Adapted Sparse Approximate Inverse Smoothers in Algebraic Multilevel Methods

Matthias Bollhöfer *
Dept. of Mathematics, MA 4-5
TU Berlin

Abstract

In this paper an algebraic multilevel method is discussed that mainly focuses on the use of a sparse approximate inverse smoother. In particular strategies are presented to adapt the sparse approximate inverse smoother to a given problem.

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

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

1   Introduction

We consider the problem of solving a large sparse linear system

$$Ax = b,$$

where A is a given $n \times n$ real matrix, $b \in \mathbb{R}^n$ is a given right hand side and we are seeking a solution $x \in \mathbb{R}^n$. In particular, we assume that the matrix is sparse and usually arises from some discretization of partial differential equations. We assume that the matrix is unstructured and no further information is available except the matrix itself. Beside direct methods and black–box incomplete LU decomposition techniques, algebraic multilevel methods have become an attractive alternative for solving these kind of systems [26, 12, 15, 29, 19, 20]. Algebraic multilevel methods construct an approximation to A^{-1} that can be mainly described by the following principles.

1. The smoothing process,

2. Coarsening process,

3. The recursive application of the same principle to a smaller matrix.

In the most simple case of a linear iteration scheme [33]

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

with initial guess $x^{(0)} \in \mathbb{R}^n$, one can describe this process by the associated iteration matrix $I - MA$. In the easiest form of multilevel methods, i.e. a two–level method, this iteration matrix could be described, e.g. by

$$(1) \qquad I - MA = (I - SA)(I - PA_H^{-1}RA)(I - SA).$$

The matrix S is called the smoother, it can be applied initially or at the end (which is referred to as pre and post smoothing) and even more than one step of smoothing is quite common. P and $R^\top$ are $n \times N$ matrices with $N \ll n$ and describe the interaction between the initial system and a smaller (nonsingular) $N \times N$ coarse grid system A_H. To generalize this approach to a multilevel approach the same principle is applied to A_H. In algebraic multigrid methods (AMG) one often prescribes the choice of the smoother and major work is spent in the construction of the matrices P and $R^\top$, which is called the coarsening process. This coarsening process uses information from the graph of the matrix and the coefficients of the matrix itself. If more information is provided, e.g. information about the mesh, one can also incorporate this data into the setup of the AMG (see e.g. [32, 16, 12]). It is quite common to use $A_H = RAP$ and $R = P^\top$.

The work in this paper is motivated by an earlier paper [4] and by the class of algebraic multigrid methods that mainly construct the prolongation P and the restriction operator R from a successive sequence of approximate block Gaussian elimination steps [25, 19, 21, 9, 28]. Given a nonsingular matrix A,

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$$

with a nonsingular submatrix A_{11}, the inverse of A can be written as

$$A^{-1} = \begin{pmatrix} A_{11}^{-1} & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} -A_{11}^{-1}A_{12} \\ I \end{pmatrix} S_{22}^{-1} \begin{pmatrix} -A_{21}A_{11}^{-1} & I \end{pmatrix},$$

where $S_{22} = A_{22} - A_{21}A_{11}^{-1}A_{12}$ denotes the Schur complement. Note that if A and A_{11} are nonsingular, so is S_{22}. The use of this formula in AMG requires

1. to select an appropriate partitioning of the matrix A, e.g. by using a symmetric permutation matrix Π and considering a suitable leading block of $\Pi^\top A \Pi$,

2. the construction of a cheap but effective approximation B_{11} for A_{11}^{-1}.

Natural candidates for smoother, prolongation, restriction operators and the coarse grid system are the matrices

$$S = \begin{pmatrix} B_{11} & 0 \\ 0 & 0 \end{pmatrix}, \quad P = \begin{pmatrix} -B_{11}A_{12} \\ I \end{pmatrix}, \quad R = \begin{pmatrix} -A_{21}B_{11} & I \end{pmatrix} \text{ and } A_H = RAP.$$

For the algebraic multilevel method one could use either

$$(2) \qquad A^{-1} \approx M_A = S + PA_H^{-1}R,$$

which we will refer as the **additive algebraic multigrid (AAMG)** version, or as mentioned previously we could use M_V from

$$I - M_V A = (I - SA)(I - PA_H^{-1}RA)(I - SA).$$

This version is called **V-cycle version of algebraic multigrid**. Again, to obtain more than two levels of multigrid one has to apply the same principle recursively to A_H.

The paper is organized as follows. In Section 2 we will present the main idea for the construction of the AMG in this paper. Section 3 will discuss how the set of fine grid nodes, the basis for the setup of the smoother, is constructed. A theoretical justification of this approach is given. The aim of Section 4 is to introduce a special adapted sparse approximate inverse smoother that is originally based on the SPAI method [10]. We will comment on some technical implementation details in Section 5. Finally we will show several examples to illustrate this approach.

2 The basic idea

Suppose that a nonsingular $n \times n$ matrix A is given. The idea of the approximate inverse preconditioner SPAI [10, 1] is to minimize

$$\|I - AB\|_F.$$

This problem can be decoupled as

$$\|I - AB\|_F^2 = \sum_{i=1}^{n} \|e_i - ABe_i\|_2^2.$$

Each minimization problem can be solved independently. This makes this approach attractive for parallel computations. The major problem with this approach is that in several applications, particulary those that arise from the discretization of partial differential equations, the inverse A^{-1} of A is not approximately sparse. In this case one may obtain a dense matrix B and the construction of B becomes inefficient. On the other hand, the use of sparse approximate inverses as smoothers in multigrid methods may still be attractive [30, 19, 20, 7, 6]. The most simple way to accelerate the construction of B is to restrict the construction of B to a submatrix $A_{\mathcal{F},\mathcal{F}}$ of A called the fine grid part of A. Suppose we have a permutation Π such that

$$\Pi^\top A \Pi = \left(\begin{array}{cc} A_{\mathcal{F},\mathcal{F}} & A_{\mathcal{F},\mathcal{C}} \\ A_{\mathcal{C},\mathcal{F}} & A_{\mathcal{C},\mathcal{C}} \end{array} \right).$$

If $A_{\mathcal{F},\mathcal{F}}$ is suitably chosen, e.g. if $A_{\mathcal{F},\mathcal{F}}$ is strictly diagonal dominant, then we may expect that $A_{\mathcal{F},\mathcal{F}}^{-1}$ is approximately sparse and therefore we expect the reduced minimization problem

$$\|I - A_{\mathcal{F},\mathcal{F}} B_{\mathcal{F},\mathcal{F}}\|_F,$$

to be well-posed. Since this is only a partial approximate inverse, one has to find a complementary part and this can be done in terms of multigrid methods. In terms of the residual matrix

$$I - \Pi^\top A \Pi \left(\begin{array}{cc} B_{\mathcal{F},\mathcal{F}} & 0 \\ 0 & 0 \end{array} \right) = \left(\begin{array}{cc} I - A_{\mathcal{F},\mathcal{F}} B_{\mathcal{F},\mathcal{F}} & 0 \\ -A_{\mathcal{C},\mathcal{F}} B_{\mathcal{F},\mathcal{F}} & I \end{array} \right)$$

we observe that this matrix has approximately lower rank if $\|I - A_{\mathcal{F},\mathcal{F}} B_{\mathcal{F},\mathcal{F}}\|_F$ is small and the remaining part

$$R = \left(\begin{array}{cc} -A_{\mathcal{C},\mathcal{F}} B_{\mathcal{F},\mathcal{F}} & I \end{array} \right)$$

is precisely the restriction operator. To construct a restriction / interpolation from the residual matrix was discussed in [4] but for a general (full size) sparse approximate inverse. Here we get the lower rank directly from the norm minimization.

To turn this idea into an efficient multigrid method we have to discuss the following problems.

1. The detection of a subset $\mathcal{F} \subset \{1, \ldots, n\}$ such that the inverse of $A_{\mathcal{F}, \mathcal{F}}$ is approximately sparse,

2. The construction of the sparse approximate inverse itself,

3. The setup of the associated algebraic multigrid method.

In the sequel we will discuss these three topics. For technical reasons we will assume that the diagonal entries of A are nonnegative. This can always be achieved, e.g. by a row scaling from the left.

One assumption which we will make throughout the paper is that the vector

$$\tag{3} \mathbb{1} = \begin{pmatrix} 1 & \cdots & 1 \end{pmatrix}^\top$$

satisfies

$$\tag{4} A\mathbb{1} \approx 0.$$

It is a realistic assumption for problems arising from the discretization of second order partial differential equations and several others, since the vector $\mathbb{1}$ represents the set of constant functions.

