matrix D, $W^\top AZ$ is a good approximation to D, [7]. Moreover success has been made over the years in using approximate inverses in combination with multilevel methods [13, 28, 27, 35, 36]. Especially in [36] it has been shown that by adjusting the quality of the approximate inverse, the smoothing property can be improved significantly.

We assume in the following that A is symmetric positive definite and that the approximate inverse B is factored as $B = LL^\top$. We set $M = L^\top AL$ and assume for simplicity that $\|M\|_2 \leqslant 1$. This can always be achieved by an appropriate scaling. We will concentrate on

*Supported by the DFG under grant BO 1680/1-1 and by the University of Minnesota. Part of this research was performed while visiting the University of Minnesota at Minneapolis. bolle@math.tu-berlin.de, http://www.math.tu-berlin.de/∼bolle/.

†Supported by SFB 393 "Numerische Simulation auf massiv parallelen Rechnern". mehrmann@math.tu-berlin.de. http://www.math.tu-berlin.de/∼mehrmann/.

sparse approximate inverses for which M is still sparse. This is for example the case if the approximate inverse is diagonal or block diagonal. Even factored sparse approximate inverses from [25, 26] can be used as long as the pattern of L is moderate. For example if the pattern of L is the same as the pattern of A (or the same pattern as the lower triangular part of A). There also exist sparse approximate inverse approaches that cannot be applied here, because they are only sparse with respect to certain basis transformations like wavelet–based sparse approximate inverses [11]. For large classes of matrices, sparse approximate inverses have proved very effective as preconditioners. But there are problems where the sparse approximate inverse needs a large number of nonzero entries to become a suitable approximation to the inverse of A. When using sparse approximate inverses based on norm–minimizing techniques, one often observes that many eigenvalues [16] of the residual matrix $E = I - M$ are quite small, while a small number of eigenvalues stay big. And allowing more fill–in in the sparse approximate inverse B does not cure this. For an example, see [8].

The observation that many eigenvalues are small but some stay large means that B approximates A^{-1} well on a subspace of large size, while there is almost no approximation on the complementary subspace. In the context of multigrid methods for the numerical solution of partial differential equations this effect is typically called smoothing property [19]. Algebraically this means that the residual $E = I - M$ can be written as

$$E = E_p + F, \tag{2}$$

where $E_p \in \mathbb{R}^{n,n}$ has rank $p < n$ and $\|F\| \leq \eta \ll 1$, i.e., the residual can be approximated well by a matrix of lower rank p. Typically one cannot expect that the size p of E_p is independent of the dimension n of A. More realistic is the assumption, that $p \approx cn$, where for example $c = \frac{1}{2}$ or $c = \frac{1}{4}$.

If one is solving a symmetric positive definite linear system $Ax = b$ and one has already determined some sparse approximate inverse B, it is therefore desirable (and our primary goal) to improve the preconditioner $LL^\top$. Our goal is to construct an updated preconditioner of the form

$$L(I + PZ^{-1}P^\top)L^\top \tag{3}$$

with sparse matrices P, Z, where Z is another symmetric positive definite matrix of smaller size. Since A and the augmented preconditioner are positive definite, this means that we are interested in the small eigenvalues (since $\|M\|_2 \leqslant 1$) of the preconditioned system

$$AL(I + PZ^{-1}P^\top)L^\top. \tag{4}$$

In other words we have to achieve that

$$\|I - M^{1/2}(I + PZ^{-1}P^\top)M^{1/2}\|_2 = \|E - M^{1/2}PZ^{-1}P^\top M^{1/2}\|_2 \tag{5}$$

is small, while at the same time P and Z are sparse.

Since the matrix E is symmetric positive semidefinite by assumption, it is well known [16], that the best approximation of E by a matrix of rank p is given by the matrix

$$\hat{E}_p = U_p \Sigma_p U_p^\top = \begin{bmatrix} u_1, \ldots, u_p \end{bmatrix} \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_p \end{bmatrix} \begin{bmatrix} u_1 & \ldots & u_p \end{bmatrix}^\top, \tag{6}$$

where $\sigma_1 \geqslant \cdots \sigma_n \geqslant 0$ are the eigenvalues of E and u_i, $i = 1, \ldots, n$ are the eigenvectors.

But in general, this best approximation will be a full matrix, since U_p is full even if E is sparse, and hence we cannot directly use $\hat{E}_p$ in the construction of sparse preconditioners.

Since we have assumed that the given approximate inverse B has the property that $E = I - L^\top AL$ is approximated well by $\hat{E}_p$ in the sense of (2), we have that the entries of $\hat{E}_p$ differ only slightly from the entries of E. So we may expect that taking an appropriate selection of columns of E as V will be a good choice for U and the approximation of E by a lower rank matrix. This expectation is justified by the following Lemma.

Lemma 1 *Let $E \in \mathbb{R}^{n,n}$ be symmetric positive semidefinite and let*

$$E = U\Sigma U^\top = [U_1, U_2] \begin{bmatrix} \Sigma_1 & \\ & \Sigma_2 \end{bmatrix} [U_1, U_2]^\top$$

be the spectral decomposition of E, where U is orthogonal, $U_1 \in \mathbb{R}^{n,p}$ and the diagonal entries of Σ are ordered in decreasing order. If E satisfies (2) (i.e. $E = E_p + F$ for a rank–p matrix E_p and $\|F\|_2 \leqslant \eta$) then there exists a permutation matrix $\Pi = [\Pi_1, \Pi_2]$, partitioned analogously, such that

$$\inf_{X \in \mathbb{R}^{p,p}} \|U_1 X - M^{1/2}\ (E\Pi_1)\|_2 \leqslant \eta. \tag{7}$$

Proof. Applying the QR decomposition with column pivoting [16] to $M^{1/2}\hat{E}_p = U_1(I - \Sigma_1)^{1/2}\Sigma_1 U_1^\top$, we obtain $Q, R^\top \in \mathbb{R}^{n,p}$, where Q is orthogonal, $R = [R_1, R_2]$, with $R_1 \in \mathbb{R}^{p,p}$, is upper triangular and $\Pi = [\Pi_1, \Pi_2]$ is a permutation matrix with Π_1 having p columns such that

$$M^{1/2}\hat{E}_p\Pi = QR.$$

It immediately follows that $M^{1/2}\hat{E}_p\Pi_1 = QR_1$ and thus there exists a nonsingular $p \times p$ matrix X such that $M^{1/2}\hat{E}_p\Pi_1 = QR_1 = U_1X$ and we have

$$\|M^{1/2}E\Pi_1 - U_1X\|_2 = \|M^{1/2}(E - \hat{E}_p)\Pi_1\|_2 \leqslant \|(E - \hat{E}_p)\Pi_1\|_2 = \min_{\substack{E_p \\ \operatorname{rank} E_p = p}} \|E - E_p\|_2 \leqslant \|F\|_2 = \eta.$$

□

Lemma 1 gives us subspaces that consist of suitably chosen columns of E, which are close to the subspace U_1 of E associated with the large eigenvalues of E in the sense of (7).

Using such subspaces in the construction of appropriate sparse representations of the updates $PZ^{-1}P^\top$ as in (3) is the topic of this paper, which is organized as follows.

We first discuss the theoretical background for this problem, i.e., to construct optimal preconditioners of this form, and show that they are closely related to algebraic multilevel methods. We derive two types (multiplicative and additive) of algebraic multilevel preconditioners in Section 2.

The approximation properties of the multiplicative correction term $I + PZ^{-1}P^\top$ in (3) for the two multilevel schemes are studied in detail in Section 3.

In view of Lemma 1, we may in principle use a QR–like decomposition of $M^{1/2}E$ to construct the desired updated preconditioners. The key in this construction is the appropriate pivoting strategy in the QR decomposition with column pivoting. We will present two heuristic pivoting strategies and interpret them as coarsening process of the multilevel scheme in Section 4.

Finally in Section 5 we present numerical examples that demonstrate the properties of this new approach and also indicate the effectiveness of the heuristics that have been used.

In the sequel, for symmetric matrices A, B we will use the notation $A \succeq B$, if $A - B$ has nonnegative eigenvalues. We also identify a matrix with the space spanned by its columns.

2 Multilevel Preconditioners

In this section we present two multilevel preconditioners for symmetric positive definite systems. Algebraic multilevel preconditioners have become popular in recent years. Several algebraic multigrid approaches focus on incomplete LU or Schur–complement approaches [4, 5, 14, 6, 30, 31] while others are based on the analogy to geometric multigrid methods [10, 32, 24, 21, 29, 23]. Here we will concentrate on the second class of approaches.

Let $A \in \mathbb{R}^{n,n}$ be symmetric positive definite and let $L \in \mathbb{R}^{n,n}$ be a given sparse matrix such that $LL^\top$ is a symmetric positive definite matrix in factored form that approximates A^{-1}.

Suppose that the approximation of A^{-1} by $LL^\top$ is not satisfactory, e.g., the condition number of $L^\top AL$ is not small enough to get good convergence in the conjugate gradient method, and we wish to improve the preconditioning properties. To do this we like to determine a matrix of the form

$$M^{(1)} = LL^\top + PZ^{-1}P^\top, \tag{8}$$

with $P \in \mathbb{R}^{n,p}$, $Z \in \mathbb{R}^{p,p}$ nonsingular, P, Z sparse and furthermore, $p \leqslant cn$ with $0 < c < 1$, so that $M^{(1)}$ is a better approximation to A^{-1} than $LL^\top$.
The particular form (8) is chosen close to the form of an algebraic two–level method, where multiplication with $P, P^\top$ corresponds to the mapping between fine and coarse grid and Z represents the coarse grid system. Note further, that using the representation $LL^\top + PZ^{-1}P^\top$ as a preconditioner for A, only a system with Z has to be solved. As shown in Lemma 1, skilfully chosen columns/rows of the residual matrix $E = I - L^\top AL$ can be used to approximate the invariant subspace of E associated with its large eigenvalues. As we will see, precisely this invariant subspace has to be approximated by P. In the sense of the underlying undirected graph of E we refer to the nodes associated with the columns/rows of E that will be used to approximate the invariant subspace of E associated with the largest eigenvalues as *coarse grid nodes* while the remaining nodes are called *fine grid nodes*. The process of detecting a suitable set of coarse grid nodes will be called *coarsening process*. Once we have selected certain nodes as coarse grid nodes, they are in a natural way embedded in the initial graph. In addition the graph of $W = P^\top AP$ is a natural graph associated with the coarse grid nodes. We will call it *coarse grid* in analogy to the notation arising in discretized partial differential equations.

Recalling the well-known techniques of constructing preconditioners for the conjugate gradient method applied to symmetric positive definite systems, e.g. [16, 20, 33], we should choose P and Z such that

$$\mu A^{-1} \preceq M^{(1)} \preceq \mu\kappa^{(1)} A^{-1}, \tag{9}$$

with $\kappa^{(1)}$ as small as possible and $\mu > 0$. Clearly $\kappa^{(1)} \geqslant 1$ is the condition number of $M^{(1)}A$, i.e., the ratio of the largest by the smallest eigenvalue of $M^{(1)}A$ and thus $\kappa^{(1)} = 1$ would be optimal. The importance of the condition number is justified from the well–known results on the performance of the conjugate gradient method with preconditioner $M^{(1)}$, see e.g. [16]. We discuss the construction of P, Z with minimal $\kappa^{(1)}$ below.

