# Algebraic multilevel preconditioners with projectors*

Yuri Kuznetsov† Konstantin Lipnikov‡

June 4, 2003

**Abstract**

A new multilevel algebraic preconditioner for symmetric positive definite matrices is proposed. The projection and smoothing steps typical for textbook multigrids are replaced by a special coarsening algorithm which is based on orthogonal projectors onto kernels of subdomain matrices. The preconditioner is compared numerically with a few other multigrid preconditioners.

## 1 Introduction

The main goal of this paper is to describe a new method for the solution of elliptic PDEs on unstructured highly distorted meshes. Iterative methods based on the multilevel principle are known to be the fastest solvers for these equations. The main idea of this principle is to combine the local exchange of information in point-wise iterative methods and the global propagation of information through coarse level systems with smaller number of unknowns. The main difference between multilevel methods lies in the definition of the coarse level unknowns and projection/interpolation operators between neighboring levels.

The results of numerical experiments reported in [2] show that the algebraic multigrid (AMG) method by Stüben and Ruge [8, 6] results in relatively good and robust convergence rate for a wide range of problems. The AMG is based on the concept of strongly coupled nodes. The coarse grid nodes are selected in such a manner that as many strong coupling as possible are taken into consideration.

The theory of the AMG method uses the assumption that the original stiffness matrix is a symmetric positive definite M-matrix. A violation of this assumption does not necessary make a negative impact on the convergence rate. However, our numerical experiments show that for the case of problems with fine heterogeneous materials and highly distorted meshes, the convergence rate of the AMG method deteriorates strongly.

In this paper we analyze numerically a new multilevel (ML) preconditioner based on a two-level preconditioner proposed in [4]. The theory of our approach is based on the

*The research work has been supported by LACSI.

†Department of Mathematics, University of Houston, kuz@math.uh.edu

‡Los Alamos National Laboratory, MS B284, lipnikov@t7.lanl.gov

assumption of a lower effective material anisotropy. This assumption is satisfied in many applications where the computational mesh is somehow adapted to the problem peculiarities.

Unlike the AMG method, the unknowns in the multilevel structure we use can not be associated with a sequence of nested grids. Assuming that a hierarchical sequence of non-overlapping partitions of the original mesh is given, these unknowns represent mean values of a function over clusters of elements. From this viewpoint, our method is slightly similar to the AMG-type methods based on the concept of smooth aggregations [10, 9]. However, instead of the projection and smoothing steps typical for the AMG-type multigrids, we use a special coarsening algorithm which is based on orthogonal projectors onto kernels of subdomain matrices.

We compare numerically the ML and AMG preconditioners for the case of highly distorted meshes and fine heterogeneous materials. We consider two discretization methods based on mimetic finite differences [7] and mixed finite elements. Both discretization have the same convergence rate [1, 11] and result in stiffness matrices with the same sparsity structure. However, the multilevel preconditioner shows a relatively good and robust convergence rate for both discretizations, unlike the AMG preconditioner, which results in a very slow convergence for the case of finite element discretization on the Kershaw grids [3].

The paper outline is as follows. In Section 2, we describe a two-level preconditioner and analyze its spectral properties. In Section 3, we describe the multilevel preconditioner. Implementation details of the ML preconditioner are presented in Appendix. In Section 4, the multilevel preconditioner is compared numerically with V and W-cycles of the AMG method.

# 2 Two-level preconditioner

Let us consider a model diffusion problem:

$$\begin{aligned} -\mathrm{div}(\boldsymbol{K}(\boldsymbol{x})\nabla u) &= f \quad \text{in} \quad \Omega, \\ u &= 0 \quad \text{on} \quad \partial\Omega, \end{aligned} \tag{1}$$

where $\Omega \in \Re^2$ is a polygonal domain and $\boldsymbol{K}(\boldsymbol{x})$ is a symmetric positive definite matrix for all $\boldsymbol{x} \in \Omega$.

Let $\Omega_h$ be a non-overlapping partition of $\Omega$ onto polygonal elements $e_i$,

$$\Omega_h = \bigcup_{i=1}^{m} e_i.$$

If we discretize problem (1) with the $P_1$ finite element method, the discretized problem can be formulated as a linear system

$$Au = f, \quad A = A^T > 0. \tag{2}$$

In many applications, diffusion problem (1) is formulated as a system of first-order partial differential equations which define the flux and describe the mass conservation law. In this

case the discretized problem results in a saddle point system which can be reduced to problem (2) by applying the standard hybridization arguments.