3 Selection of fine grid nodes

We will now give a theoretic justification for the construction of the fine grid. The key idea is the following. One way to construct a set $\mathcal{F}$ of fine grid nodes could be to use the partitioning obtained from an existing AMG. I.e. in [19] this was done for SPAI [10] based on the coarsening from [26].

Here we will use a different approach. Our aim is to construct a set $\mathcal{F}$ such that $A_{\mathcal{F}\mathcal{F}}^{-1}$ is approximately sparse. I.e., given an $n \times n$ matrix A, find a permutation matrix Π such that for the permuted matrix

$$\Pi^\top A P = \begin{pmatrix} A_{\mathcal{F}\mathcal{F}} & A_{\mathcal{F}\mathcal{C}} \\ A_{\mathcal{C}\mathcal{F}} & A_{\mathcal{C}\mathcal{C}} \end{pmatrix},$$

the leading submatrix $A_{\mathcal{F}\mathcal{F}}$ satisfies

$$A_{\mathcal{F}\mathcal{F}}^{-1} = B_{\mathcal{F}\mathcal{F}} + R,$$

where $B_{\mathcal{F}\mathcal{F}}$ is sparse and R is an error matrix of small norm. In many serious application problems arising from the discretization of second order partial differential equations the diagonal entries of A are positive and at the same time the row sums are almost zero. The most simple criterion for finding such a block is diagonal dominance.

Definition 1 *An $n \times n$ matrix $A = (a_{ij})$ is said to be (strictly) diagonally dominant, if for all $i = 1, \ldots, n$*

$$|a_{ii}| > \sum_{j:\, j \neq i} |a_{ij}|.$$

Let A be diagonal dominant and $\gamma = \max_i \sum_{j:\, j \neq i} |a_{ij}|/|a_{ii}|$. Denote by Δ the diagonal matrix which contains only the diagonal entries of A on the main diagonal. Then we can write A as

$$A = \Delta(\Delta^{-1}A) = \Delta(I - E).$$

Using the Neumann series [22] we can write A^{-1} as

$$A^{-1} = (\Delta^{-1}A)^{-1}\Delta^{-1} = \left(\sum_{k=0}^{\infty} E^k\right)\Delta^{-1} = \left(\sum_{k=0}^{l} E^k + \mathcal{O}\left(\frac{\gamma^{l+1}}{1-\gamma}\right)\right)\Delta^{-1}.$$

So if $\gamma < 1$ is small enough, e.g. we may think of $\gamma = \frac{1}{2}, \frac{2}{3}$, then only few powers of E are required (say e.g. $l = 2$) and the remaining error is small.

The problem with this approach is, that in practice the diagonal dominance criterion is too restrictive to end up in an sufficiently large submatrix $A_{\mathcal{FF}}$, since the off–diagonal entries can have any sign (not only zero or negative). There are only few elementary cases where one can guarantee that the off–diagonal entries are nonpositive. This means that in general these kind of matrices are far away from diagonal dominance. So one has to search for more appropriate criteria. A mathematically simple way to find a criterion which ensures that a given nonsingular matrix A possesses a sparse approximate inverse can be stated using an LU decomposition. We first start with the case of a diagonal dominant matrix and then discuss the general case.

Lemma 2 *Let $A = (a_{ij})$ and $A^{\top}$ be diagonal dominant $n \times n$ matrices and suppose that for all $i, j = 1, \ldots, n$ we have*

$$\sum_{j:\, j \neq i} |a_{ij}| \leqslant \gamma |a_{ii}|, \quad \sum_{i:\, i \neq j} |a_{ij}| \leqslant \gamma |a_{jj}|$$

for a constant $\gamma < 1$. Then A has an LU decomposition $A = LDU$ with unit lower triangular matrices L and $U^{\top}$ and a diagonal matrix D. The inverse triangular factors L^{-1} and U^{-1} satisfy

$$\|L^{-1}\|_1 \leqslant 1 + \frac{\gamma}{1-\gamma}, \quad \|U^{-1}\|_{\infty} \leqslant 1 + \frac{\gamma}{1-\gamma}.$$

Let $K > 0$ be a fixed natural number. Then $|L^{-1}|$ has at most K off–diagonal entries in every column that are larger than $\frac{\gamma}{K(1-\gamma)}$ and likewise $|U^{-1}|$ has at most K off–diagonal entries in every row that are larger than $\frac{\gamma}{K(1-\gamma)}$.

Proof:
Let $A = (a_{ij})$. Performing one step of the LU decomposition we obtain

$$A = \begin{pmatrix} 1 & 0 \\ L_{21} & I \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & S_{22} \end{pmatrix} \begin{pmatrix} 1 & U_{12} \\ 0 & I \end{pmatrix},$$

where

$$L_{21} = \left(\frac{a_{i1}}{a_{11}}\right)_{i \geqslant 2}, \quad U_{12} = \left(\frac{a_{1j}}{a_{11}}\right)_{j \geqslant 2}, \quad S_{22} = \left(a_{ij} - \frac{a_{i1}a_{1j}}{a_{11}}\right)_{i,j \geqslant 2}.$$

From the fact that $\sum_{j:\, j \neq i} |a_{ij}| \leqslant \gamma |a_{ii}|$ for all $i = 1, \ldots, n$ we obtain for $i = 2, \ldots, n$ that

$$\sum_{\substack{j \geqslant 2: \\ j \neq i}} \left| a_{ij} - \frac{a_{i1}a_{1j}}{a_{11}} \right| \;\leqslant\; \sum_{\substack{j \geqslant 2: \\ j \neq i}} |a_{ij}| + \left|\frac{a_{i1}}{a_{11}}\right| \sum_{\substack{j \geqslant 2: \\ j \neq i}} |a_{1j}|$$

$$\leqslant\; \gamma |a_{ii}| - |a_{i1}| + \left|\frac{a_{i1}}{a_{11}}\right| (\gamma |a_{11}| - |a_{1i}|)$$

$$\leqslant\; \gamma \left(|a_{ii}| - \left|\frac{a_{i1}a_{1i}}{a_{11}}\right|\right)$$

$$\leqslant\; \gamma \left| a_{ii} - \frac{a_{i1}a_{1i}}{a_{11}} \right|.$$

We observe that the Schur complement S_{22} satisfies the diagonal dominance criterion with the same constant γ. The same argument can be applied to $S_{22}^\top$. Using induction it follows that the resulting factors $L = (l_{ij})$ and $U = (u_{ij})$ satisfy

$$\sum_{i:\, i>j} |l_{ij}| \leqslant \gamma, \ \sum_{j:\, j>i} |u_{ij}| \leqslant \gamma.$$

From this it follows that $L = I - E_L$ and $U = I - E_U$ with matrices E_L, E_U such that $\|E_L\|_\infty \leqslant \gamma$ and $\|E_U\|_\infty \leqslant \gamma$. Applying Neumann series [22] we obtain

$$\|L^{-1}\|_1 \leqslant \frac{1 - \gamma^n}{1 - \gamma}, \ \|U^{-1}\|_\infty \leqslant \frac{1 - \gamma^n}{1 - \gamma}.$$

Since

$$\frac{1 - \gamma^n}{1 - \gamma} \leqslant 1 + \frac{\gamma}{1 - \gamma},$$

we conclude that the off-diagonal entries of $L^{-1} = (\lambda_{ij})$ and $U^{-1} = (\rho_{ij})$ satisfy

$$\sum_{i:\, i>j} |\lambda_{ij}| < \frac{\gamma}{1 - \gamma}, \ \sum_{j:\, j<i} |\rho_{ij}| < \frac{\gamma}{1 - \gamma}.$$

It follows that for a given natural number K there are at most K off–diagonal entries in each row such that each of them is at least of absolute value $\frac{\gamma}{K(1-\gamma)}$. $\qquad\square$

This lemma shows that diagonal dominance can also be used to ensure that the inverse triangular factors are approximately sparse. In the diagonal dominant case $\|L^{-1}\|_1$ and $\|U^{-1}\|_\infty$ are moderately bounded. To verify this, one does not need to compute L and U explicitly. Of course this is not true in general.

The following lemma generalizes this relatively simple criteria for the existence of a sparse approximate inverse, if the matrix owns an LU decomposition.