For discretized elliptic partial differential equations often, but not always, one can construct optimal preconditioners using multigrid methods [19]. In order to obtain a similar preconditioner augmented with a suitably chosen coarse grid correction, consider the use of $LL^\top$ in a

linear iteration scheme [37] for the solution of $Ax = b$ with initial guess $x^{(0)} \in \mathbb{R}^n$. Such an iteration is given by

$$x^{(k+1)} = x^{(k)} + LL^\top(b - Ax^{(k)}), k = 0, 1, 2, \ldots$$

The error propagation matrix $I - LL^\top A$ satisfies $x - x^{(k+1)} = (I - LL^\top A)(x - x^{(k)})$. In multilevel techniques [19] one uses such an iteration for pre- and post-smoothing and in addition one has to add a coarse grid correction. In terms of the error propagation matrix this means that instead of $I - LL^\top A$ we have $(I - LL^\top A)(I - PZ^{-1}P^\top A)(I - LL^\top A)^\top$ as error propagation matrix. A simple calculation shows that this product can be rewritten as $I - M^{(2)}A$ with

$$M^{(2)} = 2LL^\top - LL^\top ALL^\top + (I - LL^\top A)PZ^{-1}P^\top(I - ALL^\top). \tag{10}$$

Again we are interested in choosing P, Z such that

$$\mu A^{-1} \preceq M^{(2)} \preceq \mu\kappa^{(2)}A^{-1}, \tag{11}$$

with $\kappa^{(2)}$ as small as possible.

In the following we discuss the approximation properties of $M^{(1)}, M^{(2)}$. The first step will be the construction of optimal P, Z for given A, L based on the spectral decomposition

$$E \equiv I - L^\top AL = \Psi\Lambda\Psi^\top, \tag{12}$$

where $\Lambda = \operatorname{diag}(\lambda_1 \ldots, \lambda_n)$, $\lambda_1 \geqslant \cdots \geqslant \lambda_n$ and $\Psi = [\psi_1, \ldots, \psi_n]$ is orthogonal. We use the notation $\Psi_p = [\psi_1, \ldots, \psi_p]$, $\Lambda_p = \operatorname{diag}(\lambda_1 \ldots, \lambda_p)$.

Lemma 2 *Let $A, L \in \mathbb{R}^{n,n}$ with A symmetric positive definite, L nonsingular, $E = I - L^\top AL$ positive semidefinite and let $p < n$.*

1. The minimal $\kappa^{(1)}$ in (9) is obtained with $P \in \mathbb{R}^{n,p}$, $Z \in \mathbb{R}^{p,p}$ defined via

$$P = L[v_1, \ldots, v_p] \in \mathbb{R}^{n,p},\ Z = P^\top AP\left(I - P^\top AP\right)^{-1} \in \mathbb{R}^{p,p}. \tag{13}$$

In this case we have $\mu = 1 - \lambda_{p+1}, \kappa^{(1)} = \frac{1-\lambda_n}{1-\lambda_{p+1}}$.

2. For P from (13) and

$$\hat{Z} = P^\top AP \tag{14}$$

we have

$$\gamma M^{(1)} \preceq LL^\top + P\hat{Z}^{-1}P^\top \preceq \Gamma M^{(1)}, \tag{15}$$

where $\gamma = 2 - \lambda_1 \geqslant 1, \Gamma = 2 - \lambda_p \leqslant 2$.

3. The matrices P from (13) and $\hat{Z}$ from (14) yield the minimal $\kappa^{(2)}$ in (11) with $\mu = 1 - \lambda_{p+1}^2, \kappa^{(2)} = \frac{1-\lambda_n^2}{1-\lambda_{p+1}^2}$.

Proof. 1. For P, Z as in (13) we have

$$Z = (I - E)E^{-1} = (I - \Lambda_p)\Lambda_p^{-1}$$

and condition (9) is equivalent to

$$\mu(I-E)^{-1} \preceq I + \Psi_p \Lambda_p (I-\Lambda_p)^{-1} \Psi_p^\top \preceq \mu\kappa^{(1)}(I-E)^{-1}. \tag{16}$$

Multiplying with $V^\top$ from the left and V from the right we obtain an inequality for diagonal matrices as

$$\mu \begin{bmatrix} \frac{1}{1-\lambda_1} & & \\ & \ddots & \\ & & \frac{1}{1-\lambda_n} \end{bmatrix} \preceq \begin{bmatrix} \frac{1}{1-\lambda_1} & & & & & \\ & \ddots & & & & \\ & & \frac{1}{1-\lambda_p} & & & \\ & & & 1 & & \\ & & & & \ddots & \\ & & & & & 1 \end{bmatrix} \preceq \mu\kappa^{(1)} \begin{bmatrix} \frac{1}{1-\lambda_1} & & \\ & \ddots & \\ & & \frac{1}{1-\lambda_n} \end{bmatrix}$$

and for $\mu = 1-\lambda_{p+1}, \kappa^{(1)} = \frac{1-\lambda_n}{1-\lambda_{p+1}}$ these inequalities are satisfied. The optimality of $\kappa^{(1)}$ in (16) follows directly from the Courant–Fischer min–max characterization [16], which implies that $\mu \leqslant 1-\lambda_{p+1}$ and $\mu\kappa^{(1)} \geqslant 1-\lambda_n$. Thus the choice of $\kappa^{(1)}$ is optimal and with P, Z we obtain the optimal $\kappa^{(1)}$.

2. For $\hat{Z}$ as in (14), we note that we have $\lambda_i \in [0,1)$ and therefore inequalities (15) immediately follow.

3. For $M^{(2)}$ we proceed analogously. The desired inequality has the form

$$\mu(I-E)^{-1} \preceq I + E + E\Psi_p (I-\Lambda_p)^{-1} \Psi_p^\top E \preceq \mu\kappa^{(2)}(I-E)^{-1}. \tag{17}$$

Multiplying with Ψ from the right and its transpose from the left, we obtain that

$$\Psi(I + E + E\Psi_p(I-\Lambda_p)^{-1}\Psi_p^\top E)\Psi^\top = \operatorname{diag}\left(\frac{1}{1-\lambda_1}, \ldots, \frac{1}{1-\lambda_p}, 1+\lambda_{p+1}, \ldots, 1+\lambda_n\right)$$

and the optimal choices are clearly $\mu = 1-\lambda_{p+1}^2$ and $\mu\kappa^{(2)} = 1-\lambda_n^2$. □

A similar result for $M^{(1)}$ was obtained in [29]. Note that the optimal choice $M^{(1)}$ can be viewed as approximation to A^{-1} of *first order*, since $\kappa^{(1)} \approx 1/(1-\lambda_{p+1}^1)$, while $M^{(2)}$ is an approximation of *second order*, since $\kappa^{(2)} \approx 1/(1-\lambda_{p+1}^2)$.

Lemma 2 shows how the optimal choices for P, Z may be computed. But in practice we usually cannot determine these optimal choices, since the spectral decomposition is not available and even if it were available, then it would be very expensive to apply, since the matrix P would be a full matrix. Instead we would like to determine P, Z (or $P, \hat{Z}$) that are inexpensive to apply and still produce good approximation properties in $M^{(1)}$ ($M^{(2)}$). By the results of Lemma 2 it seems natural to set $Z = P^\top AP$ or to choose Z such that

$$\gamma Z \preceq P^\top AP \preceq \Gamma Z.$$

An inequality of this form is also useful if we intend to recursively repeat the technique in a multilevel way. To do this we replace in

$$LL^\top + P(P^\top AP)^{-1}P^\top \tag{18}$$

the term $(P^\top AP)^{-1}$ by an additive approximation $L_1L_1^\top + P_1(P_1^\top P^\top APP_1)^{-1}P_1^\top$. For the construction of $M^{(2)}$ the procedure is analogous. Recursively applied, this idea leads to the following algebraic multilevel scheme.

Let $A \in \mathbb{R}^{n,n}$ be symmetric positive definite and let $n = n_l > n_{l-1} > \cdots > n_0 > 0$ be integers. For chosen full rank matrices $P_k \in \mathbb{R}^{n_k, n_{k-1}}$, $k = l, l-1, \ldots, 1$, define A_k via

$$A_k = \begin{cases} A & k = l, \\ P_{k+1}^\top A_{k+1} P_{k+1} & k = l-1, l-2, \ldots, 1. \end{cases}$$

Choosing a nonsingular matrix $L_k \in \mathbb{R}^{n_k, n_k}$ such that $L_k L_k^\top \approx A_k^{-1}$, $k = 0, \ldots, l$ then *multilevel sparse approximate inverse preconditioners* $M_l^{(1)}, M_l^{(2)}$ are recursively defined via

$$M_k^{(1)} = \begin{cases} A_0^{-1} & k = 0, \\ L_k L_k^\top + P_k M_{k-1}^{(1)} P_k^\top & k = 1, \ldots, l, \end{cases} \tag{19}$$

and

$$M_k^{(2)} = \begin{cases} A_0^{-1} & k = 0, \\ L_k(2I - L_k^\top A_k L_k) L_k^\top + (I - L_k L_k^\top A_k) P_k M_{k-1}^{(2)} P_k^\top (I - A_k L_k L_k^\top) & k = 1, 2, \ldots, l, \end{cases} \tag{20}$$

respectively.

For $l = 1$ we obviously obtain the operators $M^{(1)}$ and $M^{(2)}$ in (8) and (10), respectively.

If we exactly decompose the matrix on the coarsest level, i.e., $A_0^{-1} = L_0 L_0^\top$, for example by the Cholesky decomposition and set $\Pi_k = P_l P_{l-1} \cdots P_{k+1}$, then we can rewrite $M_l^{(1)}$ as

$$M_l^{(1)} = \sum_{k=0}^{l} \Pi_k L_k L_k^\top \Pi_k^\top. \tag{21}$$

For $M_l^{(2)}$ one obtains that

$$I - M_l^{(2)} A = (I - \Pi_l L_l L_l^\top \Pi_l^\top A) \cdots (I - \Pi_0 L_0 L_0^\top \Pi_0^\top A) \cdots (I - \Pi_l L_l L_l^\top \Pi_l^\top A). \tag{22}$$

We see from (21), (22) that $M_l^{(1)}$ can be viewed as *additive multilevel method*, since all the projections Π_k are formally performed simultaneously, while $M_l^{(2)}$ can be viewed as *multiplicative multilevel method*, since the projections Π_k are performed successively. In the sequel we also refer to $M_l^{(1)}$ as additive algebraic multilevel preconditioner and to $M_l^{(2)}$ as multiplicative algebraic multilevel preconditioner.

The operator $M_l^{(2)}$ is immediately derived from V–cycle multigrid methods in the numerical solution of partial differential equations. A special case for the operator $M_l^{(1)}$ is that $L_k L_k^\top = \frac{1}{\alpha_k} I$ is a multiple of the identity. In this case for $E = I - \alpha_k A_k$, the choice of some columns of E can be expressed as applying a permutation $\Phi_k \in \mathbb{R}^{n_k, n_{k-1}}$ to E, i.e. $P_k = (I - \alpha_k A_k)\Phi_k$. In this case $M_l^{(1)}$ reduces to

$$M_l^{(1)} = \frac{1}{\alpha_l}(I + \alpha_l P_l M_{l-1} P_l^\top) = \frac{1}{\alpha_l}\left(I + \frac{\alpha_l}{\alpha_{l-1}} P_l \left(I + \alpha_{l-1} P_{l-1} M_{l-2} P_{l-1}^\top\right) P_l^\top\right) = \cdots,$$

where the dots indicate that M_{l-2} has to successively substituted in a similar way. For operators of this form in [22] optimal choices for α_k have been discussed according to a wisely a priori chosen permutation matrix Φ_k. Such operators have also been studied in detail in [2, 3].