We assume that the matrix $A$ can be written in an assembled form by

$$A = \sum_{i=1}^{m} \mathcal{N}_i \, A_i \, \mathcal{N}_i^T \tag{3}$$

where $A_i$ are the elemental matrices and $\mathcal{N}_i$ are the corresponding assembling matrices. Therefore, a natural choice for a preconditioner $B$ could be

$$B = \sum_{i=1}^{m} \mathcal{N}_i \, B_i \, \mathcal{N}_i^T \tag{4}$$

with the elemental matrices $B_i$ where $B_i$ is in some sense close to $A_i$ and $\ker B_i = \ker A_i$.

In order to construct the elemental matrices $B_i$ we consider the generalized eigenvalue problem

$$A_i w = \lambda M_i w \tag{5}$$

where $M_i$ is a positive definite diagonal matrix. Let $W_i$ be the matrix of $M_i$-orthogonal eigenvectors, $W_i = [w_i^{(1)}, \ldots, w_i^{(s_i)}]$, where $s_i$ is the size of matrix $A_i$. Furthermore, let $\Lambda_i$ be the matrix of eigenvalues:

$$\Lambda_i = \mathrm{diag}\{\lambda_i^{(1)}, \lambda_i^{(2)}, \ldots, \lambda_i^{(s_i)}\}, \qquad \lambda_i^{(1)} \leq \lambda_i^{(2)} \leq \ldots \leq \lambda_i^{(s_i)}.$$

Then, the spectral decomposition of $A_i$ is

$$A_i = M_i W_i \Lambda_i W_i^T M_i. \tag{6}$$

Let $e_i$ be an interior element of partition $\Omega_h$, i.e. $\partial e_i \cap \partial\Omega = \emptyset$. Then, the lowest eigenvalue is zero, $\lambda_i^{(1)} = 0$, and the corresponding eigenvector is $w_i^{(1)} = \sqrt{\sigma_i}(1, 1, \ldots, 1)^T$ where $\sigma_i$ is such that $[w_i^{(1)}]^T M_i w_i^{(1)} = 1$. Define

$$B_i = M_i \, W_i \, \tilde{\Lambda}_i \, W_i^T \, M_i \tag{7}$$

where

$$\tilde{\Lambda}_i = \mathrm{diag}\{0, 1, 1, \ldots, 1\}.$$

The $M_i$-orthogonality of the eigenvectors implies that

$$B_i = M_i W_i W_i^T \, M_i - M_i \, w_i^{(1)} [w_i^{(1)}]^T \, M_i = M_i - M_i P_i M_i$$

where $P_i = w_i^{(1)} [w_i^{(1)}]^T$ is the one-rank matrix,

$$P_i = \sigma_i \begin{bmatrix} 1 & 1 & \ldots & 1 \\ 1 & 1 & \ldots & 1 \\ \ldots & \ldots & \ldots & \ldots \\ 1 & 1 & \ldots & 1 \end{bmatrix}.$$

It is easy to check that $(P_i M_i)^2 = P_i M_i$, i.e. $P_i M_i$ is the $M_i$-orthogonal projector onto kernel of both $A_i$ and $B_i$. The definition of matrix $B_i$ implies that

$$\lambda_i^{(2)}(B_i\, x,\, x) \le (A_i\, x,\, x) \le \lambda_i^{(s_i)}\,(B_i\, x,\, x) \qquad \forall x \in \Re^{s_i} \tag{8}$$

and the ratio $\lambda_i^{(s_i)}/\lambda_i^{(2)}$ depends only on the element shape and tensor anisotropy. Hereafter we assume that the effective material anisotropy is small, i.e. $\lambda_i^{(s_i)}/\lambda_i^{(2)}$ is $O(1)$. The more restrictive assumptions are that the diffusion tensor is isotropic and constant inside each mesh element and the mesh elements are shape-regular. In this case, the ratio $\lambda_i^{(s_i)}/\lambda_i^{(2)}$ is again $O(1)$. In addition to that, the global preconditioner $B$ requires a proper scaling of lower and upper bounds in (8). It can be achieved by letting $M_i$ be the main diagonal of $A_i$. Then, $\lambda_i^{(2)} = O(1)$ and $\lambda_i^{(s_i)} = O(1)$.