Lemma 3 *Suppose that $A = LDU$, where L and $U^\top$ are lower triangular matrices with unit diagonal and D is a diagonal matrix. Let $K > 0$ be a natural number. Suppose that $\|L^{-1}\|_1 \leqslant 1 + \alpha$ and $\|U^{-1}\|_\infty \leqslant 1 + \alpha$. Then $|L^{-1}|$ has at most K off–diagonal entries in any column that are larger than $\frac{\alpha}{K}$, $|U^{-1}|$ has at most K off–diagonal entries in any row that are larger than $\frac{\alpha}{K}$.*

Proof:
This is clear. $\qquad\square$

Of course we are not going to compute a complete LU decomposition of A or a submatrix of A in order to find out whether its inverse triangular factors are nicely bounded or not. As mentioned earlier in (4) we assume that $A\mathbb{1} \approx 0$, where $\mathbb{1}$ is as in (3). If $A\mathbb{1} \approx 0$, then this vector can serve as good estimate for $\|A^{-1}\|$. Therefore we simplify the criterion from Lemma 3 as follows

1. We compute an incomplete LU decomposition instead of an exact LU decomposition. The incomplete LU decomposition we use is MILU(0) [18, 17], i.e., we ignore fill–in and at the same time the approximate LU decomposition satisfies

$$LDU = A + E, \quad LDU\mathbb{1} = A\mathbb{1}.$$

In other words diagonal compensation is applied to ensure that the factorization is exact for the vector $\mathbb{1}$. By this constraint we also try to enforce to keep $\|L^{-1}\|$ and $\|U^{-1}\|$ close to the norm of the exact inverse triangular factors.

2. The matrices L^{-1} and U^{-1} are of course not explicitly computed. Their norm is estimated by

$$\|L^{-1}\| \to |L^{-1}\mathbb{1}|, \ \|U^{-\top}\| \to |U^{-\top}\mathbb{1}|.$$

To do this, it is simply required to perform one forward and one backward solve during the computation of MILU(0). The components of $|L^{-1}\mathbb{1}|$ are obviously a lower bound for $\|L^{-1}\|_\infty$ since $\|\mathbb{1}\|_\infty = 1$ and hopefully the estimate is close enough to the true norm. Similar arguments apply to U.

To find a submatrix $A_{\mathcal{F},\mathcal{F}}$ such that its inverse triangular factors are approximately sparse we proceed as follows. Initially we do not know whether the norms of $\|L^{-1}\|$ and $\|U^{-1}\|$ are moderately bounded. So we have to seek for a submatrix of A such that in any step k of the incomplete LU decomposition, the norms of $\|L_k^{-1}\|$ and $\|U_k^{-1}\|$ do not exceed a prescribed tolerance. This will give the fine grid. The cheapest way of doing this is to start with the initial matrix A, compute the incomplete LU decomposition step by step as in Algorithm 4.

Algorithm 4 (MILU(0) detection of a sparse approximate inverse) *Given a matrix A and a bound α we set $\mathcal{F} = \mathcal{C} = \emptyset$. Perform an MILU(0) decomposition of A.*
for $k = 1, 2, \ldots, n$
    compute column k of L and row k of U according to MILU(0).
    if *the k–th component of $|L^{-1}\mathbb{1}|$ or $|U^{-\top}\mathbb{1}|$ is greater than α,*
    then $\mathcal{C} = \mathcal{C} \cup \{k\}$
    else $\mathcal{F} = \mathcal{F} \cup \{k\}$

This procedure of detecting fine / coarse grid nodes has one slight disadvantage. It strongly depends on the prescribed initial permutation of the coefficient matrix A. It may happen that for a couple of steps $k = 1, \ldots, l$ the norm estimates of the inverse triangular factors stay below the given bound α, i.e. many nodes from $k = 1, \ldots, l$ can be moved to the fine grid. Then at a certain step $l + 1$ suddenly the norm estimates of the triangular factors start to grow more and more, resulting in many nodes that have to be moved to the coarse grid. Another undesired effect may be caused by the fact that the factorization is only incomplete and the norms of L^{-1} and U^{-1} might be too inaccurate. Of course one should avoid these effects. To do this we use an additional algorithm to reorder the matrix beforehand. The idea is to preselect columns and rows according to a heuristic that constructs a (hopefully) large block $A_{\mathcal{F},\mathcal{F}}$, such that for a given $\gamma < 1$

(5) $$A_{\mathcal{F},\mathcal{F}}\mathbb{1} \geqslant (1 - \gamma)\Delta_{\mathcal{F},\mathcal{F}}\mathbb{1}.$$

Here Δ again denotes the diagonal part of A and we assume for simplicity that its diagonal entries are positive. This criterion means that for all $i \in \mathcal{F}$

$$-\sum_{j \in \mathcal{F}: j \neq i} a_{ij} \leqslant \gamma a_{ii}.$$

If A has positive diagonal entries and non–positive off-diagonal entries, then this criterion is equivalent to diagonal dominance in the sense of Definition 1. Otherwise it means, that for the test vector $\mathbb{1}$,

$$\Delta_{\mathcal{F},\mathcal{F}}^{-1} A_{\mathcal{F},\mathcal{F}}\mathbb{1} \geqslant (1 - \gamma)\mathbb{1},$$

the matrix $\Delta_{\mathcal{F},\mathcal{F}}^{-1} A_{\mathcal{F},\mathcal{F}} \mathbb{1}$ is uniformly bounded from below. We hope that the inverse relation

$$A_{\mathcal{F},\mathcal{F}}^{-1} \Delta_{\mathcal{F},\mathcal{F}} \mathbb{1} \leqslant \frac{1}{1-\gamma} \mathbb{1}$$

is also satisfied. But of course this is only a heuristic argument, except for the case that $A_{\mathcal{F},\mathcal{F}}^{-1}$ is non-negative. The purpose of the MILU(0) fine grid detection in Algorithm 4 is precisely the counter part that fits with this heuristic. While the following heuristic only computes an initial guess for $\mathcal{F}$, Algorithm 4 verifies and ensures (within the limits of an ILU and the norm estimates), that the inverse triangular factors are bounded.

The heuristic algorithm that is put in front of Algorithm 4 can roughly be described as follows. We compute three sets $\mathcal{F}, \mathcal{C}$ and $\mathcal{U}$. $\mathcal{F}$ reflects the desired set of fine grid nodes, $\mathcal{C}$ the coarse grid nodes that have to be taken out and, finally $\mathcal{U}$ denotes a set of nodes that are left over. The heuristic constructs a permutation matrix Π such that the permuted system $\Pi^\top A \Pi$ can be written as

$$\Pi^\top A \Pi = \begin{pmatrix} A_{\mathcal{F},\mathcal{F}} & A_{\mathcal{F},\mathcal{U}} & A_{\mathcal{F},\mathcal{C}} \\ A_{\mathcal{U},\mathcal{F}} & A_{\mathcal{U},\mathcal{U}} & A_{\mathcal{U},\mathcal{C}} \\ A_{\mathcal{C},\mathcal{F}} & A_{\mathcal{C},\mathcal{U}} & A_{\mathcal{C},\mathcal{C}} \end{pmatrix}.$$

For these initial guesses of $\mathcal{F}, \mathcal{C}$ and $\mathcal{U}$ we ensure that

(6) $$\begin{pmatrix} A_{\mathcal{F},\mathcal{F}} & A_{\mathcal{F},\mathcal{U}} \end{pmatrix} \mathbb{1} \geqslant (1-\gamma) \Delta_{\mathcal{F},\mathcal{F}} \mathbb{1}.$$

This can be read as an (unsafe) indicator to find out if the $MILU(0)$ decomposition of

(7) $$\begin{pmatrix} A_{\mathcal{F},\mathcal{F}} & A_{\mathcal{F},\mathcal{U}} \\ A_{\mathcal{U},\mathcal{F}} & A_{\mathcal{U},\mathcal{U}} \end{pmatrix}$$

is moderately bounded. I.e. MILU(0) is not performed on the whole matrix but only on this preprocessed submatrix (7). In Algorithm 4 any column / row that does not fit with the estimates for L^{-1}, U^{-1} is removed from $\mathcal{F}$, $\mathcal{U}$, and added to $\mathcal{C}$. We give a simple sketch for this heuristic algorithm. It can be seen as some kind of generalization of the coarse grid selection scheme in [26].