3 Approximation Properties

In this section we discuss the approximation properties of $M^{(1)}, M^{(2)}$ from (8),(10) for the case $l = 1$ and later for arbitrary $l \geqslant 1$.

For given Z, P we compare the approximation properties of $M^{(1)}, M^{(2)}$ in (9), (11) with the optimal choices in Lemma 2. For this we use the following theorem.

Theorem 3 ([20]) *Consider a symmetric positive definite matrix $M \in \mathbb{R}^{n,n}$ and matrices $P_k \in \mathbb{R}^{n,n_k}$ with $\operatorname{rank} P_k = n_k$ for $k = 1, \ldots, l$ and $\operatorname{rank} [P_1, \ldots, P_l] = n$. Consider, furthermore, positive definite matrices $B_k \in \mathbb{R}^{n_k,n_k}$ and*

$$M_S^{-1} := \sum_{k=1}^{l} P_k B_k^{-1} P_k^\top. \tag{23}$$

If $K > 0$ is a constant, such that for every $x \in \mathbb{R}^n$ there exists a decomposition $x = \sum_{k=1}^{l} P_k x_k$ satisfying

$$\sum_{k=1}^{l} x_k^\top B_k x_k \preceq K x^\top M x, \tag{24}$$

then $M_S \preceq KM$.

Applying this Theorem we can prove the following result.

Theorem 4 *Let $A \in \mathbb{R}^{n,n}$ be symmetric positive definite and let $L \in \mathbb{R}^{n,n}$ be nonsingular such that $M = L^\top A L \preceq I$. Set $E = I - M$ and $P = LV$, where $V \in \mathbb{R}^{n,p}$ has $\operatorname{rank} V = p$ and let $W \in \mathbb{R}^{n,n-p}$ be such that $\operatorname{rank} W = n - p$ and $W^\top M V = 0$. Finally let $Z \in \mathbb{R}^{p,p}$ be symmetric positive definite such that*

$$\gamma P^\top A P \preceq Z \preceq \Gamma P^\top A P \tag{25}$$

with positive constants γ, Γ.

1. If

$$W^\top W \preceq \Delta\ W^\top M W, \tag{26}$$

for some positive constant Δ, then for the matrix $M^{(1)}$ in (8) we have

$$\frac{\gamma}{\gamma + 1} A \preceq \left(M^{(1)}\right)^{-1} \preceq \max\{\Gamma, \Delta\} A. \tag{27}$$

2. If in (25) $\gamma \geqslant 1$ and

$$\begin{bmatrix} 0 & 0 \\ 0 & W^\top M W \end{bmatrix} \preceq \Delta\ [V, W]^\top (M - EME) [V, W], \tag{28}$$

for some positive constant Δ, then for the matrix $M^{(2)}$ in (10) we have

$$A \preceq \left(M^{(2)}\right)^{-1} \preceq \max\{\Gamma, \Delta\} A. \tag{29}$$

Proof. 1. We apply Theorem 3 to the matrices M, $B_1 = I$, $B_2 = Z$, $P_1 = I, P_2 = L^{-1}P = V$. Set $\Pi = P_2(P_2^\top M P_2)^{-1}P_2^\top M$ and $\Omega = I - \Pi$. Since $\Pi^\top M(I - \Pi) = 0$, we have $\Omega = W(W^\top MW)^{-1}W^\top M$. It follows that every $x \in \mathbb{R}^n$ can be written as

$$x = \underbrace{(I - \Pi)x}_{P_1 x_1} + \underbrace{\Pi x}_{P_2 x_2} = P_1 x_1 + P_2 x_2,$$

where $x_2 = (P_2^\top P_2)^{-1}P_2^\top x$ and $x_1 = \Omega x$. By Theorem 3 it suffices to find a constant $K > 0$ such that

$$x_1^\top x_1 + x_2^\top Z x_2 \leqslant K x^\top M x.$$

From (25) it follows that

$$\Omega^\top \Omega \preceq \Delta\ \Omega^\top M \Omega.$$

Substituting the representations of x_1, x_2 we obtain

$$\begin{aligned} x_1^\top x_1 + x_2^\top Z x_2 &= x^\top \Omega^\top \Omega x + x_2^\top Z x_2 \\ &\leqslant \max\{\Gamma, \Delta\}(x^\top \Omega^\top M \Omega x + x_2^\top (P_2^\top M P_2) x_2) \\ &= \max\{\Gamma, \Delta\}(x^\top \Omega^\top M \Omega x + x^\top \Pi^\top M \Pi x) \\ &= \max\{\Gamma, \Delta\} x^\top (\Omega + \Pi)^\top M (\Omega + \Pi) x \\ &= \max\{\Gamma, \Delta\} x^\top M x. \end{aligned}$$

Thus we have $K = \max\{\Gamma, \Delta\}$ in Theorem 3.

For the other inequality we obtain from

$$M + M^{1/2} P_2 Z^{-1} P_2 M^{1/2} \preceq M + \frac{1}{\gamma} M^{1/2} P_2 (P_2^\top M P_2)^{-1} P_2^\top M^{1/2} \preceq M + \frac{1}{\gamma} I$$

that

$$I + P_2 Z^{-1} P_2 \preceq I + \frac{1}{\gamma} M^{-1} \preceq (1 + \frac{1}{\gamma}) M^{-1}.$$

Hence we get

$$M^{(1)} = L L^\top + P Z^{-1} P^\top \preceq (1 + \frac{1}{\gamma}) A^{-1}.$$

2. To derive the inequalities for $M^{(2)}$ we multiply $M^{(2)}$ by $M^{1/2}L^{-1}$ from the left and its transpose from the right. We obtain

$$\begin{aligned} M^{1/2} L^{-1} M^{(2)} L^{-\top} M^{1/2} &= 2M - M^2 + E M^{1/2} V Z^{-1} (M^{1/2} V)^\top E \\ &= I - E \left(I - (M^{1/2} V) Z^{-1} (M^{1/2} V)^\top \right) E. \end{aligned}$$

Setting $\hat{V} = M^{1/2} V$, $T = I - \hat{V}(\hat{V}^\top \hat{V})^{-1} \hat{V}^\top$ and $\tilde{T} = I - \hat{V} Z^{-1} \hat{V}^\top$, it follows that $P^\top A P = \hat{V}^\top \hat{V}$ and

$$\begin{aligned} M^{1/2} L^{-1} M^{(2)} L^{-\top} M^{1/2} &= I - E \tilde{T} E \\ &\preceq I - E \left((1 - \frac{1}{\gamma}) I + \frac{1}{\gamma} T \right) E \\ &\preceq I - \left(1 - \frac{1}{\gamma} \right) E^2. \end{aligned}$$

If $\gamma \geq 1$, then the last term is bounded by I, otherwise the bound will be $\frac{1}{\gamma}$ and hence it follows that

$$(M^{(2)})^{-1} \succeq \min\{\gamma, 1\} A.$$

For the other direction we can adapt the proof of Theorem 3.1 in [32]. We have to estimate $E\tilde{T}E$ by a multiple of the identity from above. Note that since $W^\top M^{1/2}\hat{V} = W^\top MV = 0$, inequality (28) is equivalent to

$$M^{1/2}TM^{1/2} \preceq \Delta\ (M - EME)$$

or

$$E^2 \preceq I - \frac{1}{\Delta}T.$$

Observe that $E\tilde{T}E \preceq \beta I$ if and only if $\tilde{T}^{1/2}E^2\tilde{T}^{1/2} \preceq \beta I$ and hence, since $\gamma \geqslant 1$, we have that $\tilde{T}^{1/2}$ exists and it follows that

$$\tilde{T} = T + \hat{V}\left((\hat{V}^\top\hat{V})^{-1} - Z^{-1}\right)\hat{V}^\top \preceq T + \left(1 - \frac{1}{\Gamma}\right)\hat{V}(\hat{V}^\top\hat{V})^{-1}\hat{V}^\top.$$

Since $\tilde{T}T = T = T\tilde{T}$ we obtain

$$\begin{aligned}
\tilde{T}^{1/2}E^2\tilde{T}^{1/2} &\preceq \tilde{T} - \frac{1}{\Delta}\tilde{T}^{1/2}T\tilde{T}^{1/2} \\
&= \tilde{T} - \frac{1}{\Delta}T \\
&\preceq (1 - \frac{1}{\Delta})T + \left(1 - \frac{1}{\Gamma}\right)\hat{V}(\hat{V}^\top\hat{V})^{-1}\hat{V}^\top \\
&\preceq \max\{1 - \frac{1}{\Delta}, 1 - \frac{1}{\Gamma}\}\left(T + \hat{V}(\hat{V}^\top\hat{V})^{-1}\hat{V}^\top\right) \\
&= \max\{1 - \frac{1}{\Delta}, 1 - \frac{1}{\Gamma}\}I.
\end{aligned}$$

From this we finally obtain that

$$(M^{(2)})^{-1} = L^{-\top}M^{1/2}(I - E\tilde{T}E)^{-1}M^{1/2}L^{-1} \preceq \max\{\Delta, \Gamma\}L^{-\top}ML^{-1} = \max\{\Delta, \Gamma\}A.$$

□

For the operator $M^{(1)}$ the condition number of $M^{(1)}A$ may also be estimated in terms of the angle between the invariant subspaces associated with the p smallest eigenvalues of M and V. We refer to [29] for this approach. Note that in (26), (28) we always have $\Delta \geqslant 1$, since $M \preceq I$. Thus if we set $Z = P^\top AP$ in Theorem 4, then $\gamma = \Gamma = 1$ and the bounds for $M^{(1)}$ are determined by Δ only. Via (26) we see that the inequality for M is only needed on the subspace W which is the M–orthogonal complement of span V. Especially for the choice P in Lemma 2 it is easy to verify that $\Delta = \frac{1}{1-\lambda_{p+1}}$. Thus we obtain a condition number $\kappa^{(1)} = \frac{2}{1-\lambda_{p+1}}$ in Theorem 4, which is only slightly worse than the optimal condition number obtained via Lemma 2, which would give $\kappa^{(1)} = \frac{(1-\lambda_n)(2-\lambda_p)}{(1-\lambda_{p+1})(2-\lambda_1)}$. In a similar way we can compare the bound for $M^{(2)}$ obtained by Theorem 4 with the result of Lemma 2. In this case we obtain $\Delta = \frac{1}{1-\lambda_{p+1}^2}$ and thus $\kappa^{(2)} = \frac{1}{1-\lambda_{p+1}^2}$. Again this is almost the bound of Lemma 2, which would give $\kappa^{(2)} = \frac{1-\lambda_n^2}{1-\lambda_{p+1}^2}$. In this respect, the bounds in Theorem 4 are (almost) as sharp as the optimal bounds in Lemma 2. In contrast to Lemma 2 , Theorem 4 can be applied to any prescribed choice of P that has full rank!

Our next theorem extends Theorem 4 to the case $l \geqslant 1$.