Let $e_i$ be a near-boundary element, i.e. $\partial e_i \cap \partial\Omega \neq \emptyset$. The Dirichlet boundary condition implies that all eigenvalues of (5) are of the same order of magnitude. Therefore, the choice

$$B_i = M_i, \qquad M_i = \mathrm{diag}\{A_i\},$$

results in inequalities similar to (8) with lower and upper bounds equal to $\lambda_i^{(1)}$ and $\lambda_i^{(s_i)}$, respectively. Again we assume that $\lambda_i^{(s_i)}/\lambda_i^{(1)} = O(1)$.

**Lemma 1** *Under the above assumptions, the matrix $B$ is spectrally equivalent to the matrix $A$, i.e.*

$$c_1(B\, x,\, x) \le (A\, x,\, x) \le c_2(B\, x,\, x) \qquad \forall x \in \Re^n,$$

*where constants $c_1$ and $c_2$ are independent of the mesh and jumps in the diffusion tensor.*

The lemma is proved with standard arguments of the superelement analysis (see, e.g. [5]).

A two-level preconditioner comes naturally from the analysis of the structure of the matrix $B$. Let $\tilde{m}$ be the number of interior cells which are enumerated from 1 to $\tilde{m}$. Then, definitions (4) and (7) give

$$\begin{aligned} B = \sum_{i=1}^{m} \mathcal{N}_i\, B_i\, \mathcal{N}_i^T &= \sum_{i=1}^{\tilde{m}} \mathcal{N}_i (M_i - M_i\, P_i\, M_i) \mathcal{N}_i^T + \sum_{i=1}^{m-\tilde{m}} \mathcal{N}_i M_i \mathcal{N}_i^T \\ &= M - \sum_{i=1}^{\tilde{m}} \mathcal{N}_i\, M_i\, P_i\, M_i\, \mathcal{N}_i^T \equiv M - C \end{aligned} \tag{9}$$

where $M$ is the diagonal matrix and $C$ is a matrix with non-negative entries. It is easy to check that $B$ is an M-matrix with a weak diagonal dominance. Therefore, the algebraic multigrid, like the AMG by Stüben and Ruge [8, 6], may result in a better preconditioner if it is based on entries of matrix $B$ rather than on entries of matrix $A$. We shall address this issue in more details in the future work.

Let us consider a system

$$Bv = (M - C)v = g \tag{10}$$

and define

$$\hat{v}_i = \left[w_i^{(1)}\right]^T M_i \mathcal{N}_i^T v \qquad \text{and} \qquad \hat{g}_i = \left[w_i^{(1)}\right]^T M_i \mathcal{N}_i^T M^{-1} g.$$

If $M_i$ were the mass matrix for element $e_i$, then $\hat{v}_i$ would be the mean value of a finite element function $v^h$ over the element, i.e.

$$\hat{v}_i = \frac{1}{|e_i|} \int_{e_i} v^h \, \mathrm{d}x$$

where $|e_i|$ denotes the area of $e_i$. In the general case, $\hat{v}_i$ is a weighted mean value induced by matrix $M_i$. Let us multiply equation (10) by $[w_i^{(1)}]^T M_i \mathcal{N}_i^T M^{-1}$, $i = 1, \ldots, \tilde{m}$. We get the following system for the mean values:

$$(I - \hat{Q})\hat{v} = \hat{g}, \qquad \hat{g} \in \Re^{\tilde{m}},$$

where

$$\hat{Q} = \{\hat{q}_{ij}\}_{i,j=1}^{\tilde{m}} \qquad \text{and} \qquad \hat{q}_{ij} = [w_i^{(1)}]^T M_i \mathcal{N}_i^T M^{-1} \mathcal{N}_j M_j \, w_j^{(1)}. \tag{11}$$

Let $\hat{S} = I - \hat{Q}$. It is easy to show that the matrix $\hat{S}$ preserves many properties of the matrix $B$. In particular, $\hat{S}$ is symmetric positive definite M-matrix. Let us assume for a moment that entries of the matrix $S^{-1} = Z = \{z_{ij}\}_{i,j=1}^{\tilde{m}}$ are known. Then, we can write down the explicit formula for $B^{-1}$,

$$B^{-1} = M^{-1} + M^{-1} \left[ \sum_{i,j=1}^{\tilde{m}} z_{ij} \, \mathcal{N}_i \, M_i \, w_i^{(1)} \, [w_j^{(1)}]^T \, M_j \, \mathcal{N}_j^T \right] M^{-1}.$$

**Theorem 1 (Two-level preconditioner)** *Let matrix $R = \{r_{ij}\}_{i,j=1}^{m}$ be spectrally equivalent to the matrix $Z$ and*

$$B_R^{-1} = M^{-1} + M^{-1} \left[ \sum_{i,j=1}^{\tilde{m}} r_{ij} \, \mathcal{N}_i \, M_i \, w_i^{(1)} \, [w_j^{(1)}]^T \, M_j \, \mathcal{N}_j^T \right] M^{-1}.$$

*Then, matrix $B_R$ is spectrally equivalent to the matrix $B$.*

The proof of the theorem is based on simple linear algebra and properties of spectrally equivalent matrices. It can be found in [4]. In practice, implementation of preconditioner $B_R$ means that instead of solving system $(I - \hat{Q})\,\hat{v} = \hat{g}$ we compute $\hat{v} = R\,\hat{g}$.