Algorithm 5 (Heuristics for finding an initial fine grid set $\mathcal{F}$)   *Given a matrix A and a bound α we set $\mathcal{F} = \mathcal{C} = \emptyset$, $\mathcal{U} = \{1, \ldots, n\}$. For all i we define sets Λ_i, Σ_i and $\mathcal{N}_i$.*

$\Lambda_i := \{j \in \mathcal{U} : a_{ij} \neq 0, \sum_{k \in \mathcal{F} \cup \mathcal{U} \setminus j} a_{ik} \geqslant (1-\gamma) a_{ii}\}$.

Λ_i describes those nodes such that the row sum in row i becomes sufficiently large if one of the nodes from Λ_i is moved to the coarse grid $\mathcal{C}$.
$\Lambda_i^\top := \{j \in \mathcal{U} : i \in \Lambda_j\}$. *$\Lambda_i^\top$ is the dual set with respect to Λ_i.*

$\Sigma_i := \{j \in \mathcal{U} : a_{ij} < 0 \text{ smallest possible such that } \sum_{k \in \mathcal{F} \cup \mathcal{U} \setminus \Sigma_i} a_{ik} \geqslant (1-\gamma) a_{ii}\}$.

Σ_i describes those nodes such that the row sum in row i becomes sufficiently large if ALL nodes from Σ_i would be moved to the coarse grid $\mathcal{C}$.
$\Sigma_i^\top := \{j \in \mathcal{U} : i \in \Sigma_j\}$. *$\Sigma_i^\top$ is the dual set with respect to Σ_i.*

For all $i \in \mathcal{U}$: $\mathcal{N}_i = \{j \in \mathcal{F} : j \to \mathcal{U} \text{ if } i \to \mathcal{F}\}$
$\mathcal{N}_i$ describes those fine grid nodes that have to be moved back to the set $\mathcal{U}$ of undetermined nodes, if i becomes a member of the fine grid $\mathcal{F}$.

__while__ $\max |\Lambda_i^\top| > 0$ *or* $\max |\Sigma_i^\top| > 0$
      __if__ *there exists* Λ_i *such that* $|\Lambda_i| = 1$
      __then__ *choose one of these and set* $\mathcal{C} = \mathcal{C} \cup \Lambda_i$
      __elseif__ *there exists a* $\Lambda_i^\top$ *with* $|\Lambda_i^\top| > 0$
      __then__ *choose* $\Lambda_i^\top$ *such that* $|\Lambda_i^\top| - |\mathcal{N}_i|$ *is maximal and set* $\mathcal{C} = \mathcal{C} \cup \{i\}$
      __else__ *choose* $\Sigma_i^\top$ *with maximal* $|\Sigma_i^\top|$ *and set* $\mathcal{C} = \mathcal{C} \cup \{i\}$
      update $\Lambda_j, \Lambda_j^\top, \Sigma_j, \Sigma_j^\top, \mathcal{N}_j, \mathcal{F}, \mathcal{U}$.

The concept of Algorithm 5 roughly reads as follows.

1. If for a single row i there exists exactly one j such that row sum in row i becomes sufficiently large after removing a_{ij}, then j is moved to the coarse grid.

2. Otherwise we are seeking for a node i such that as many rows as possible become sufficiently large, if i is moved to the coarse grid. This is measured by the size of the adjoint set $\Lambda_i^\top$.

3. It may happen, that it is not enough to move a single node i to the coarse grid in order to produce sufficiently large row sums. For this case the sets $\Sigma_i, \Sigma_i^\top$ are created. In this case $|\Sigma_i^\top|$ counts the number of rows such that their row sum increases significantly if i is moved to the coarse grid.

We summarize the results of this section. We have developed two algorithms for the construction of a submatrix $A_{\mathcal{F},\mathcal{F}}$ such that its inverse is approximately sparse. Initially Algorithm 5 computes sets $\mathcal{F}, \mathcal{C}$ and $\mathcal{U}$ of fine grid nodes, coarse grid nodes and remaining undetermined nodes. This heuristic enforces that the relation (3) is at least partially satisfied (see (6)). This algorithm is followed by Algorithm 4, which computes a modified incomplete LU decomposition with no fill–in. Any node from $\mathcal{F} \cup \mathcal{U}$ is removed and added to $\mathcal{C}$, if the prescribed estimated bounds for the L^{-1} and U^{-1} are exceeded.

After the detection of a set of fine grid nodes $\mathcal{F}$ the next step consists of computing a sparse approximate inverse to $A_{\mathcal{F},\mathcal{F}}$.

4   Construction of an adapted sparse approximate inverse

For the computation of a sparse approximate inverse of a given matrix $A_{\mathcal{F},\mathcal{F}}$ there are essentially two major concepts. One concept is based on the AINV–type algorithms [2, 3]. These kind of algorithms directly construct unit upper triangular factors W and Z such that $W^\top A Z$ is approximately diagonal. Another concept is based on a non–factored sparse inverse matrix B, such that $\|I - AB\|_F$ is small (see SPAI, [10]). Here we will concentrate on the second approach. Let e_j denote the j–th unit vector. The idea is based on the observation that

$$\|I - AB\|_F^2 = \sum_{j=1}^{n} \|e_j - ABe_j\|_2^2$$

can be minimized for each column separately. The problem in practice is to select a suitable nonzero pattern for Be_j, since otherwise the minimization problem $\|e_j - ABe_j\|_2^2 = \min$ becomes too costly. The matrix for which we construct the sparse approximate inverse here is not A itself but $A_{\mathcal{F},\mathcal{F}}$. The purpose of the selection process to determine $\mathcal{F}$ is precisely to

construct $A_{\mathcal{F},\mathcal{F}}$ such that $A_{\mathcal{F},\mathcal{F}}^{-1}$ is approximately sparse. But even in this case we have no prediction on the positions of large nonzero components in $|A_{\mathcal{F},\mathcal{F}}^{-1}|$. There are several ways to define the pattern of $B_{\mathcal{F},\mathcal{F}}$. One could e.g. use powers of $I + |A_{\mathcal{F},\mathcal{F}}|$ or use the residual $e_j - ABe_j$ [10]. A closely related strategy consists of minimizing $\|e_j - ABe_j\|_2^2$ using GMRES [27] applied to sparse vectors and drop small entries [8]. In order to minimize $\|e_j - ABe_j\|_2^2$ one prescribes for all strategies a tolerance τ. As long as $\|e_j - ABe_j\|_2^2 > \tau$, the minimization process is continued, i.e. depending on the strategy for defining the pattern of B, new nonzero components are added or a further GMRES step is performed.

But simply to reduce the error $\|I - A_{\mathcal{F},\mathcal{F}}B_{\mathcal{F},\mathcal{F}}\|_F$ in algebraic multilevel methods is not enough. From the point of view of a block ILU decomposition one has to take the Schur complement (coarse grid system) into account. Even if $\|(I - A_{\mathcal{C},\mathcal{F}}B_{\mathcal{C},\mathcal{F}})A_{\mathcal{F},\mathcal{C}}\|_F$ is small, $\|(I - A_{\mathcal{C},\mathcal{F}}B_{\mathcal{C},\mathcal{F}})A_{\mathcal{F},\mathcal{C}}A_H^{-1/2}\|_F$ may be large.

This can be seen from the following theorem, which discusses the positive definite case.

Theorem 6 *Let A be symmetric positive definite,*

$$A = \begin{pmatrix} A_{\mathcal{F},\mathcal{F}} & A_{\mathcal{F},\mathcal{C}} \\ A_{\mathcal{C},\mathcal{F}} & A_{\mathcal{C},\mathcal{C}} \end{pmatrix}$$

and let $B_{\mathcal{F},\mathcal{F}}$ a nonsingular approximate inverse of $A_{\mathcal{F},\mathcal{F}}$. Define E via

$$E = A_{\mathcal{C},\mathcal{F}}(A_{\mathcal{F},\mathcal{F}}^{-1} - B_{\mathcal{F},\mathcal{F}}^{\top})A_{\mathcal{F},\mathcal{F}}(A_{\mathcal{F},\mathcal{F}}^{-1} - B_{\mathcal{F},\mathcal{F}})A_{\mathcal{F},\mathcal{C}}$$

and introduce the Schur complement by

$$S_{\mathcal{C},\mathcal{C}} = A_{\mathcal{C},\mathcal{C}} - A_{\mathcal{C},\mathcal{F}}A_{\mathcal{F},\mathcal{F}}^{-1}A_{\mathcal{F},\mathcal{C}}.$$

Define an approximate block LU decomposition M via

$$M = \begin{pmatrix} I & 0 \\ A_{\mathcal{C},\mathcal{F}}B_{\mathcal{F},\mathcal{F}} & I \end{pmatrix} \begin{pmatrix} A_{\mathcal{F},\mathcal{F}} & 0 \\ 0 & A_H \end{pmatrix} \begin{pmatrix} I & B_{\mathcal{F},\mathcal{F}}A_{\mathcal{C},\mathcal{F}} \\ 0 & I \end{pmatrix},$$

where $A_H = S_{\mathcal{C},\mathcal{C}} + E$.
Then in the sense of quadratic forms the following inequalities hold:

$$\left(1 - \sqrt{\sigma}\right) M \leqslant A \leqslant \left(1 + \sqrt{\sigma}\right) M,$$

where the parameter σ is has to satisfy the quadratic form relation

$$E \leqslant \sigma A_H.$$

Proof:
The proof can be easily adapted from [11]. □

Note that the block 2-level preconditioner M in Theorem 6 can easily be extended to the case when $A_{\mathcal{F},\mathcal{F}}$ is replaced by a preconditioner.