Theorem 5 *Let $A \in \mathbb{R}^{n,n}$ be symmetric positive definite and consider the algebraic multilevel operators $M_l^{(1)}, M_l^{(2)}$ in (19) and (20), respectively. Suppose that the matrices L_k are chosen such that $M_k = L_k^\top A L_k \preceq I$ for all $k = 1, \ldots, l$. Set $E_k = I - M_k$, $P_k = L_k V_k$ and let $W_k \in \mathbb{R}^{n_k, n_k - n_{k-1}}$ be such that $\operatorname{rank} W_k = n_k - n_{k-1}$ and $W_k^\top M_k V_k = 0$, for all $k = 1, \ldots, l$.*

1. If Δ is a constant such that

$$W_k^\top W_k \preceq \Delta\ W_k^\top M_k W_k, \tag{30}$$

for all $k = 1, \ldots, l$, then we have

$$\frac{1}{l+1} A \preceq \left(M_l^{(1)}\right)^{-1} \preceq \Delta A. \tag{31}$$

2. If Δ is a constant such that

$$\begin{bmatrix} 0 & 0 \\ 0 & W_k^\top M_k W_k \end{bmatrix} \preceq \Delta\ \left[V_k, W_k\right]^\top \left(M_k - E_k M_k E_k\right) \left[V_k, W_k\right], \tag{32}$$

for all $k = 1, \ldots, l$, then we have

$$A \preceq \left(M_l^{(2)}\right)^{-1} \preceq \Delta\ A. \tag{33}$$

Proof. We proceed by induction on l. For $l = 1$ the assertion follows by Theorem 4 applied to $Z = P^\top A P$. If we apply Theorem 4 to $A_{l-1}, M_{l-1}^{(1)}$, i.e., let Δ be a constant such that

$$\frac{1}{l} A_{l-1} \preceq (M_{l-1}^{(1)})^{-1} \preceq \Delta\ A_{l-1},$$

then, with $Z = (M_{l-1}^{(1)})^{-1}$, we obtain $\gamma = \frac{1}{l}, \Gamma = \Delta$. But $\frac{\gamma}{1+\gamma} = \frac{1}{l+1}$ and hence (31) follows. Inequality (33) follows analogously. □

By Theorem 5 we only loose a factor $\frac{1}{l+1}$ in the condition number by using $l+1$ levels compared with the case $l = 1$ (exact 2–level method). If the reduction in size of A_k in every step is sufficient, i.e., for example if the size of A_{k-1} is half the size of A_k or less, then we need at most $l \leqslant \log_2(n)$ levels. In this case the factor $1/(l+1) \approx 1/\log_2(n)$ is (almost) neglectible.

For the multilevel method we still need a method for the construction of a well–suited matrix P_k in each step. This will be the topic of the next section.

4 The Coarsening Process

So far we have not discussed the construction of the coarse grid projection matrix P for given L, A. As before we set $L^\top A L = M$, $E = I - M$ and assume that $E \succeq 0$.

4.1 Construction of P via the QR–Decomposition

We have already seen in Lemma 2, that in terms of conditioning an invariant subspace V of E associated with the large eigenvalues of E yields the optimal choice for $P = LV$. But

in practice we neither have this invariant subspace available nor is this a favorable choice, since in this case P would typically be full and a further coarsening of $P^\top AP$ will be almost impossible, since this matrix is no longer sparse. So we need a different choice for $P = LV$.
By Lemma 1 we may use a suitably chosen set of columns of E as V to approximate the space spanned by the eigenvectors associated with the large eigenvalues. But Lemma 1 does not give bounds on the preconditioning property of the resulting preconditioner.
On the other hand the approximation results from Section 3 and especially (28) show that choosing a suitable space V will give the desired approximation properties.
To find this suitable space V, we need to establish the connection between the approximation results and Lemmas 1 and 2. According to the proof of Lemma 1 we need a QR–like decomposition $M^{1/2}E = QR$ (or more precisely of $M^{1/2}\hat{E}_p = QR$) if we want to approximate the eigenvectors associated with the large eigenvalues. Equivalently we can compute $E = QR$, where $Q^\top MQ = I$. So if V, satisfying (28), arises from a QR decomposition of E with $Q^\top MQ = I$, then Lemma 1 is applicable. In other words this choice of V should ensure that E is well approximated by a rank–p matrix up to a small error. Lemma 6 gives precisely this connection.

Lemma 6 *Let $M \in \mathbb{R}^{n,n}$ be symmetric positive definite and let $E = I - M$. Suppose that we have a decomposition*

$$E \underbrace{[\Pi_1, \Pi_2]}_{\Pi} = \underbrace{[V, W]}_{Q} \underbrace{\begin{bmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{bmatrix}}_{R}, \tag{34}$$

where Π is a permutation matrix, $Q = [V, W]$ is nonsingular and $V^\top MW = 0$. Then there exist matrices R, F such that

$$E = M^{1/2}(E\Pi_1)R + F. \tag{35}$$

If there exists a constant Δ that satisfies (28), then $\|F\|_2^2 \leqslant 1 - \frac{1}{\Delta}$.

Proof. Since $[V, W]$ is nonsingular and $W^\top MV = 0$, we have

$$I = M^{1/2}V(V^\top MV)^{-1}V^\top M^{1/2} + M^{1/2}W(W^\top MW)^{-1}W^\top M^{1/2}.$$

With $R = R_{11}^{-1}(V^\top MV)^{-1}V^\top M^{1/2}E$ we have

$$\begin{aligned} F &\equiv E - M^{1/2}(E\Pi_1)R \\ &= E - M^{1/2}VR_{11}R \\ &= E - M^{1/2}V(V^\top MV)^{-1}V^\top M^{1/2}E \\ &= M^{1/2}W(W^\top MW)^{-1}W^\top M^{1/2}E \end{aligned}$$

and it follows that

$$\begin{aligned} \|F\|_2^2 &= \|(W^\top MW)^{-1/2}W^\top M^{1/2}E\|_2^2 \\ &= \sup_{x \neq 0} \frac{x^\top W^\top EMEWx}{x^\top W^\top MWx} \\ &= 1 - \sup_{x \neq 0} \frac{x^\top W^\top (M - EME)Wx}{x^\top W^\top MWx} \\ &\leqslant 1 - \frac{1}{\Delta}. \end{aligned}$$

□

Lemma 6 shows that if V satisfies (28) with a small Δ, then by Lemma 1 the spaces spanned by the columns of $M^{1/2}V$ and those of $M^{1/2}E\Pi_1$ are good approximations to the invariant subspace of E associated with the p largest eigenvalues.

As a consequence of Lemma 6 we may use a QR–decomposition with column pivoting of E,

$$E[\Pi_1, \Pi_2] = [V, W]R,\ V^\top MW = 0 \tag{36}$$

to obtain a projection matrix $P = LE\Pi_1 = LVR_{11}^{-1}$ such that the remaining error matrix F has small norm. Clearly there is no restriction in replacing V by $E\Pi_1$, since by $V = E\Pi_1 R_{11}^{-1}$ both sets of columns span the same space. But the preconditioners $M^{(1)}, M^{(2)}$ do not change when replacing V by VR_{11}. In contrast to V, $E\Pi_1$ is typically sparse. Moreover, we can determine $P = LE\Pi_1$ as coarse grid projection matrix from the QR–decomposition (36) for which the bounds of Lemma 4 hold. Here the columns of V, W are not required to be orthogonal in the standard inner product as one typically requires in a QR–decomposition, see e.g. [16, 34], but they are orthogonal with respect to the inner product defined by M. We will not discuss in detail how to compute an approximate QR–decomposition. One possibility is to adapt a QR–like decomposition as in [34] but other constructions are possible as well. See [8] for a detailed description of this quite technical construction.

4.2 Selection of Coarse Grid Nodes

The next issue that has to be discussed is the pivoting strategy in the QR-decomposition. Clearly the best we can do is to locally maximize Δ in the inequalities (26), (28) to obtain a feasible coarse grid matrix $P = LE\Pi_1$ for the preconditioners $M^{(1)}$ in (8) and $M^{(2)}$ in (9). Since we only have the freedom to choose the permutation Π_1 in each step, we could choose p columns of E to locally optimize (26), (28). It is clear that for a fixed number of columns p there exist $\binom{n}{p}$ permutations which have to be checked and for any of these choices one has to compute a QR decomposition of an $n \times p$ matrix $E\Pi_1$ to get the corresponding Δ. Already for small p the costs are prohibitively expensive, e.g., for $p = 2$, $n(n-1)/2$ possibilities have to be checked. So in practice not more than $p = 1$ can be used in one step. Using the M-orthogonality of V, i.e., that $V^\top MV = I$, we set

$$T = I - VV^\top M. \tag{37}$$

Then it is easy to see that the M–orthogonal complement W of V is given by

$$W = TE\Pi_2. \tag{38}$$

Using T from (37), identity (26) can be written as

$$\frac{1}{\Delta} = \min_{y \neq 0} \frac{y^\top W^\top MWy}{y^\top W^\top Wy} \tag{39}$$

or equivalently as

$$\frac{1}{\Delta} = \min_{Tx \neq 0} \frac{x^\top T^\top MTx}{x^\top T^\top Tx}. \tag{40}$$

Likewise we can reformulate (28) as

$$\frac{1}{\Delta} = \min_{Tx \neq 0} \frac{x^\top (M - EME)x}{x^\top T^\top MTx}. \tag{41}$$

The minimal quotient (40) is obtained if Tx is the eigenvector associated with the smallest eigenvalue of M.

After a certain pivot index has been chosen in step p, we can compute the best pivot index from the remaining matrix using (40), (41) and get the next pivot column.

Expressions (40), (41) require the solution of an eigenvalue problem in every step. Since even for small matrices it is almost impossible to solve all the eigenvalue problems completely for any possible choice in step $p+1$, the eigenvector of M associated with the smallest eigenvalue can serve as test vector. Initially the minimum is achieved for the eigenvector associated with the smallest eigenvalue λ. Suppose that x with $x^\top x = 1$ is a normalized eigenvector of M associated to the smallest eigenvalue, say λ. Then we have

$$\begin{aligned} \hat{\lambda} &:= \frac{x^\top T^\top M T x}{x^\top T^\top T x} \\ &= \frac{x^\top (M - MVV^\top M)x}{x^\top (I - 2VV^\top M + MVV^\top VV^\top M)x} \\ &= \lambda \frac{1 - \lambda \|V^\top x\|_2^2}{1 - 2\lambda \|V^\top x\|_2^2 + \lambda^2 (x^\top V) V^\top V (V^\top x)}. \end{aligned} \tag{42}$$

If $V^\top V$ is not too big then, once a projection operator T is applied, the change in $\hat{\lambda}$ is essentially determined by the norm of $V^\top x$. Examining equation (42) we see that if $\|V^\top x\|_2$ is large, then $\hat{\lambda}$ will still be close to λ, while if $\|V^\top x\|_2$ is small then λ and $\hat{\lambda}$ will be even much closer.

We can do similar calculations for (41) and obtain

$$\hat{\lambda} = \frac{x^\top (M - EME)x}{x^\top T^\top M T x} = \frac{1 - (1-\lambda)^2}{1 - \lambda \|V^\top x\|_2^2}.$$

Here the changes are precisely driven by the angle $\|V^\top x\|_2$ independently of $V^\top V$.

This analysis justifies to replace both (40), (41) by $\|V^\top x\|_2$. In [8] approximations to x were computed using a simple heuristic approach but clearly there exist many other strategies. Let us postpone the concrete choice of a test vector that approximates the eigenvector x for a moment and let us discuss pivoting strategies based on a given angle $\|V^\top x\|_2$. A first strategy would be that, after p coarse grid nodes have been chosen, we choose the next coarse grid node such that $\|V^\top x\|_2$ is maximized for all possible T of the form

$$T = I - VV^\top M,\ V = [V_p, v_{p+1}].$$

Here V_p corresponds to the already chosen first p coarse grid nodes in the QR decomposition (34) while v_{p+1} represents column $p+1$ and we want that $[V_p, v_{p+1}]^\top M [V_p, v_{p+1}] = I$.