**Remark 1** *Note that the number of mesh elements may be bigger than the size of the original problem. The typical example here is the $P_1$ finite element discretization on a triangular grid $\Omega_h$. In this case, the triangular partition $\Omega_h$ may be replaced by a coarser partition $\tilde{\Omega}_h$, for example, by uniting triangles into quadrilaterals.*

# 3 Multilevel preconditioner

Let us define a hierarchical sequence of non-overlapping partitions

$$\Omega_{l,h} = \bigcup_{i=1}^{m_l} \Omega_{l,i}, \qquad l = 1, \ldots, L, \tag{12}$$

where

$$\Omega_{1,i} = e_i \qquad \text{and} \qquad \Omega_{l,i} = \bigcup_{j \in \mathfrak{S}_l(i)} \Omega_{l-1,j}, \qquad l > 1.$$

Here $\mathfrak{S}_l(i)$ denotes a collection of indexes of subdomains from level $l-1$ whose union is the subdomain $\Omega_{l,i}$. We assume that subdomains $\Omega_{l,i}$ are simply-connected.

Let us denote the two-level preconditioner introduced in Section 2 by $A_1$, i.e. $A_1 = B$, and rewrite formula (9) as follows:

$$A_1 = M - \sum_{i=1}^{\tilde{m}_1} \mathcal{N}_{1,i}\, M_{1,i}\, P_{1,i}\, M_{1,i}\, \mathcal{N}_{1,i}^T, \tag{13}$$

where $M_{1,i}$, $P_{1,i}$ and $\mathcal{N}_{1,i}$ are new notations for the matrices $M_i$, $P_i$ and $\mathcal{N}_i$, respectively.

For every partition in hierarchy (12), the original stiffness matrix $A$ can be written in the assembled form:

$$A = \sum_{i=1}^{m_l} \mathcal{N}_{l,i}\, A_{l,i}\, \mathcal{N}_{l,i}^T, \qquad A_{l,i} = \sum_{j \in \mathfrak{S}_l(i)} \hat{\mathcal{N}}_{l,j}\, A_{l-1,j}\, \hat{\mathcal{N}}_{l,j}^T \tag{14}$$

where $A_{l,i}$ are the subdomain stiffness matrices and $\mathcal{N}_{l,i}$, $\hat{\mathcal{N}}_{l,i}$ are the assembling matrices. Similar to (14), we can rewrite matrix $M$ in the assembled form:

$$M = \sum_{i=1}^{m_l} \mathcal{N}_{l,i}\, M_{l,i}\, \mathcal{N}_{l,i}^T, \qquad M_{l,i} = \sum_{j \in \mathfrak{S}_l(i)} \hat{\mathcal{N}}_{l,j}\, M_{l-1,j}\, \hat{\mathcal{N}}_{l,j}^T. \tag{15}$$

In the previous section we introduced the $M_{1,i}$-orthogonal projectors $P_{1,i}$, $i = 1, \ldots, \tilde{m}_1$. Now, we can repeat the same speculations for every partition in hierarchy (12). Let $P_{l,i}$ be the $M_{l,i}$-orthogonal projector onto kernel of the subdomain matrix $A_{l,i}$ from (14), i.e. $P_{l,i} = w_{l,i}^{(1)}[w_{l,i}^{(1)}]^T$ where $w_{l,i}^{(1)} = \sqrt{\sigma_{l,i}}(1, 1, \ldots, 1)^T$ represents the lowest eigenmode of $A_{l,i}$. It is obvious that $P_{l,i}$ is a one-rank matrix. Let us define matrices $A_l$, $l = 2, \ldots, L$, as follows:

$$A_l = M - \sum_{i=1}^{\tilde{m}_l} \mathcal{N}_{l,i}\, M_{l,i}\, P_{l,i}\, M_{l,i}\, \mathcal{N}_{l,i}^T. \tag{16}$$

At the moment, the matrices $A_l$, $l = 1, \ldots, L$, have the same size and rank. However, matrices

$$C_l = A_l - M = -\sum_{i=1}^{\tilde{m}_l} \mathcal{N}_{l,i} M_{l,i} P_{l,i} M_{l,i} \mathcal{N}_{l,i}^T, \quad l = 1, \ldots, L,$$

have different ranks. The rank of the matrix $C_l$ equals to the number of involved projectors, i.e. $\mathrm{rank}(C_l) = \tilde{m}_l$. Moreover, $\tilde{m}_l < \tilde{m}_{l-1}$. This feature will be exploited in the multilevel algorithm formulated below.