Suppose that A is symmetric and positive definite. The main problem is to obtain a small σ. If $B_{\mathcal{F},\mathcal{F}} \approx A_{\mathcal{F},\mathcal{F}}^{-1}$, then we may expect that in general $\|E\|$ is relatively small. But to obtain a small σ one has to ensure that

$$v^{\top}Ev \leqslant \sigma v^{\top}A_H v$$

for all v and especially for those eigenvectors v that are associated with the small eigenvalues of A_H. A similar observation can also be found in [23]. In practice we do not know v in advance. But we assume that the test vector $\mathbb{1}$ may serve as a good test vector to obtain a small σ. The optimal case would be to force the approximate inverse $B_{\mathcal{F},\mathcal{F}}$ to satisfy the equation

$$B_{\mathcal{F},\mathcal{F}} A_{\mathcal{F},\mathcal{C}}\, \mathbb{1} = A_{\mathcal{F},\mathcal{F}}^{-1} A_{\mathcal{F},\mathcal{C}}\, \mathbb{1},$$

or equivalently

(8) $$A_{\mathcal{F},\mathcal{F}} B_{\mathcal{F},\mathcal{F}}\, w = w, \text{ where } w = A_{\mathcal{F},\mathcal{C}}\, \mathbb{1}.$$

Instead of only minimizing $\|I - A_{\mathcal{F},\mathcal{F}} B_{\mathcal{F},\mathcal{F}}\|_F$ one has to minimize

$$\|I - A_{\mathcal{F},\mathcal{F}} B_{\mathcal{F},\mathcal{F}}\|_F \text{ subject to } A_{\mathcal{F},\mathcal{F}} B_{\mathcal{F},\mathcal{F}}\, w = w, \text{ where } w = A_{\mathcal{F},\mathcal{C}}\, \mathbb{1}.$$

The concept of adding a linear constraint to the minimization process of the sparse approximate inverse can already be found in [14], there for the whole system and with a different test vector. The linear constraint (8) has one essential disadvantage. We do not know, if the components of $|w|$ are bounded from below. As we will see later, this is an important condition to make the solution of this constraint minimization problem numerically stable. For this reason below we describe a different linear constraint.

In general, problems arising from the discretization of partial differential equations fulfill the condition

$$\begin{pmatrix} A_{\mathcal{F},\mathcal{F}} & A_{\mathcal{F},\mathcal{C}} \\ A_{\mathcal{C},\mathcal{F}} & A_{\mathcal{C},\mathcal{C}} \end{pmatrix} \mathbb{1} = f \approx 0,$$

where f is vector that is zero almost everywhere except for a few components that are associated with boundary conditions. Suppose for simplicity that

$$A_{\mathcal{F},\mathcal{F}}\, \mathbb{1} + A_{\mathcal{F},\mathcal{C}}\, \mathbb{1} = 0.$$

Multiplying by $B_{\mathcal{F},\mathcal{F}}$ yields

$$B_{\mathcal{F},\mathcal{F}} A_{\mathcal{F},\mathcal{F}}\, \mathbb{1} = -B_{\mathcal{F},\mathcal{F}} A_{\mathcal{F},\mathcal{C}}\, \mathbb{1}.$$

If we ensure that the sparse approximate inverse $B_{\mathcal{F},\mathcal{F}}$ is constructed such that

(9) $$B_{\mathcal{F},\mathcal{F}} A_{\mathcal{F},\mathcal{F}}\, \mathbb{1} = \mathbb{1},$$

then we also obtain

$$-B_{\mathcal{F},\mathcal{F}} A_{\mathcal{F},\mathcal{C}}\, \mathbb{1} = B_{\mathcal{F},\mathcal{F}} A_{\mathcal{F},\mathcal{F}}\, \mathbb{1} = \mathbb{1} = -A_{\mathcal{F},\mathcal{F}}^{-1} A_{\mathcal{F},\mathcal{C}}\, \mathbb{1}.$$

Thus instead of simply minimizing $\|I - A_{\mathcal{F},\mathcal{F}} B_{\mathcal{F},\mathcal{F}}\|_F$ we consider the following constraint minimization problem

(10) $$\|I - A_{\mathcal{F},\mathcal{F}} B_{\mathcal{F},\mathcal{F}}\|_F = \min \text{ subject to } B_{\mathcal{F},\mathcal{F}} A_{\mathcal{F},\mathcal{F}}\, \mathbb{1} = \mathbb{1}.$$

Lemma 7 *The constraint minimization problem (10) is equivalent to the following linear system. Suppose that $|\mathcal{F}| = l$. Denote by $E_1, \ldots, E_l$ the sequence of unit vectors such that the columns of E_i correspond to the nonzero pattern of Be_i. Set $w = A_{\mathcal{F},\mathcal{F}}\, \mathbb{1}$.*

$$\begin{pmatrix} C^\top C & W \\ W^\top & 0 \end{pmatrix} \begin{pmatrix} x \\ \lambda \end{pmatrix} = \begin{pmatrix} C^\top e \\ \mathbb{1} \end{pmatrix},$$

where

$$
C = \begin{pmatrix} A_{\mathcal{F},\mathcal{F}}E_1 & & 0 \\ & \ddots & \\ 0 & & A_{\mathcal{F},\mathcal{F}}E_l \end{pmatrix}, W = \begin{pmatrix} w_1 E_1^\top \\ \vdots \\ w_l E_l^\top \end{pmatrix}, x = \begin{pmatrix} E_1^\top B_{\mathcal{F},\mathcal{F}}e_1 \\ \vdots \\ E_l^\top B_{\mathcal{F},\mathcal{F}}e_l \end{pmatrix}, e = \begin{pmatrix} e_1 \\ \vdots \\ e_l \end{pmatrix}.
$$

u If the nonzero pattern of each column $B_{\mathcal{F},\mathcal{F}}e_i$ covers $\{1,\dots,l\}$ and if $w = A_{\mathcal{F},\mathcal{F}}\mathbb{1}$ is a nonzero vector, then the constraint linear system is uniquely solvable.

Proof:
Set $x_i = E_i^\top B_{\mathcal{F},\mathcal{F}}e_i$ and $A_i = A_{\mathcal{F},\mathcal{F}}E_i$. We note that for all $i = 1,\dots,l$ we have

$$
\|e_i - A_{\mathcal{F},\mathcal{F}}B_{\mathcal{F},\mathcal{F}}e_i\|_2^2 = \|e_i - A_i x_i\|_2^2 = x_i^\top A_i^\top A_i x_i - e_i^\top A_i x_i - x_i^\top A_i^\top e_i + 1.
$$

Thus to minimize $\|e_i - A_{\mathcal{F},\mathcal{F}}B_{\mathcal{F},\mathcal{F}}e_i\|_2^2$ obviously is a quadratic minimization problem. A necessary and sufficient condition for the minimization problem is that its Jacobian satisfies

$$
A_i^\top A_i x_i = A_i^\top e_i,
$$

since $A_i^\top A_i$ is symmetric positive definite. Altogether we have

$$
C^\top C x = C^\top e.
$$

The linear constraint $B_{\mathcal{F},\mathcal{F}}A_{\mathcal{F},\mathcal{F}}\mathbb{1} = v$ can be rewritten as

$$
\mathbb{1} = B_{\mathcal{F},\mathcal{F}}A_{\mathcal{F},\mathcal{F}}\mathbb{1} = \sum_{i=1}^{l} B_{\mathcal{F},\mathcal{F}}e_i e_i^\top A v = \sum_{i=1}^{l} E_i x_i w_i = W^\top x.
$$

Using the Lagrangian multiplier approach we augment $\|e_i - A_{\mathcal{F},\mathcal{F}}B_{\mathcal{F},\mathcal{F}}e_i\|_2^2$ with the dual linear constraint $W\lambda$. This resulting matrix of the Jacobian is precisely

$$
\begin{pmatrix} C^\top C & W \\ W^\top & 0 \end{pmatrix}.
$$

For the second part one can see immediately that W has full rank, if all weights $w_1,\dots,w_l$ are nonzero and the matrix $[E_1,\dots,E_l]$ contains all unit vectors. $\qquad\square$

Note that the condition $A_{\mathcal{F},\mathcal{F}}\mathbb{1} \geqslant \frac{1}{1-\gamma}\Delta_{\mathcal{F},\mathcal{F}}\mathbb{1}$ in equation (5) is used again in Lemma 7. It ensures that the weights $w_1,\dots,w_l$ are of the same order as the associated diagonal entries.