A second and better approach is the following block strategy. Since V spans the same space as suitably chosen columns of E, we have that two columns i, j of V or E are M–orthogonal, if their distance is larger than 3 in the graph of M. This can be seen from the fact that E, M have the same graph and $E^\top M E$ may have nonzeros elements only for pairs (i, j) that have a distance less than or equal to 3. For this reason for $k = 1, \ldots, n$ we introduce the sets

$$\mathcal{N}^t(k) = \{l :\ e_k^\top |E|^t e_l \neq 0\}. \tag{43}$$

that contain the nodes of distance t from k in the undirected graph associated with E. Since any two possible choices for v_{p+1} commute if their distance in the undirected graph of M is

larger than 3, we can choose as many new nodes in step $p+1$ as there are nodes with distance 4 or more between each other. Hence, after p coarse grid nodes have been chosen, we may choose the next coarse grid node such that $V^\top x$ is maximized for all T of the form

$$T = I - VV^\top M, V = [V_p, v_{p+1}^{(1)}].$$

Then we can continue this procedure for every node of distance larger than 3 from node $p+1$ and obtain

$$T = I - [V_p, v_{p+1}^{(1)}, v_{p+1}^{(2)}][V_p, v_{p+1}^{(1)}, v_{p+1}^{(2)}]^\top M$$

We can repeat this strategy until there exists no new nodes outside $\mathcal{N}^3(k)$ for any selected coarse grid node k. Since all these new nodes are independent of each other, eigenvalue problems (40), (41) need not be updated during this step and likewise $V^\top x$ is maximized independently.

Numerical experiments with these two strategies have shown that in practice the second strategy is preferable, since it does not run into a local but non–global optimum as often as the first strategy.

We will also introduce a locking mode. Suppose that one pass of the block strategy has determined a certain set of coarse grid nodes, while the remaining nodes so far are not considered, since they are within a distance of 3 to one of the members of the set. Let us omit indices for a moment and set $T = I - VV^\top M$. Suppose that in step $p+1$, the index l is chosen as coarse grid node in the second strategy. For all neighbouring nodes k we can compute the arithmetic mean of $(v_k^\top x)^2$. Then we lock all those nodes m for which the value $(v_k^\top x)^2$ is smaller than the arithmetic mean, i.e. we do not consider m as coarse grid node anymore. In our experience this strategy is save when being applied a–posteriori after a set of coarse grid nodes has been determined such that all remaining nodes are within a distance of 3 to at least one coarse grid node or more. We also apply this strategy during the detection of the coarse grid nodes to all nodes within distance 3 of the recently detected coarse grid node. But in contrast to the strategy that locks nodes a–posteriori we need to be much more careful when locking nodes during the construction of coarse grid nodes. In other words we add some constraint before we lock nodes in order to make sure that we do not lock nodes that might become potential coarse grid nodes later on. For this reason we only lock those nodes which are within a distance of 3 to the coarse grid node that is currently determined and require that for any of these nodes there exists a neighbour node belonging to the coarse grid. This is much more restrictive but accelerates the process, since during the construction the number of nodes that need to be updated or that are considered as coarse grid nodes decreases significantly.

The basic form of the coarsening process then looks as follows.

Set $x^\top E = \alpha$, $\nu_i = (Ee_i)^\top MEe_i$ and $p = 0$.

while nodes available
    Choose node $p+1$ subject to maximize α_{p+1}^2/ν_{p+1} among all available nodes.
    Exclude nodes within distance 3 or less.
    Perform one step of the QR–decomposition (34).
    Replace α by $T\alpha$ and ν_i by $(TEe_i)^\top MTEe_i$.
    $p = p+1$
Lock nodes.

To perform this procedure, we have to sort the list of angles $\left(\|v_j^\top x\|_2^2\right)_j$. This could for example be done initially and then the list can be updated whenever angles change. Our experiments have shown that after a step of the QR–decomposition (full or approximate) was performed, the angles often drastically change. Although this is a local effect for the case of an approximate QR–decomposition, to update a sorted list of angles was very costly. So instead of the first step in the described procedure we take the maximum only among the nodes of $\mathcal{N}^t(i)$, where i is the coarse grid node that has been just chosen in the previous step. Since the nodes of $\mathcal{N}^t(i)$ are locked from the previous step, they cannot serve as coarse grid nodes. But what one could do is to use one node $j \in \mathcal{N}^t(i)$, that maximizes $\|v_j^\top x\|_2$. Instead of taking this node j as coarse grid node, we only simulate one step of the approximate QR decomposition and take a related unlocked node from $\mathcal{N}^t(j)$ as next coarse grid node that maximizes $\left(\|v_j^\top x\|_2^2\right)_j$. Only if $\mathcal{N}^t(j)$ consists of nothing but locked nodes or coarse grid nodes, then the step of maximizing $\|V^\top x\|_2$ is carried out. In general there are typically much more than only one node that might serve as next coarse grid node. Therefore the set of candidates is stored in a list and candidates from this list can serve as coarse grid nodes in a later steps (following the first–in first–out principle). This simplifies the detection of coarse grid nodes massively and steps that require a simulation of an additional QR step become relatively rare.
At the end of the procedure we will end up in a situation, where every node is either locked or it belongs to the coarse grid. Then we keep those nodes j locked for which $\|v_j^\top x\|_2$ was below the arithmetic mean taken over $\mathcal{N}^t(j)$. After that new unlocked nodes appear and the process to detect coarse grid nodes is repeated. We also unlock nodes j if there is either no coarse grid node in $\mathcal{N}^1(j)$ or if there is no unlocked node with a larger angle in $\mathcal{N}^1(j)$.

In every step of the procedure that determines the new coarse grid we need a step of the QR decomposition. To do this exactly would again be too expensive. In the next subsection we therefore discuss an approximate QR decomposition.

4.3 A simple approximate QR decomposition

To derive an approximate QR decomposition we have to discuss which problems occur. One problem is that a full QR decomposition will typically end up in a full matrix Q even if the original matrix is sparse. But there is a simple way to work around this large memory requirement. If a partitioned matrix $A = \begin{bmatrix} A_1 & A_2 \end{bmatrix}$ is factored as

$$A = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} \\ \mathrm{O} & R_{22} \end{bmatrix},$$

then Q_1 can be obtained from A_1 by solving a linear system with R_{11}. As long as only the last column, say column k, of Q_1 is required, then we can compute $Q_1 e_k := A_1 r$ via the solution of the linear system $R_{11} r = e_k$ and $e_k^\top R_{12}$ from $e_k^\top Q_1^\top M A_2$, see [34] for an application of this approach. Clearly $Q_1 e_k$ will still be full and the costs are increasing as k increases, since one has to solve a linear system with a $k \times k$ matrix R_{11}, but solving a linear system with R_{11} corresponds to a reorthogonalization of $A_1 e_k$ against the leading $k-1$ columns of Q_1 in the modified Gram–Schmidt process. So a natural simplification is to restrict the reorthogonalization procedure to a neighbourhood of k in the sense of the graph of A. Here the matrix for which a QR decomposition is performed is the residual matrix E and the inner product is given by the preconditioned matrix M. So a natural way to define a neighbourhood

of k is given by sparsity pattern of $E^\top ME$, i.e., we consider the nodes of $\mathcal{N}^3(k)$ and reduce R_{11} to the diagonal block associated with $\mathcal{N}^3(k)$.

Now suppose that we have generated a test vector (see subsection 4.4). Even if we can detect a reasonable set of coarse grid nodes from the approximate QR decomposition, we loose our test vector x_0 from the initial grid. Also we need a new test vector, when the coarsening process is repeated on the next coarser grid. Of coarse one can use those components of x or x_0 that are associated with the coarse grid nodes. More sensible is to modify the coarsening process such that recycling of components of the test vector is supported. In principle we should have $Mx \approx 0$, i.e. $Ex \approx x$. Suppose that $\mathcal{C}$ is the set of coarse grid nodes. We should try to modify the selection of coarse grid nodes subject to $Ex \approx \sum_{k\in\mathcal{C}} Ee_k x_k$. In this case we can recycle the test vector and use $\Big(x_k\Big)_{k\in\mathcal{C}}$ as test vector for the repeated coarsening process applied to the second grid. Since $E \preceq I$ we can add a post–processing step to the algorithm in which the condition of maximizing the angle $\|V^\top x\|_2$ by taking all nodes j such that $\|v_j^\top x\|_2 \geqslant c \max_l \|v_l^\top x\|_2$ is supplemented with additional nodes k that maximize $\|\sum_{k\in\mathcal{C}} Ee_k x_k + Ee_j x_j\|_2$. In practice we use $c = \frac{3}{4}$. To complete this post processing step the final set $\mathcal{C}$ is supplemented with additional nodes j subject to minimize $\|Ex - \rho(\sum_{k\in\mathcal{C}} Ee_k x_k + Ee_j x_j)\|_1$. Here ρ is chosen to minimize $\|Ex - \rho \sum_{k\in\mathcal{C}} Ee_k x_k\|_1$, since we typically do not obtain $\rho = 1$.

4.4 Construction of a test vector

We cannot afford to compute the exact smallest eigenvector, since this would typically be more expensive than solving the linear system. We need to find a test vector that can be easily generated. Throughout the computations we use $x_0 = (1, \ldots, 1)^\top$ for the initial matrix A and start with $x = L^{-1}x_0$ for the preconditioned system M. This test vector is known to satisfy $Ax_0 \approx 0$ in many applications which arise from partial differential equations but other choices for x_0 may also be used. To use x as test vector some more work is necessary. Small components of x may be important but they do not contribute to the measure $\|V^\top x\|_2$. This is even more serious, if x is only an approximate eigenvector and if $V^\top MV \neq I$, which is the case for an approximate QR factorization.

To make sure that the information on x is not overlayed by the approximation errors we split the approximate test vector x as

$$x = x^{(1)} + x^{(2)},$$

where $\|x^{(1)}\| \gg \|x^{(2)}\|$ and then instead of one test vector x we use the pair of normalized vectors

$$[x^{(1)}/\|x^{(1)}\|,\ x^{(2)}/\|x^{(2)}\|]$$

together as test vectors. This means that for a potential coarse grid node k, the measure $|v_k^\top x|^2$ which reflects the angle, is replaced by

$$\Big\| v_k^\top \Big[x^{(1)}/\|x^{(1)}\|,\ x^{(2)}/\|x^{(2)}\|\Big] \Big\|_2^2$$

which is the angle between v_k and the space spanned by $x^{(1)}$, $x^{(2)}$.

The same strategy is recursively applied to $x^{(2)}$. For the small contribution $x^{(2)}$ it is no longer clear, whether $\|Mx^{(2)}\| \ll \|M\| \cdot \|x^{(2)}\|$. For this reason we check for each component of $x^{(2)}$ if its sign should be changed. In principle we could simply take the large components of x as $x^{(1)}$ and the small components as $x^{(2)}$. But one has to examine the situation in more detail.

There are simple cases where small components x_j of x most likely do not contribute to Mx. This is the case if $\|Mx\| \approx \|M(x - e_j x_j)\|$. To detect these cases we compare $\|Me_j x_j\|_\infty$ with all $\|Me_k x_k\|_\infty$, $k \in \mathcal{N}^1(j)$. If

$$\|Me_j x_j\|_\infty \leqslant c \max_{k \in \mathcal{N}^1(j)} \|Me_k x_k\|_\infty, c \ll 1, \tag{44}$$

then x_j is considered to be a component of $x^{(2)}$ but not a component of $x^{(1)}$. In practice we used $c = 1/4$. Condition (44) can be viewed as small local contribution with respect to j's neighbours $\mathcal{N}^1(j)$.