Let $V_0 \equiv \Re^n$ and $V_l$ be the images of matrices $C_l$, i.e. $V_l = \mathrm{im}(C_l)$, $l = 1, \ldots, L$. Then, the definition of matrix $C_l$ implies that

$$V_l = \{x \in V_0 \colon \; x = \sum_{i=1}^{\tilde{m}_l} a_i \mathcal{N}_{i,l} M_{l,i} w_{l,i}^{(1)}, \quad a_i \in \Re^1\}. \tag{17}$$

Since $w_{l,i}^{(1)} = \sqrt{\sigma_{l,i}}(1, 1, \ldots, 1)^T$ and the partitions $\Omega_{l,h}$ are hierarchical, the spaces $V_l$, $l = 0, \ldots, L$, are nested:

$$V_L \subset V_{L-1} \subset \ldots \subset V_0.$$

**Lemma 2** *Let the partitions $\Omega_{l,h}$, $l = 1, \ldots, L$, be quasi-uniform with the shape-regular subdomains, i.e.* $\mathrm{diam}(\Omega_{l,i}) \sim 2^{l-1}h$. *Then*

$$\frac{c_1}{2^l}(A_l x,\, x) \le (A_{l-1} x,\, x) \le c_2 (A_l x,\, x) \qquad \forall x \in V_{l-1}, \quad l = 1, \ldots, L. \tag{18}$$

*where $A_0 \equiv A$ and constants $c_1$ and $c_2$ are independent of $l$ and the discretization parameter $h$.*

The proof of the lemma is again based on the standard superelement analysis (see, e.g. [5]) and therefore is omitted here.

Let $B_{l+1}$ be a multilevel preconditioner for level $l+1$. We assume that on the coarsest level $B_L = A_L$. On the finer level $l$, the preconditioner $B_l$ is given by a recursive algorithm for computing $\tilde{r} = B_l^{-1} r$.

**Algorithm 1 (Multilevel Preconditioner)**

1. *take a special initial guess*

$$x^0 = M^{-1} r, \qquad r \in V_{l-1};$$

2. *enter the subspace $V_l$:*

$$\xi_l^0 = A_l x^0 - r = -C_l M^{-1} r, \qquad \xi_l^0 \in V_l;$$

3. *set $y_l^0 = -\xi_l^0$ and perform $\beta$ Chebyshev iterations in the subspace $V_l$:*

$$\begin{aligned} y_l^i &= y_l^{i-1} - \gamma_{l,i}\, C_l\, B_{l+1}^{-1}\, \xi_l^{i-1}, \\ \xi_l^i &= \xi_l^{i-1} - \gamma_{l,i}\, A_l\, B_{l+1}^{-1}\, \xi_l^{i-1}, \qquad i = 1, \ldots, \beta; \end{aligned}$$

4. *leave the subspace $V_l$:*

$$\tilde{r} = M^{-1}(y_l^\beta + r).$$

Note that on Step 3, the preconditioner $B_{l+1}$ acts on a residual $\xi_l^i \in V_l$. This is in the agreement with Step 1 where $r \in V_{l-1}$.

The Algorithm 1 fits into the standard multigrid scheme. The projection to a coarser grid is replaced by the subspace entering (Steps 1 and 2). The interpolation to a finer grid is replaced by the subspace leaving (Step 4). Note that Step 2 has some smoothing properties coming from smoothing properties of matrix $C_l$ which computes mean values over subdomains. The Chebyshev iterations provides the best polynomial approximation to $A_l^{-1}$ in subspace $V_l$. Indeed, Algorithm 1 results in the following implicit formula for the preconditioner $B_l$:

$$B_l^{-1} = (I - T_l)A_l^{-1}$$

where

$$T_l = (I - M^{-1}A_l)\,\mathcal{P}_{l,\beta}(B_{l+1}^{-1}A_l)\,(I - M^{-1}A_l)$$