To construct the sparse approximate inverse subject to a linear constraint (10) we will proceed as follows. We start solving the unconstraint minimization problem $\|I - A_{\mathcal{F},\mathcal{F}}B_{\mathcal{F},\mathcal{F}}\|_F$. Here one can use GMRES [8] or, if A is symmetric positive definite, cg [13] could be used. After the unconstraint minimization problem has been solved sufficiently accurate, we use the solution as initial guess for the constrained minimization process. For the constraint minimization process we work on a projected system. We define a projector P via

$$
P = I - W(W^\top W)^{-1}W^\top,
$$

where $W^\top W = \sum_{i=1}^{l} w_i^2 E_i E_i^\top$ is a diagonal matrix. The projection is cheap to apply. We decompose the solution x from Lemma 7 as

$$
x = W(W^\top W)^{-1}\mathbb{1} + Py
$$

for a suitable y. By this definition we ensure that

$$W^\top x = \mathbb{1}$$

is already fulfilled. The constraint linear system from Lemma 7 reduces to

$$P^\top CC(W(W^\top W)^{-1}\mathbb{1} + Py) = P^\top C^\top e$$

This is equivalent to the system

$$(P^\top CCP)y = P^\top C^\top \left(e - CW(W^\top W)^{-1}\mathbb{1} \right).$$

In other words, y has to minimize the projected minimization problem

(11) $$\|CPy - \left(e - CW(W^\top W)^{-1}\mathbb{1} \right)\|_2.$$

This problem is again solved using cg applied to the projected normal equations. Usually it suffices only to apply one or two steps of the constraint minimization problem, since the initial guess is a relatively accurate solution of the unconstraint minimization problem.

In this section we have discussed how to minimize the residual matrix $\|I - A_{\mathcal{F},\mathcal{F}}B_{\mathcal{F},\mathcal{F}}\|_F$ in order to construct a sparse approximate inverse $B_{\mathcal{F},\mathcal{F}}$. In the sense of multigrid methods more has to be done than to simply minimize the residual. For this purpose the linear constraint $B_{\mathcal{F},\mathcal{F}}A_{\mathcal{F},\mathcal{F}}\mathbb{1} = \mathbb{1}$ is added. The solution of this constrained minimization problem leads to an augmented linear system. The associated numerical solution can be computed solving the projected minimization problem (11).

5   Basic construction of the AMG

After the description of the selection of the fine grid nodes and the construction of the smoother we comment on the practical set up of the AMG preconditioner.

For the numerical experiments we focus on the additive AMG (2), since this approach is attractive because it does not require to store the coarse grid systems.

We concentrate on the class of symmetric positive definite matrices, although many arguments also apply to general matrices. This will be discussed in a later paper.

5.1   AMG setup

Starting with the initial matrix $A_0 = A$, on any level $k = 0, 1, 2, \ldots$ we are faced with a matrix A_k of size $n_k \times n_k$. We select fine grid nodes $\mathcal{F} \subset \{1, \ldots, n_k\}$ such that for a suitable permutation matrix Π we have

$$\Pi_k^\top A_k \Pi_k = \begin{pmatrix} A_{\mathcal{F}\mathcal{F}} & A_{\mathcal{F}\mathcal{C}} \\ A_{\mathcal{C}\mathcal{F}} & A_{\mathcal{C}\mathcal{C}} \end{pmatrix},$$

Then we construct a sparse approximate inverse $B_{\mathcal{F}\mathcal{F}}$ subject to the constraint minimization problem (10). This approximate inverse in general will not be symmetric. We define the smoother S_k by

$$S_k = \Pi_k \frac{1}{2} \begin{pmatrix} B_{\mathcal{F}\mathcal{F}} + B_{\mathcal{F}\mathcal{F}}^\top & 0 \\ 0 & 0 \end{pmatrix} \Pi_k^\top,$$

the prolongation operator P and restriction R by

$$P_k = \Pi_k \begin{pmatrix} -B_{\mathcal{FF}} A_{\mathcal{FC}} \\ I \end{pmatrix}, \quad R_k = P_k^\top.$$

Finally we set

$$A_{k+1} = R_k \begin{pmatrix} A_{\mathcal{FF}} & A_{\mathcal{FC}} \\ A_{\mathcal{CF}} & A_{\mathcal{CC}} \end{pmatrix} P_k.$$

Using this sequence of matrices, the **additive algebraic multigrid preconditioner (AAMG)** $M \equiv M_0$ is defined via

$$\begin{aligned} M_k &= S_k + P_k M_{k+1} R_k, \text{ if } k < l \\ M_l &= A_l^{-1}. \end{aligned}$$

5.2 Further details

In this subsection we comment on further practical details. For Algorithm 5 we use $\gamma = \frac{2}{3}$ in inequality (5). For the bound α of the inverse triangular factors in Algorithm 4 we use $\alpha = \frac{\gamma}{1-\gamma}$ which is motivated by the diagonal dominant case. From a practical point of view it is not recommended to minimize precisely $\|I - A_{\mathcal{F},\mathcal{F}} B_{\mathcal{F},\mathcal{F}}\|_F$, since the diagonal entries of A may have different orders of magnitude. From the point of view of block Gaussian elimination we locally compute

$$\begin{pmatrix} I & 0 \\ -A_{\mathcal{C},\mathcal{F}} B_{\mathcal{F},\mathcal{F}} & I \end{pmatrix} \begin{pmatrix} A_{\mathcal{F},\mathcal{F}} & A_{\mathcal{F},\mathcal{C}} \\ A_{\mathcal{C},\mathcal{F}} & A_{\mathcal{C},\mathcal{C}} \end{pmatrix} \begin{pmatrix} I & -B_{\mathcal{F},\mathcal{F}} A_{\mathcal{F},\mathcal{C}} \\ 0 & I \end{pmatrix}$$
$$= \begin{pmatrix} A_{\mathcal{F},\mathcal{F}} & (I - A_{\mathcal{F},\mathcal{F}} B_{\mathcal{F},\mathcal{F}}) A_{\mathcal{F},\mathcal{C}} \\ A_{\mathcal{C},\mathcal{F}} (I - B_{\mathcal{F},\mathcal{F}}^\top A_{\mathcal{F},\mathcal{F}}) & A_H \end{pmatrix},$$

where A_H denotes the coarse grid system. Before we compute $B_{\mathcal{F},\mathcal{F}}$, we could use an simple initial guess $B_{\mathcal{F},\mathcal{F}}$ by using

$$B_{\mathcal{F},\mathcal{F}}^{(0)} = 2\Delta_{\mathcal{F},\mathcal{F}}^{-1} - \Delta_{\mathcal{F},\mathcal{F}}^{-1} A_{\mathcal{F},\mathcal{F}} \Delta_{\mathcal{F},\mathcal{F}}^{-1},$$

which is the linear part of the Neumann expansion for $A_{\mathcal{F},\mathcal{F}}^{-1}$. Here $\Delta_{\mathcal{F},\mathcal{F}}$ again denotes the diagonal part of $A_{\mathcal{F},\mathcal{F}}$. We use the diagonal matrix

$$G = \operatorname{diag}(g_1, \ldots, g_l), \text{ where } g_i = \sqrt{\|B_{\mathcal{F},\mathcal{F}}^{(0)} e_i\|_\infty}$$