Another case when we should take x_j as part of $x^{(2)}$ is when (44) is not fulfilled but

$$\|Me_j x_j\|_\infty \leqslant (1 + c) \sum_{k \in \mathcal{N}^1(j)} \|Me_k x_k\|_\infty / |\mathcal{N}^1(j)|.$$

(Here $|\mathcal{N}^1(j)|$ denotes the cardinality of $\mathcal{N}^1(j)$). This means that with respect to the average over the neighbours of j, $\|Me_j x_j\|_\infty$ is relatively large. If in this case

$$\|Me_j x_j\|_\infty \leqslant c \max_{k=1,\ldots,n} \|Me_k x_k\|_\infty,$$

then $\|Me_j x_j\|_\infty$ can be viewed as globally small contribution, but not necessarily as noise, since the neighbours k of j do not have significantly larger $\|Me_k x_k\|_\infty$ in the average.

This strategy is repeated with x replaced by $x^{(2)}$.

The strategies that we have presented so far, to split and modify the test vector x are based on examining contributions of x that may be small but become big once the parts are rescaled. The final modification of x is based on contributions that do no not immediately show up because they (almost) cancel each other. I.e. we might find proper subsets $J \subset \{1, \ldots, n\}$ such that $(m_{ij})_{i,j \in J}(x_j)_{j \in J} \approx 0$. To detect these sets we check for any $i = 1, \ldots, n$ row i of Mx. We try to detect a subset $J_0 \subset \mathcal{N}^1(i)$ such that

$$\sum_{j \in J_0} m_{ij} x_j \approx 0.$$

J_0 is constructed starting with $J_0 = \{i\}$ and adding additional nodes step by step. Additional nodes j are added if $m_{ij} x_j$ has a different sign than $m_{ii} x_i$. This is done until $|\sum_{j \in J_0} m_{ij} x_j|$ has reached its minimal value or at most a tolerance (we used $0.05\,|m_{ii}|\,\|\,(x_j)_{j \in \mathcal{N}^1(i)}\,\|_\infty$). Additional nodes j with same sign as $m_{ii} x_i$ are added, if $|\sum_{j \in J_0} m_{ij} x_j|$ can be reduced further. After J_0 has been detected, we repeat this strategy for all remaining $i \in J_0$. If new i are found with an analogous property, then J_0 is enlarged to obtain a new set J_1. It is clear that nodes which were excluded when J_0 was constructed will not be added in a later step. This limits the nodes i which might be considered in the next step. Finally this strategy yields one or more sets J.

4.5 Final Comments

Finally note that in order to approximately satisfy $M \preceq I$, we use four steps of the Lanczos method to compute an approximation to the largest eigenvalue of M.
Since the use of approximate inverses introduces entries that are small in absolute value compared with the other entries in the row, we used diagonal compensation for M for any

entry $|m_{ij}|$ that was less than $10^{-4} \cdot \max_k |m_{ik}|$. For E we also used diagonal compensation but with $5 \cdot 10^{-2}$ instead of 10^{-4}. The reason to use different tolerances is due to the fact that M with diagonal compensation should well approximate the original M, while E is only used for the coarse grid projection.
As iterative solver, cg with initial solution x_0 was used. As stopping criterion we used $\|Ax_k - b\|_2 \leqslant \sqrt{\text{eps}}\|Ax_0 - b\|_2$, where $\text{eps} = 2.2203 \cdot 10^{-16}$ denotes the machine precision.

We have described several heuristic ideas to generate the updating procedure for a given preconditioner. We have seen that this updating can be viewed as an algebraic multigrid process. In the next section we give several numerical examples and compare with other multigrid techniques.

5 Numerical results

In this section we illustrate the effectiveness of the new procedures and, in particular, our chosen heuristic approximations. Our computations were done in MATLAB 5.3 [1] on a LINUX PC with a PENTIUM III/400 processor.

In all our examples we start with a given sparse approximate inverse for the initial matrix. There are several choices that we discuss. These are (depending on the example) the classical Jacobi preconditioner, i.e., the diagonal of the matrix, a factored approximate inverse using the graph of the initial matrix (again from [26, 25]) and finally factored block Jacobi preconditioners. For this latter type of preconditioner a diagonal block is factored using the eigenvalue decomposition of the block.

We updated the preconditioner recursively and at each level we stopped the coarsening process if there were no more nodes available (because of the locking strategy). In the multigrid process we always used diagonal preconditioning on the coarser levels. We terminated the coarsening process, when at some level the reduction of the system size was not significant any more, i.e., more than 75% of the previous system. In this case the coarse grid system was solved via the Cholesky factorization.

The algebraic multilevel method based on the approximate QR–decomposition will be denoted by AMG–QR. We will denote the geometric multigrid by GMG and the algebraic multigrid from [32] will be denoted by AMG–RS.

Example 7 Our first example is the matrix LANPRO/NOS2 from the Harwell–Boeing collection. Table 1 shows the results for the QR–based AMG compared with AMG from [32]. The original system has size $n = 957$ and an average number of 4.3 nonzero entries per row. The condition number of the initial system is $5.1 \cdot 10^9$. The matrix has large positive off–diagonal entries.

Table 2 gives the results for the number of iteration steps. From Table 2 we can see that the coarse grids generated by the QR–based AMG perform very well, while in contrast to this AMG–RS constructs an unsatisfactory coarse grid hierarchy.

For a tridiagonal preconditioner obtained from a factored sparse approximate inverses in [25, 26], the results for the coarsening process as well as for the iterative process are essentially identical for all three methods.

Table 1: **NOS2, diagonal preconditioner, coarsening**

			Levels: system size and number of nonzeros (average per row)					
		flops	2	3	4	5	6	7
AMG–RS	size	$3.9 \cdot 10^5$	477	237	117	19		
	nonzeros		4.3	4.3	4.3	2.9		
AMG–QR	size	$1.4 \cdot 10^6$	477	238	114	55	22	11
	nonzeros		4.3	4.3	4.2	4.2	3.2	3.4

Table 2: **NOS2, diagonal preconditioning, iteration**

type of precond.	no prec.	dgl.	AMG–RS $M_l^{(1)}$	AMG–RS $M_l^{(2)}$	AMG–QR $M_l^{(1)}$	AMG–QR $M_l^{(2)}$
cg steps	59765	6920	4632	2103	92	39
flops	$1.1{\cdot}10^9$	$1.5{\cdot}10^8$	$1.7{\cdot}10^8$	$1.7{\cdot}10^8$	$4.0{\cdot}10^6$	$3.5{\cdot}10^6$

For the factored sparse approximate inverse from [25, 26] with the same sparsity pattern as the initial matrix the results for the coarsening process can be found in Table 3.

Table 3: **NOS2, pattern of A for preconditioning, coarsening**

			Levels: system size and number of nonzeros (average per row)			
		flops	2	3	4	5
AMG–RS	size	$5.7 \cdot 10^5$	426	212	79	13
	nonzeros		5.5	4.5	2.9	2.8
AMG–QR	size	$2.0 \cdot 10^6$	449	152	19	
	nonzeros		8.3	5.3	2.8	

Here the use of a sparse approximate inverse does not improve the coarsening process. The results are significantly worse than for the case where diagonal preconditioning is used. However, the QR–based AMG still performs much better than AMG–RS. This is no surprise, since this example has large positive off–diagonal entries which is known to cause problems for AMG-RS. The numerical results for the iterative solution are given in Table 4.

Finally we will consider a block–diagonal preconditioner. The matrix NOS2 is block tridiagonal with blocks of size 3×3. So natural block diagonal preconditioners should have block

Table 4: **NOS2, pattern of A for preconditioning, iteration**

type of precond.	no prec.	pattern of A	AMG–RS $M_l^{(1)}$	AMG–RS $M_l^{(2)}$	AMG–QR $M_l^{(1)}$	AMG–QR $M_l^{(2)}$
cg steps	59765	3360	4518	2087	1340	714
flops	$1.1{\cdot}10^9$	$9.4{\cdot}10^7$	$1.9{\cdot}10^8$	$1.9{\cdot}10^8$	$7.3{\cdot}10^7$	$7.7{\cdot}10^7$

size $3, 6, 9, \ldots$. We will use a block–Jacobi preconditioner of block size 6. Table 5 shows the results for the generation of the coarse grid hierarchy and Table 6 the numerical results.

Table 5: **NOS2, block diagonal (6×6) preconditioning, coarsening**

			Levels: system size and number of nonzeros (average per row)			
		flops	2	3	4	5
AMG–RS	size	$5.2 \cdot 10^5$	476	158	39	19
	nonzeros		4.6	3.0	2.9	2.9
AMG–QR	size	$1.7 \cdot 10^6$	323	99	36	
	nonzeros		7.0	5.6	4.5	

Again the numerical results are not as good for the diagonal case but still one can observe a smaller coarse grid hierarchy and a significantly smaller number of iteration steps for the QR–based AMG.

Table 6: **NOS2, block diagonal (6×6) preconditioning, iteration**

type of precond.	no prec.	block dgl.	AMG–RS $M_l^{(1)}$	AMG–RS $M_l^{(2)}$	AMG–QR $M_l^{(1)}$	AMG–QR $M_l^{(2)}$
cg steps	59765	5037	3517	1675	657	299
flops	$1.1{\cdot}10^9$	$1.5{\cdot}10^8$	$1.5{\cdot}10^8$	$1.6{\cdot}10^8$	$3.3{\cdot}10^7$	$2.9{\cdot}10^7$

The last two preconditioners, i.e., the block diagonal preconditioner and the factored sparse approximate inverse preconditioner with same sparsity pattern as A illustrate that even the QR–based AMG does not always construct a satisfactory grid, but it is still better than that of AMG–RS.

Example 8 Consider the problem

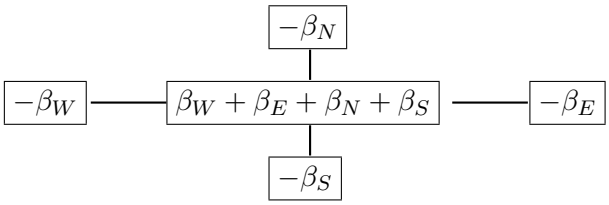
$$\begin{aligned} -\mathrm{div}\,(a\,\mathrm{grad}\,u) &= f \text{ in } [0,1]^2 \\ u &= g \text{ on } \partial[0,1]^2 \end{aligned}$$

where $a : [0,1]^2 \to \mathbb{R}$ has different weights in parts of the domain. In detail we consider in each quarter the weights

100	1
1	100

The discretization is done using a uniform grid and a five point star difference discretization. With local weights $\beta_N, \beta_W, \beta_E, \beta_S$, then the discretization is described by Figure 1.

Figure 1: **Dirichlet, 5–point difference star**



In every subdomain the value of β is identical to the weights and for nodes on the interface between the subdomain the arithmetic mean is used.

In this case we will also compare the results with those of geometric multigrid, for which the compact 7–point stencil $1/2 \begin{pmatrix} 0 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 0 \end{pmatrix}$ is used. Since for this problem the vector $x = (1, \ldots, 1)^\top$ represents the constant function, it makes sense to modify the QR–based AMG slightly. In general we have adapted the AMG such that the coarse grid projection matrix $E(:, C)$ with the set of coarse grid nodes C roughly satisfies $Ev \approx E(:, C)x(C)$. In this specific problem we may satisfy this constraint exactly by replacing $E(:, C)$ with $DE(:, C)$, where D is a diagonal scaling such that $Ev = DE(:, C)x(C)$.