and $\mathcal{P}_{l,\beta}(\cdot)$ is the Chebyshev polynomial of degree $\beta$,

$$\mathcal{P}_{l,\beta}(t) = \prod_{i=1}^{\beta}(1 - \gamma_{l,i}t).$$

The Chebyshev parameters $\gamma_{l,i}$ are chosen such that to minimize the spectral radius of $\mathcal{P}_{l,\beta}(B_{l+1}^{-1}A_l)$. By the definition of matrices $M$ and $A_l$, the spectral radius of $I - M^{-1}A_l$ is less than 1. Therefore, Lemma 2 and the minimization properties of the Chebyshev polynomial imply that matrices $B_l$ and $A_l$ are spectrally equivalent if $\beta$ is sufficiently large.

On the other hand, the optimal arithmetical complexity of the method is achieved for sufficiently small $\beta$. For instance, the uniform hierarchical coarsening of a square grid results in $\tilde{m}_l/\tilde{m}_{l+1} \approx 4$. Therefore, the optimal arithmetical complexity is achieved for $\beta \leq 3$. The numerical study of Algorithm 1 shows that it is robust for a large number of multigrid levels. This feature of the algorithm is still an open question.

# 4 Numerical experiments

In this section we compare the multilevel preconditioner with the algebraic multigrid preconditioner written by J.Ruge, K.Stüben, and R.Hempel (release 1.5, 1990).

Let us consider a test problem

$$\begin{aligned} -\mathrm{div}(\boldsymbol{K}(\boldsymbol{x})\nabla u) + c\,u &= 1 \quad \text{in} \quad \Omega, \\ u &= 0 \quad \text{on} \quad \Gamma_D, \\ (\boldsymbol{K}(\boldsymbol{x})\nabla u)\,\boldsymbol{n} &= 0 \quad \text{on} \quad \Gamma_N, \end{aligned}$$

where $c = 0.00225$ and $\Omega$ is a unit square with boundary $\partial\Omega$ divided into two parts, $\Gamma_D$ and $\Gamma_N$. Let $\Gamma_D$ be the top side of $\Omega$.

We consider three different diffusion tensors: $\boldsymbol{K}_1(\boldsymbol{x}) = \boldsymbol{I}$,

$$\boldsymbol{K}_2(\boldsymbol{x}) = \begin{cases} 10^3, & \text{when} \quad \boldsymbol{x} \in (0;\,0.5)^2 \cup (0.5;\,1)^2, \\ 1, & \text{otherwise}, \end{cases}$$

and $\boldsymbol{K}_3(\boldsymbol{x}) = \boldsymbol{K}_2(\boldsymbol{x})\,\psi(i)$, where $\psi(i)$ a random function which is constant on each mesh element $e_i$ and takes arbitrary values between 1 and $10^3$.

We compare the ML and AMG preconditioners on a sequence of uniformly refined grids shown in Fig. 1. The first grid is constructed from a reference square grid by the mapping

$$x(\xi,\eta) = \xi + 0.1\,\sin(2\pi\eta)\,\sin(2\pi\xi), \qquad y(\xi,\eta) = \eta + 0.1\,\sin(2\pi\eta)\,\sin(2\pi\xi).$$

The second one is the piece-wise smooth Kershaw grid proposed in [3]. The Kershaw grid is more close to a grid we can expect in real applications. Different orientation of mesh lines catches a possible difference in material properties. We consider two approaches to a problem discretization: the mimetic finite differences (MFD) [7] and the lowest order Raviart-Thomas mixed finite elements (MFE). Both discretizations result in saddle point problems which can be reduced to problems with symmetric positive definite matrices by applying the standard hybridization arguments. The unknowns are associated with mesh edges and called the Lagrange multipliers in the finite element community. The discretizations provide the same accuracy [1, 11]; however, the mimetic discretizations results usually in matrices with less number of positive off-diagonal entries than the finite element discretizations.

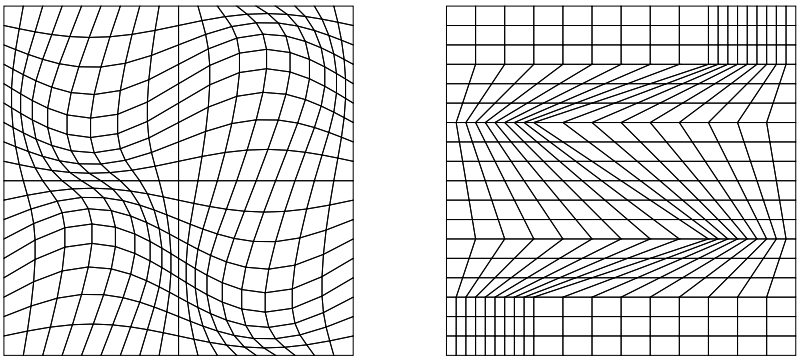

Figure 1: Examples of smooth and Kershaw grids.