to scale the residual matrix $I - A_{\mathcal{F},\mathcal{F}} B_{\mathcal{F},\mathcal{F}}$. Thus, instead of minimizing $\|I - A_{\mathcal{F},\mathcal{F}} B_{\mathcal{F},\mathcal{F}}\|_F$, we minimize

$$\|G(I - A_{\mathcal{F},\mathcal{F}} B_{\mathcal{F},\mathcal{F}}) G^{-1}\|_F$$

Similarly we compute a crude approximation to each column of $\|A_H e_i\|_\infty$,

$$h_i^2 = \|A_{\mathcal{C},\mathcal{C}} - A_{\mathcal{C},\mathcal{F}} \operatorname{diag}(B_{\mathcal{F},\mathcal{F}}^{(0)}) A_{\mathcal{F},\mathcal{C}}\|_\infty$$

and use the associated diagonal matrix H with h_i on the main diagonal for scaling, i.e. we consider the diagonally scaled problem

$$\begin{pmatrix} G A_{\mathcal{F},\mathcal{F}} G & G(I - A_{\mathcal{F},\mathcal{F}} B_{\mathcal{F},\mathcal{F}}) G^{-1} (G A_{\mathcal{F},\mathcal{C}} H^{-1}) \\ (H^{-1} A_{\mathcal{C},\mathcal{F}} G) G^{-1} (I - B_{\mathcal{F},\mathcal{F}}^\top A_{\mathcal{F},\mathcal{F}}) G & H^{-1} A_H H^{-1} \end{pmatrix}.$$

For each column of $G(I - A_{\mathcal{F},\mathcal{F}} B_{\mathcal{F},\mathcal{F}}) G^{-1}$ we minimize the norm of the residual

$$r_i = G(I - A_{\mathcal{F},\mathcal{F}} B_{\mathcal{F},\mathcal{F}}) e_i \frac{1}{g_i}.$$

For each column we require that

$$\|r_i\| \leqslant \min\{\sqrt{\tau}, \tau \|g_i e_i^\top A_{\mathcal{F},\mathcal{C}} H^{-1}\|_\infty\}.$$

The value $\sqrt{\tau}$ is used to make sure that the residual $\|r_i\|$ is small in any case, even if the i–th row of $A_{\mathcal{F},\mathcal{C}}$ is empty.

6 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 [31]. The experiments are performed on an Linux PC with Pentium III processor (1 GHz).

The algebraic multilevel method based on the adapted sparse approximate inverse preconditioner will be denoted by AMG–SPAI. We will compare this AMG with the classical AMG [26] (AMG–RS). For AMG–SPAI we will use the additive algebraic multigrid preconditioner (AAMG). For AMG–RS we will use the V–cycle preconditioner with one pre smoothing and one post smoothing step. In principle an additive variant could also be used [24]. As usual, Gauss–Seidel (and its adjoint) are used as smoothers. Both preconditioners are used in conjunction with the cg method [13].

Example 8 We consider the problem

$$
\begin{aligned}
-\varepsilon^2 u_{xx} - u_{yy} &= f \text{ in } [0,1]^2 \\
u &= g \text{ on } \partial[0,1]^2
\end{aligned}
$$

where ε strongly varies from 10^0 to 10^{-5}. For this problem we use the variational formulation and piecewise quadratic finite elements, cf. e.g. [5]. 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.
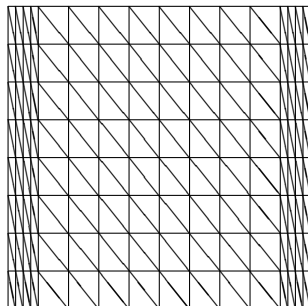
We consider linear systems of size 961, 3969 and 16129. Table 1 shows the number of cg iteration steps for both AMGs. We examine the aspect of scalability (with respect to the

Table 1: **Anisotropic Dirichlet, quadratic elements, cg iteration steps**

AMG	ε	ε versus scalability (system size)		
		961	3969	16129
	10^0	16	26	50
RS	10^{-3}	20	34	71
	10^{-5}	20	33	61
	10^0	17	19	22
SPAI	10^{-3}	13	16	18
	10^{-5}	17	19	21

system size) and robustness (with respect to ε). For this reason Table 1 denotes the number of iteration steps in both directions. Note that this problem has positive off–diagonal entries for all values of ε. This explains, why here AMG–RS does not scale that well. The dependence with respect to the variation of ε seems to be small for both approaches.

Next we examine the computational amount of work in flops (see Table 2). We scale the number of flops by the number of flops for one matrix–vector multiplication. The number have to be multiplied by 10^2 (see legend). The construction of the preconditioner scales perfectly for AMG–RS, a slight increase is observed for AMG–SPAI. Overall the construction of AMG–SPAI is more costly, which is caused by the computation of the sparse approximate inverse smoother while at the same time the Gauss–Seidel smoother in AMG–RS is for free. This additional amount of work is compensated by the smaller and bounded number of iteration steps. The results also show that not only the number of iterations is almost constant for AMG–SPAI. But the cost for the iterations also scales quite well.

Table 2: **Anisotropic Dirichlet, quadratic elements, $\frac{\text{flops} \cdot 10^2}{\#\,\text{flops}\,A\cdot x}$ (coarsening + cg)**

AMG	ε	ε versus scalability (system size)		
		961	3969	16129
	10^0	0.4+2.3	0.4+3.7	0.4+7.2
RS	10^{-3}	0.3+2.5	0.3+4.4	0.3+9.2
	10^{-5}	0.3+2.5	0.3+4.2	0.3+7.9
	10^0	3.5+1.0	4.0+1.1	4.4+1.3
SPAI	10^{-3}	2.1+0.6	2.7+0.7	3.2+0.9
	10^{-5}	2.1+0.7	2.8+0.9	3.3+1.0

For $n = 16129$ we now present the size of the coarse grid systems. We observe in Table 3 that the coarseing process for AMG–SPAI constructs coarse grids that are slightly smaller than for AMG–RS. For $\varepsilon = 0$ the coarsening process of AMG–RS is unsatisfactory due to the high number of positive off–diagonal entries.

Table 3: **Anisotropic Dirichlet, quadratic elements, coarsening for** $n = 16129$

AMG	ε	size of each level						
		1	2	3	4	5	6	7
	10^0	16129	8064	2204	2			
RS	10^{-3}	16129	8064	3200	574	224	96	32
	10^{-5}	16129	8064	3200	574	224	96	32
	10^0	16129	5655	1944	685	236	70	20
SPAI	10^{-3}	16129	5557	1826	601	173	97	60
	10^{-5}	16129	5569	1842	601	172	98	71

We note that the sparse approximate inverse smoothers that are constructed by AMG–SPAI are very sparse. In contrast to AMG–RS we need more space to store the sparse approximate inverse smoothers, while AMG–RS uses Gauss–Seidel smoothing.

Table 4: **Anisotropic Dirichlet, quadratic elements, density of the smoother**

AMG	ε	average number of nonzeros per row for each level						
		1	2	3	4	5	6	7
	10^0	6.7	8.6	10.7	13.6	17.4	17.8	10.5
SPAI	10^{-3}	4.7	5.3	6.4	7.2	10.6	6.7	5.0
	10^{-5}	4.7	5.2	6.4	7.2	7.8	4.3	2.6

We can see from Table 4 that the memory requirement for the sparse approximate inverse is close to the size of the original system (approximately 6 entries per row). So the overhead for storing the approximate inverse is small.

Example 9 We consider the same problem as in Example 8. But this time we use piecewise bilinear elements. Note that due to the anisotropy for small ε the associated stencil degenerates in the middle of the domain but the off–diagonal entries will **not** tend to zero (Figure 1). In this example, the associated matrix is an M–matrix if $\varepsilon = 1$, but for other values of ε the matrix has significant positive off–diagonal entries (as one can see from Figure 1).

We first examine the number of cg steps (see Table 5). Here we observe that AMG–RS scales perfectly for $\varepsilon = 1$, which is the M–matrix case. For $\varepsilon = 10^{-3}, 10^{-5}$ the number of iteration steps significantly grows. Also the method scales poorly as the problem size increases. In contrast to this, AMG–SPAI only shows a small increase in the number of iteration steps.

Next we examine the computational amount of work in flops (Table 6).