We use $n = 65025$ and the initial system has on average 5 entries per row. Table 7 shows the results of the coarsening process, i.e., the size of the coarser systems and also the average amount of nonzero elements per row. Table 8 gives the number of iteration steps and flops using multigrid (geometric/algebraic) as preconditioner for cg.

In order to see how the new method scales we compare the flops for the generation of the coarsening process for $n = 961, 3969, 16129, 65025$, see Table 9.

The results so far demonstrate that AMG–QR performs well, even better than classical AMG–RS. It is better with respect to the coarsening process as well as with respect to the iterative process. One problem that can be seen from Table 9, however, is that the QR–based AMG is more expensive (a factor 3) than AMG–RS. This is no surprise, since its construction involves an approximate QR–factorization. Despite of this construction it also scales linearly.

For a sparse approximate inverse as in [26, 25] with the same sparsity pattern as the initial matrix the preconditioned system is still an M–matrix, which has been observed to be helpful for the application of the classical AMG. Table 10 shows that both methods use a much coarser grid than in the case of diagonal preconditioning. But still AMG–QR needs fewer and smaller levels. The iterative process is also faster for the QR–based AMG as shown in Table 11.

Scalability is shown in Table 12. It is interesting that for this approximate inverse the QR–based AMG performs better (less flops) than in the diagonal case, while the classical AMG

Table 7: **Dirichlet, diagonal preconditioning, coarsening**

		Levels: system size and number of nonzeros (average per row)							
	level	2	3	4	5	6	7	8	9
AMG–RS	size	32509	8756	2547	822	280	102	37	10
	nonzeros	8.9	9.7	11.2	13.9	15.9	17.2	15.5	8.2
AMG–QR	size	32509	7313	1534	463	84	12		
	nonzeros	8.9	9.7	10.6	15.6	12.5	5.5		
GMG	size	16129	3969	961	225	49	9	1	
	nonzeros	5.0	4.9	4.9	4.7	4.4	3.7	1.0	

Table 8: **Dirichlet, diagonal preconditioning, iteration**

type of precond.	no prec.	dgl.	AMG–RS $M_l^{(1)}$	AMG–RS $M_l^{(2)}$	AMG–QR $M_l^{(1)}$	AMG–QR $M_l^{(2)}$	GMG $M_l^{(1)}$	GMG $M_l^{(2)}$
cg steps	5129	862	84	26	61	24	42	16
flops	$6.7{\cdot}10^9$	$1.3{\cdot}10^9$	$2.4{\cdot}10^8$	$1.8{\cdot}10^8$	$1.7{\cdot}10^8$	$1.6{\cdot}10^8$	$1.3{\cdot}10^8$	$1.0{\cdot}10^8$

Table 9: **Dirichlet, diagonal preconditioning, scalability**

size	961	3969	16129	65025
	flops for the coarse grid generation			
AMG–RS	$6.0{\cdot}10^5$	$2.6{\cdot}10^6$	$1.0{\cdot}10^7$	$4.2{\cdot}10^7$
AMG–QR	$1.6{\cdot}10^6$	$7.1{\cdot}10^6$	$3.0{\cdot}10^7$	$1.3{\cdot}10^8$
	flops for the iteration (using prec. $M^{(2)}$)			
AMG–RS	$1.9{\cdot}10^6$	$8.8{\cdot}10^6$	$3.9{\cdot}10^7$	$1.8{\cdot}10^8$
AMG–QR	$1.1{\cdot}10^6$	$5.9{\cdot}10^6$	$3.1{\cdot}10^7$	$1.6{\cdot}10^8$

Table 10: **Dirichlet, sparsity of A for preconditioning, coarsening**

		Levels: system size and number of nonz. (average per row)					
	level	2	3	4	5	6	7
AMG–RS	size	14386	7249	2241	466	98	10
	nonzeros	13.4	20.8	22.0	15.1	9.4	2.6
AMG–QR	size	9010	1737	338	72	13	
	nonzeros	13.6	12.7	11.2	10.7	6.4	

Table 11: **Dirichlet, sparsity of A for preconditioning, iteration**

type of precond.	no prec.	pattern of A	AMG–RS $M_l^{(1)}$	AMG–RS $M_l^{(2)}$	AMG–QR $M_l^{(1)}$	AMG–QR $M_l^{(2)}$
cg steps	5129	436	210	66	64	25
flops	$6.7{\cdot}10^9$	$9.1{\cdot}10^8$	$6.6{\cdot}10^8$	$4.8{\cdot}10^8$	$2.0{\cdot}10^8$	$1.6{\cdot}10^8$

becomes slower. The overhead in the construction is now more than compensated by the accelerated iterative part. Again both methods scale linearly with respect to the coarsening process, but AMG–QR is much faster and scales much better in the iterative part.

Finally we use a block diagonal preconditioner with small blocks. For a block diagonal matrix where each diagonal block has size 4×4 we see in Table 13 that the coarse grid generation for the QR–based AMG is much superior to AMG–RS. Here it is important to note that due to the use of block diagonal approximate inverses, the preconditioned system has many positive off–diagonal entries which causes problem for the classical AMG. But the QR–based AMGs can exploit the benefits of the sparse approximate inverses to construct only a few small coarser grids. The number of iteration steps here is not so much different between both AMG methods as shown in Table 14. The construction of much smaller grids for AMG–QR is reflected by a much faster coarse grid generation and a significant acceleration when applying the preconditioner in the iteration process. For the coarse grid generation this can be seen from the surprisingly small difference between the number of flops needed by both AMGs. For the iterative part one can observe that AMG–QR needs less flops although it requires more iteration steps.

The numerical results for this problem show that the QR–based AMG better adapts to the given initial sparse approximate inverse. This should be the case because they have been constructed to do so. The drawback is that this approach consumes more time for its construction because of using an approximate QR factorization. However AMG–QR scales as good as AMG–RS.

Example 9 Finally consider the problem

$$-\varepsilon^2 u_{xx} - u_{yy} \quad = \quad f \text{ in } [0,1]^2$$

Table 12: **Dirichlet, sparsity of A for preconditioning, scalability**

size	961	3969	16129	65025
	flops for the coarse grid generation			
AMG–RS	$1.2{\cdot}10^6$	$4.6{\cdot}10^6$	$1.8{\cdot}10^7$	$7.2{\cdot}10^7$
AMG–QR	$3.0{\cdot}10^6$	$1.4{\cdot}10^7$	$5.9{\cdot}10^7$	$2.4{\cdot}10^8$
	flops for the iteration (using prec. $M^{(2)}$)			
AMG–RS	$1.8{\cdot}10^6$	$9.4{\cdot}10^6$	$6.3{\cdot}10^7$	$4.8{\cdot}10^8$
AMG–QR	$1.1{\cdot}10^6$	$6.0{\cdot}10^6$	$2.9{\cdot}10^7$	$1.6{\cdot}10^8$

Table 13: **Dirichlet, block diagonal (4×4) preconditioning, coarsening**

		Levels: system size and number of nonzeros (average per row)							
	level	2	3	4	5	6	7	8	9
AMG–RS	size	32592	16251	5773	2185	763	274	106	33
	nonzeros	13.8	17.7	20.7	27.7	25.8	24.7	24.7	16.8
AMG–QR	size	8142	1824	382	75	17			
	nonzeros	9.1	9.7	9.4	8.2	5.6			

Table 14: **Dirichlet, block diagonal (4×4) preconditioning, iteration**

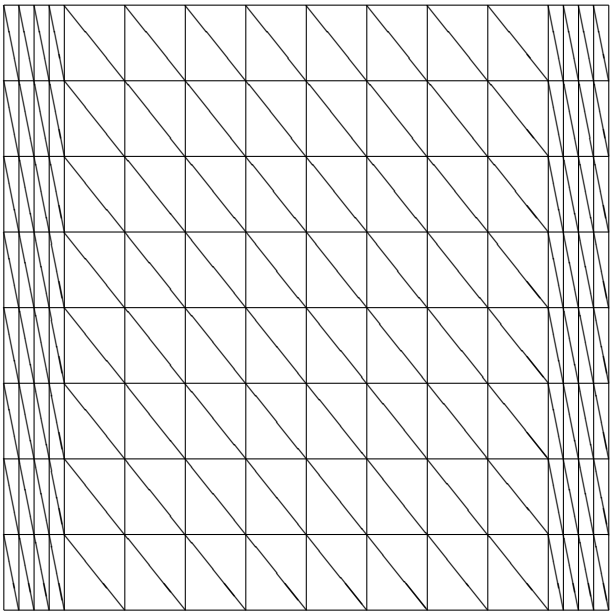
type of precond.	no prec.	pattern of A	AMG–RS $M_l^{(1)}$	AMG–RS $M_l^{(2)}$	AMG–QR $M_l^{(1)}$	AMG–QR $M_l^{(2)}$
cg steps	5129	640	76	23	83	32
flops	$6.7{\cdot}10^9$	$1.5{\cdot}10^9$	$3.1{\cdot}10^8$	$2.5{\cdot}10^8$	$2.6{\cdot}10^8$	$2.0{\cdot}10^8$

Table 15: **Dirichlet, block diagonal (4×4) preconditioning, scalability**

size	961	3969	16129	65025
	flops for the coarse grid generation			
AMG–RS	$1.1{\cdot}10^6$	$5.0{\cdot}10^6$	$2.1{\cdot}10^7$	$8.4{\cdot}10^7$
AMG–QR	$1.6{\cdot}10^6$	$6.7{\cdot}10^6$	$2.8{\cdot}10^7$	$1.1{\cdot}10^8$
	flops for the iteration (using prec. $M^{(2)}$)			
AMG–RS	$2.7{\cdot}10^6$	$1.3{\cdot}10^7$	$5.6{\cdot}10^7$	$2.5{\cdot}10^8$
AMG–QR	$1.7{\cdot}10^6$	$8.0{\cdot}10^6$	$4.2{\cdot}10^7$	$2.0{\cdot}10^8$

$$u \;=\; g \text{ on } \partial[0,1]^2$$

where ε strongly varies from 10^0 to 10^{-4}. For this problem we use the variational formulation and piecewise quadratic finite elements, cf. e.g. [9]. The discretization is done using a uniform triangulation with two additional boundary layers of size $\frac{\varepsilon}{4} \times 1$ near the left and also near the right boundary (see picture below). Within these boundary layers the triangles are condensed by an additional factor $\varepsilon/4$ in x–direction.



We examine the aspect of scalability (with respect to the system size) and robustness (with respect to ε).

Table 16 shows the number of cg iteration steps for both AMGs for the case of a diagonal approximation using $M^{(2)}$ as preconditioner. The same comparison is made in Table 17 for the case of the sparse approximate inverse with the same pattern as A.

Next we examine the computational amount of work in flops.