The results of numerical experiments are shown in Tables 1 and 2. The experiments have been performed with a modified version of Algorithm 1. We performed one pre-smoothing and one post-smoothing Gauss-Seidel iteration on level $l = 1$. We used three Chebyshev iterations, i.e. $\beta = 3$.

In the tables, we present the number of PCG iterations and the total CPU time for the methods. Note that the initialization time of the AMG method is problem dependent and has to be included in the total solution time. The PCG iterations were stopped when the initial Euclidean norm of the residual has been decreased by a factor of $10^6$. The size of a coarse grid problem was 144 in all experiments.

On smooth grids, the AMG preconditioner requires about 50% less total time than the ML preconditioner in the case of the mimetic FD discretization. However, both preconditioners are competitive in the case of the MFE discretization. Indeed, in this case the matrix $A$ has

Table 1: Arithmetical complexity of preconditioners on smooth grids.

| $\boldsymbol{K}$ | | mimetic FD discretization | | | | | | mixed FE discretization | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | ML(3) | | V(1,1) | | W(1,1) | | ML(3) | | V(1,1) | | W(1,1) | |
| | $1/h$ | # | time | # | time | # | time | # | time | # | time | # | time |
| $\boldsymbol{K}_1$ | 12 | 11 | 0.02 | 8 | 0.01 | 8 | 0.02 | 14 | 0.02 | 11 | 0.03 | 8 | 0.02 |
| | 24 | 14 | 0.09 | 10 | 0.08 | 9 | 0.12 | 17 | 0.12 | 12 | 0.12 | 11 | 0.13 |
| | 48 | 17 | 0.53 | 10 | 0.44 | 10 | 0.76 | 20 | 0.70 | 17 | 0.69 | 15 | 0.95 |
| | 96 | 20 | 2.94 | 12 | 2.21 | 11 | 4.17 | 21 | 3.54 | 25 | 4.04 | 21 | 7.71 |
| | 192 | 23 | 15.0 | 13 | 9.57 | 13 | 21.8 | 26 | 19.3 | 37 | 22.4 | 34 | 54.1 |
| $\boldsymbol{K}_2$ | 12 | 11 | 0.02 | 8 | 0.01 | 6 | 0.02 | 14 | 0.03 | 11 | 0.01 | 8 | 0.01 |
| | 24 | 15 | 0.09 | 11 | 0.08 | 9 | 0.08 | 17 | 0.12 | 13 | 0.09 | 11 | 0.13 |
| | 48 | 18 | 0.57 | 12 | 0.49 | 10 | 0.75 | 21 | 0.73 | 17 | 0.59 | 15 | 0.97 |
| | 96 | 22 | 3.18 | 12 | 2.17 | 12 | 4.33 | 24 | 3.86 | 26 | 3.70 | 22 | 7.54 |
| | 192 | 26 | 16.7 | 16 | 10.9 | 15 | 23.9 | 29 | 20.3 | 42 | 22.9 | 35 | 56.7 |
| $\boldsymbol{K}_3$ | 12 | 11 | 0.03 | 10 | 0.02 | 9 | 0.02 | 14 | 0.03 | 8 | 0.02 | 8 | 0.02 |
| | 24 | 14 | 0.11 | 11 | 0.08 | 10 | 0.09 | 16 | 0.12 | 12 | 0.08 | 11 | 0.13 |
| | 48 | 21 | 0.71 | 14 | 0.49 | 13 | 0.82 | 21 | 0.72 | 17 | 0.70 | 15 | 0.97 |
| | 96 | 25 | 3.98 | 19 | 2.95 | 16 | 5.57 | 25 | 3.99 | 23 | 4.14 | 20 | 6.74 |
| | 192 | 28 | 20.5 | 22 | 14.1 | 20 | 32.1 | 29 | 20.3 | 25 | 23.1 | 23 | 40.1 |

more positive off-diagonal entries but the good spectral properties of the AMG preconditioner are guaranteed only for M-matrices.

On the Kershaw grids, the AMG preconditioner is more than twice better than our preconditioner in the case of the MFD discretization. In the case of the MFE discretization, the comparison is in favor of the ML preconditioner. The replacement of V-cycle by W-cycle improves the convergence rate but drastically increases the total computational time.