Table 7 shows the size of the sequence of coarse grid systems when using $n = 16129$. For $\varepsilon = 1$, A is an M–matrix and clearly AMG–RS is superior to AMG–SPAI. For the other cases the size of the coarse grid systems are comparable.

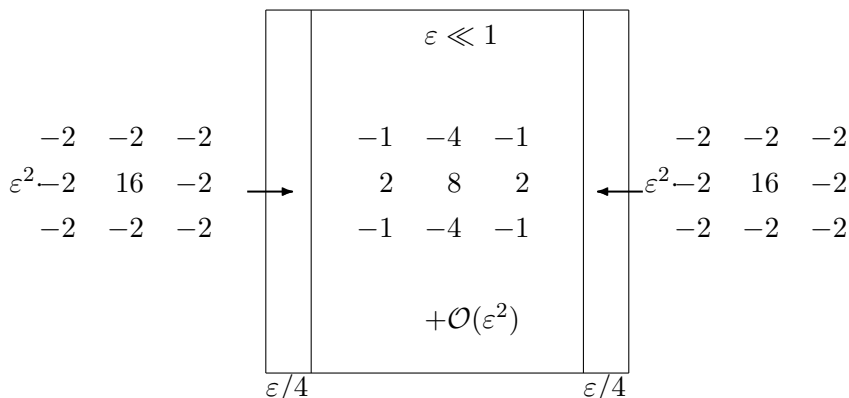Figure 1: **Bilinear stencils in different parts of the domain**

$$\varepsilon \ll 1$$

$$
\begin{array}{ccc}
-2 & -2 & -2 \\
\varepsilon^2 \leftharpoondown -2 & 16 & -2 \\
-2 & -2 & -2
\end{array}
\qquad \longrightarrow
\begin{array}{ccc}
-1 & -4 & -1 \\
2 & 8 & 2 \\
-1 & -4 & -1
\end{array}
\qquad \longleftarrow
\begin{array}{ccc}
-2 & -2 & -2 \\
\varepsilon^2 \leftharpoondown -2 & 16 & -2 \\
-2 & -2 & -2
\end{array}
$$

$$+\mathcal{O}(\varepsilon^2)$$

$$\varepsilon/4 \qquad\qquad \varepsilon/4$$

Table 5: **Anisotropic Dirichlet, bilinear elements, cg iteration steps**

		ε versus scalability (system size)		
AMG	ε	961	3969	16129
	10^0	13	15	16
RS	10^{-3}	50	107	212
	10^{-5}	50	104	211
	10^0	19	18	24
SPAI	10^{-3}	23	29	35
	10^{-5}	29	36	41

Table 6: **Anisotropic Dirichlet, bilinear elements, $\frac{\text{flops} \cdot 10^2}{\#\,\text{flops}\,A\cdot x}$ (coarsening + cg)**

		ε versus scalability (system size)		
AMG	ε	961	3969	16129
	10^0	2.2+1.0	2.4+ 1.1	2.4+ 1.2
RS	10^{-3}	2.4+4.8	2.5+10.0	2.5+20.0
	10^{-5}	2.4+4.8	2.5+10.0	2.5+20.0
	10^0	2.9+1.0	3.5+ 0.9	4.5+ 1.4
SPAI	10^{-3}	3.3+1.0	4.1+ 1.4	4.8+ 1.8
	10^{-5}	3.4+1.3	4.1+ 1.7	4.9+ 2.1

Table 7: **Anisotropic Dirichlet, bilinear elements, coarsening for** $n = 16129$

AMG	ε	size of each level						
		1	2	3	4	5	6	7
	10^0	16129	4258	1111	318	100	33	6
RS	10^{-3}	16129	6219	2626	256	75	7	
	10^{-5}	16129	6219	2626	256	75	7	
	10^0	16129	5923	2125	749	254	80	19
SPAI	10^{-3}	16129	5557	1831	591	169	25	1
	10^{-5}	16129	5594	1861	607	156	17	

The sparse approximate inverse smoothers constructed using AMG–SPAI are almost as sparse as in the case of quadratic elements, as shown in Table 8. Note that here the original system is also more dense (approximately 9 entries per row).

Table 8: **Anisotropic Dirichlet, bilinear elements, density of the smoother**

AMG	ε	average number of nonzeros per row for each level						
		1	2	3	4	5	6	7
	10^0	7.2	9.4	12.6	15.7	18.3	22.9	10.0
SPAI	10^{-3}	8.6	10.6	12.0	11.3	8.5	6.3	1.0
	10^{-5}	8.5	10.3	11.1	11.5	7.0	4.8	

7 Conclusions

We have presented a new algebraic multigrid method that is based on a special interplay between the selection of the fine grid nodes and the construction of the adapted sparse approximate inverse. One essential constraint that we have used is that $A\mathbf{1} \approx 0$. This limits of course the applicability of the new AMG. The numerical experiments show that for anisotropic problems the AMG scales well. This compensates the overhead that is needed for the setup of the preconditioner.

References

[1] S. T. Barnard, L. M. Bernardo, and H. D. Simon. An MPI implementation of the SPAI preconditioner on the T3E. Preprint LBNL–40794 UC–405, Lawrence Berkeley National Laboratory, Computer Science Directorate, 1997.

[2] 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.

[3] M. Benzi and M. Tůma. A sparse approximate inverse preconditioner for nonsymmetric linear systems. *SIAM J. Sci. Comput.*, 19(3):968–994, 1998.

[4] M. Bollhöfer and V. Mehrmann. Algebraic multilevel methods and sparse approximate inverses. *SIAM J. Matrix Anal. Appl.* to appear.

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

[6] O. Bröker and M. J. Grote. Sparse approximate inverse smoothers for geometric and algebraic multigrid. *Appl. Num. Math.*, 41(1):61–80, 2002.

[7] O. Bröker, M. J. Grote, C. Mayer, and A. Reusken. Robust parallel smoothing for multigrid via sparse approximate inverses. *SIAM J. Sci. Comput.*, 23(4):1396–1417, 2001.

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

[9] 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.

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

[11] G. Haase, U. Langer, and A. Meyer. The approximate dirichlet domain decomposition method. part I: An algebraic approach. *Computing*, 47:137–151, 1991.

[12] 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.

[13] M. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *J. Research Nat. Bur. Standards*, 49:409–436, 1952.

[14] T. Huckle. Sparse approximate inverses and applications. 1999 International Conference on Preconditioning Techniques for Large Sparse Matrix Problems in Industrial Applications, Minneapolis, June 1999. talk.

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

[16] 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.

[17] T. Manteuffel. An incomplete factorization technique for positive definite linear systems. *Math. Comp.*, 34:473–490, 1980.

[18] J. Meijerink and H. A. V. der Vorst. An iterative solution method for linear systems of which the coefficient matrix is a symmetric M–matrix. *Math. Comp.*, 31:148–162, 1977.

[19] G. Meurant. A multilevel AINV preconditioner. Technical report, CEA/DIF Bruyères le Chatel, 2000.

[20] G. Meurant. Numerical experiments with algebraic multilevel preconditioners. *Electr. Trans. Num. Anal.*, 12:1–65, 2001.

[21] G. Meurant. A multilevel AINV preconditioner. *Numer. Alg.*, 29:107–129, 2002.

[22] C. D. Meyer. *Matrix Analysis and Applied Linear Algebra.* SIAM Publications, 2000.

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

[24] H. Regler and U. Rüde. Layout optimization with algebraic multigrid methods. SFB-Report 342/11/93 A, TU München, Institut für Informatik, 1993.

[25] A. Reusken. A multigrid method based on incomplete Gaussian elimination. *Numer. Lin. Alg. w. Appl.*, 3:369–390, 1996.

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

[27] Y. Saad and M. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Statist. Comput.*, 7:856–869, 1986.

[28] Y. Saad and B. J. Suchomel. ARMS: An algebraic recursive multilevel solver for general sparse linear systems. Technical Report umsi–1999-107, University of Minnesota at Minneapolis, Dep. of Computer Science and Engineering, 1999.

[29] J. Staudacher. *Multigrid methods for matrices with structure and applications in image processing.* PhD thesis, TU Munich, Dept. of Computer Science, 2002.

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

[31] The MathWorks Inc. MATLAB version 6.1 release 12, 2001. a software program.

[32] P. Vanek, J. Mandel, and M.Brezina. Algebraic multigrid by smoothed aggregation for second order and forth order elliptic problems. *Computing*, 56:179–196, 1996.

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