Table 16: **Anisotropic Dirichlet, diagonal precond., cg steps using $M^{(2)}$**

		ε versus scalability			
	ε	961	3969	16129	65025
AMG–RS	10^0	31	47	89	170
	10^{-2}	42	84	174	268
	10^{-4}	38	65	135	264
AMG–QR	10^0	23	33	56	109
	10^{-2}	23	33	60	98
	10^{-4}	23	31	49	85

Table 17: **Anisotropic Dirichlet, pattern of A for precond., cg steps using $M^{(2)}$**

		ε versus scalability			
	ε	961	3969	16129	65025
AMG–RS	10^0	24	56	103	232
	10^{-2}	47	90	196	318
	10^{-4}	48	91	177	336
AMG–QR	10^0	19	24	47	61
	10^{-2}	22	38	49	84
	10^{-4}	16	31	43	60

Table 18: **Anisotropic Dirichlet, diagonal precond.,** flops **(coarsening + cg)**

		ε versus scalability		
	ε	3969	16129	65025
AMG–RS	10^0	$3.5{\cdot}10^6{+}2.5{\cdot}10^7$	$1.4{\cdot}10^7{+}2.0{\cdot}10^8$	$5.9{\cdot}10^7{+}1.5{\cdot}10^9$
	10^{-2}	$2.9{\cdot}10^6{+}4.1{\cdot}10^7$	$1.2{\cdot}10^7{+}3.5{\cdot}10^8$	$5.0{\cdot}10^7{+}2.2{\cdot}10^9$
	10^{-4}	$2.9{\cdot}10^6{+}3.2{\cdot}10^7$	$1.2{\cdot}10^7{+}2.7{\cdot}10^8$	$5.0{\cdot}10^7{+}2.2{\cdot}10^9$
AMG–QR	10^0	$1.4{\cdot}10^7{+}1.5{\cdot}10^7$	$7.1{\cdot}10^7{+}1.1{\cdot}10^8$	$4.3{\cdot}10^8{+}8.4{\cdot}10^8$
	10^{-2}	$9.8{\cdot}10^6{+}1.4{\cdot}10^7$	$5.1{\cdot}10^7{+}1.0{\cdot}10^8$	$3.3{\cdot}10^8{+}7.0{\cdot}10^8$
	10^{-4}	$9.4{\cdot}10^6{+}1.3{\cdot}10^7$	$4.9{\cdot}10^7{+}8.5{\cdot}10^7$	$3.3{\cdot}10^8{+}6.2{\cdot}10^8$

Table 19: **Anisotropic Dirichlet, pattern of A for precond.,** flops **(coarsening + cg)**

		ε versus scalability		
	ε	3969	16129	65025
AMG–RS	10^0	$6.1{\cdot}10^6{+}3.0{\cdot}10^7$	$2.5{\cdot}10^7{+}2.2{\cdot}10^8$	$1.0{\cdot}10^8{+}2.0{\cdot}10^9$
	10^{-2}	$5.4{\cdot}10^6{+}4.3{\cdot}10^7$	$2.2{\cdot}10^7{+}3.8{\cdot}10^8$	$9.0{\cdot}10^7{+}2.5{\cdot}10^9$
	10^{-4}	$5.5{\cdot}10^6{+}4.4{\cdot}10^7$	$2.2{\cdot}10^7{+}3.5{\cdot}10^8$	$9.0{\cdot}10^7{+}2.6{\cdot}10^9$
AMG–QR	10^0	$2.8{\cdot}10^7{+}9.0{\cdot}10^6$	$1.2{\cdot}10^8{+}7.2{\cdot}10^7$	$5.2{\cdot}10^8{+}3.8{\cdot}10^8$
	10^{-2}	$2.5{\cdot}10^7{+}2.0{\cdot}10^7$	$1.3{\cdot}10^8{+}1.1{\cdot}10^8$	$6.7{\cdot}10^8{+}7.3{\cdot}10^8$
	10^{-4}	$2.2{\cdot}10^7{+}1.5{\cdot}10^7$	$1.1{\cdot}10^8{+}8.7{\cdot}10^7$	$6.5{\cdot}10^8{+}5.0{\cdot}10^8$

As the number of iteration steps in Table 16 and Table 17 have indicated, the scalability of AMG–RS performs poorer with increasing system size than AMG–QR which roughly needs only half as many flops (see Table 18 and Table 19). One additional observation can be made. AMG–QR is designed as a supplement for a given sparse approximate inverse. This does not mean that it will always be able to compensate a poor smoothing property of the initial sparse approximate inverse. This can be seen when looking at the scalability of the coarse grid generation. Although AMG–QR needs more flops for the coarse grid generation when a sparse approximate inverse with same pattern as A is used compared with the diagonal approximate inverse, it scales better than in the diagonal case. Apparently the sparse approximate inverse with same pattern as A compensates the anisotropy much better than the diagonal approximate inverse and this property is detected by AMG–QR. Although this is not part of this kind of AMG, we expect an improvement if the initial sparse approximate inverse is more adapted to the anisotropic behaviour than those simple two sparse approximate inverses that were chosen in these examples.

6 Conclusions

We have derived new approaches for the construction of algebraic multilevel methods that automatically detect the coarse grid by suitably chosen columns of the residual matrix. We have presented the mathematical theory to develop optimal preconditioners. The key feature of the new approach is the choice of an effective pivoting strategy to detect the correct set of columns. The numerical examples indicate that to obtain a good choice is a challenging problem. Simple techniques like locking of some nodes or taking several nodes in one step seem to be useful. Clearly none of these strategies is successful if the sparse approximate preconditioner does not have a smoothing property, i.e., if most of the eigenvalues of the preconditioned system clustered at the large end of the spectrum. A more detailed analysis of methods to construct good pivoting strategies needs further research.

References

[1] MATLAB – The language of technical computing. The MathWorks Inc., 1996.

[2] O. Axelsson, M. Neytcheva, and B. Polman. An application of the bordering method to solve nearly singular systems. *Vestnik Moskovskogo Universiteta, Seria 15, Vychisl. Math. Cybern.*, 1:3–25, 1996.

[3] O. Axelsson, A. Padiy, and B. Polman. Generalized augmented matrix preconditioning approach and its application to iterative solution of ill-conditioned algebraic systems. Technical report, Katholieke Universiteit Nijmegen, Fakulteit der Wiskunde en Informatica, 1999.

[4] O. Axelsson and P. Vassilevski. Algebraic multilevel preconditioning methods I. *Numer. Math.*, 56:157–177, 1989.

[5] O. Axelsson and P. Vassilevski. Algebraic multilevel preconditioning methods II. *SIAM J. Numer. Anal.*, 27:1569–1590, 1990.

[6] R. E. Bank and C. Wagner. Multilevel ILU decomposition. *Numer. Math.*, to appear, 1999.

[7] M. Benzi, C. D. Meyer, and M. Tůma. A sparse approximate inverse preconditioner for the conjugate gradient method. *SIAM J. Sci. Comput.*, 17:1135–1149, 1996.

[8] M. Bollhöfer and V. Mehrmann. A new approach to algebraic multilevel methods based on sparse approximate inverses. Preprint SFB393/99–22, TU Chemnitz, Germany, Dep. of Mathematics, August 1999.

[9] D. Braess. *Finite Elements: Theory, Fast Solvers and Applications in Solid Mechanics*. Cambridge University Press, 2001.

[10] A. Brandt. Algebraic multigrid theory: the symmetric case. *Appl. Math. Comput.*, 19:23–65, 1986.

[11] T. Chan, W.-P. Tang, and W. L. Wan. Fast wavelet based sparse approximate inverse preconditioner. *BIT*, 37(3):644–660, 1997.

[12] E. Chow and Y. Saad. Approximate inverse preconditioners via sparse-sparse iterations. *SIAM J. Sci. Comput.*, 19(3):995–1023, 1998.

[13] W. Dahmen and L. Elsner. Hierarchical iteration. In W. Hackbusch, editor, *Robust Multi-grid Methods*, volume 23 of *Notes on Numerical Fluid Mechanics*. Vieweg, Braunschweig, Germany, 1988.

[14] A. V. der Ploeg, E. Botta, and F. Wubs. Nested grids ILU–decomposition (NGILU). *J. Comput. Appl. Math.*, 66:515–526, 1996.

[15] R. W. Freund, G. H. Golub, and N. M. Nachtigal. Iterative solution of linear systems. *Acta Numerica*, pages 1–44, 1992.

[16] G. H. Golub and C. F. V. Loan. *Matrix Computations*. The Johns Hopkins University Press, third edition, 1996.

[17] A. Greenbaum. *Iterative Methods for Solving Linear Systems*. Frontiers in Applied Mathematics. SIAM Publications, 1997.

[18] M. J. Grote and T. Huckle. Parallel preconditioning with sparse approximate inverses. *SIAM J. Sci. Comput.*, 18(3):838–853, 1997.

[19] W. Hackbusch. *Multigrid Methods and Applications*. Springer-Verlag, 1985.

[20] W. Hackbusch. *Iterative Solution of Large Sparse Systems of Equations*. Springer-Verlag, 1994.

[21] V. E. Henson and P. S. Vassilevski. Element-free AMG-e: General algorithms for computing interpolation weights. Technical report UCRL–VG–138290, Lawrence Livermore National Laboratory, Livermore, CA, USA, March 2000. submitted to to SIAM Journal of Scientific Computing.

[22] T. Huckle. Matrix multilevel methods and preconditioning. Technical report SFB–Bericht Nr. 342/11/98 A, Technische Universität München, Fakulktät für Informatik, 1998.

[23] T. Huckle and J. Staudacher. Matrix multilevel methods and preconditioning. *BIT*, 42(4), December 2002. to appear.

[24] J. E. Jones and P. S. Vassilevski. AMGe based on agglomeration. Technical report UCRL–JC–135441, Lawrence Livermore National Laboratory, August 1999. to appear in SIAM J. Sci. Comput.

[25] I. E. Kaporin. New convergence results and preconditioning strategies for the conjugate gradient method. *Numer. Lin. Alg. w. Appl.*, 1(2):179–210, 1994.

[26] L. Kolotilina and A. Yeremin. Factorized sparse approximate inverse preconditionings. I. Theory. *SIAM J. Matrix Anal. Appl.*, 14:45–58, 1993.

[27] Y. Notay. Optimal V-cycle algebraic multilevel preconditioner. *Numer. Lin. Alg. w. Appl.*, 5(5):441–459, 1998.

[28] Y. Notay. Using approximate inverses in algebraic multigrid methods. *Numer. Math.*, 80:397–417, 1998.

[29] A. Padiy, O. Axelsson, and B. Polman. Generalized augmented matrix preconditioning approach and its application to iterative solution of ill-conditioned algebraic systems. *SIAM J. Matrix Anal. Appl.*, 22:793–818, 2001.

[30] A. Reusken. Approximate cyclic reduction preconditioning. In W. Hackbusch and G. Wittum, editors, *Multigrid Methods 5, Proceedings of the Fifth European Multigrid Conference*, pages 243–259. Springer-Verlag, 1998.

[31] A. Reusken. On the approximate cyclic reduction preconditioner. *SIAM J. Sci. Comput.*, 21:565–590, 2000.

[32] J. Ruge and K. Stüben. Algebraic multigrid. In S. McCormick, editor, *Multigrid Methods*, pages 73–130. SIAM Publications, 1987.

[33] Y. Saad. *Iterative Methods for Sparse Linear Systems*. PWS Publishing, Boston, 1996.

[34] G. Stewart. Four algorithms for the efficient computation of truncated pivoted QR approximations to a sparse matrix. *Numer. Math.*, 83:313–323, 1999.

[35] W.-P. Tang. Towards an effective sparse approximate inverse preconditioners. *SIAM J. Matrix Anal. Appl.*, 20(4):970–986, 1999.

[36] W.-P. Tang and W. L. Wan. Sparse approximate inverse smoother for multigrid. *SIAM J. Matrix Anal. Appl.*, 21(4):1236–1252, 2000.

[37] R. S. Varga. *Matrix Iterative Analysis*. Prentice-Hall, Englewood Cliffs, New Jersey, 1962.