# 5 Conclusion

We compared numerically the ML and AMG preconditioners. The ML preconditioner resulted in relatively good and robust convergence rate for all test problems, specially problems on highly disturbed meshes. It is pertinent to note that the geometric nature of the ML preconditioner is transparent for its parallel implementation.

# References

[1] M. Berndt, D. Moulton, K. Lipnikov, and M. Shashkov. Convergence of mimetic finite difference discretizations of the diffusion equation. *East-West J. Numer. Anal.*, 9(4):253—284, 2001.

Table 2: Arithmetical complexity of preconditioners on Kershaw grids.

| $\boldsymbol{K}$ | | mimetic FD discretization | | | | | | mixed FE discretization | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | ML(3) | | V(1,1) | | W(1,1) | | ML(3) | | V(1,1) | | W(1,1) | |
| | $1/h$ | # | time | # | time | # | time | # | time | # | time | # | time |
| $\boldsymbol{K}_1$ | 12 | 17 | 0.04 | 12 | 0.02 | 11 | 0.03 | 17 | 0.03 | 16 | 0.02 | 11 | 0.02 |
| | 24 | 26 | 0.17 | 16 | 0.10 | 16 | 0.18 | 23 | 0.16 | 27 | 0.18 | 19 | 0.18 |
| | 48 | 32 | 1.03 | 18 | 0.65 | 16 | 1.38 | 29 | 0.94 | 49 | 1.51 | 29 | 2.08 |
| | 96 | 40 | 6.00 | 20 | 2.88 | 18 | 7.74 | 34 | 5.22 | 101 | 13.4 | 63 | 24.5 |
| | 192 | 49 | 32.5 | 24 | 13.8 | 19 | 39.6 | 42 | 28.2 | 209 | 110. | 132 | 247. |
| $\boldsymbol{K}_2$ | 12 | 23 | 0.04 | 13 | 0.02 | 13 | 0.03 | 22 | 0.04 | 22 | 0.03 | 11 | 0.02 |
| | 24 | 35 | 0.22 | 20 | 0.13 | 14 | 0.18 | 30 | 0.19 | 33 | 0.21 | 17 | 0.20 |
| | 48 | 38 | 1.19 | 23 | 0.80 | 18 | 1.52 | 36 | 1.12 | 57 | 1.71 | 33 | 2.48 |
| | 96 | 49 | 7.27 | 25 | 3.89 | 19 | 8.72 | 43 | 6.40 | 114 | 15.0 | 64 | 24.7 |
| | 192 | 58 | 38.1 | 28 | 17.8 | 21 | 43.5 | 51 | 33.6 | 232 | 123. | 125 | 232. |
| $\boldsymbol{K}_3$ | 12 | 23 | 0.04 | 18 | 0.02 | 16 | 0.03 | 23 | 0.04 | 13 | 0.02 | 11 | 0.02 |
| | 24 | 34 | 0.22 | 21 | 0.13 | 19 | 0.20 | 30 | 0.20 | 27 | 0.16 | 21 | 0.23 |
| | 48 | 44 | 1.39 | 30 | 0.96 | 25 | 1.68 | 39 | 1.22 | 79 | 2.21 | 32 | 2.19 |
| | 96 | 60 | 8.65 | 40 | 5.64 | 32 | 12.9 | 49 | 7.23 | 151 | 19.4 | 92 | 36.0 |
| | 192 | 66 | 43.1 | 43 | 25.1 | 39 | 80.2 | 57 | 37.6 | 299 | 157. | 179 | 347. |


[2] T. Grauschopf, M. Griebel, and H. Regler. Additive multilevel preconditioners based on bilinear interpolation, matrix dependent geometric coarsening and algebraic multigrid coarsening for second order elliptic PDEs. *Applied Numerical Mathematics*, 23:63–96, 1997.

[3] D. Kershaw. Differencing of the diffusion equation in the Lagrangian hydrodynamic codes. *J. Comp. Phys.*, 39:375–395, 1981.

[4] Y. Kuznetsov. Two-level preconditioners with projectors for unstructured grids. *Russian J. Numer. Anal. Math. Modelling*, 15:247–256, 2000.

[5] Y. Kuznetsov and S.Y.Maliassov. Substructuring preconditioners for nonconforming finite element approximations of second order elliptic problems with anisotropy. *Russian J. Numer. Anal. Math. Modelling*, 10:511–533, 1995.

[6] J. Ruge and K. Stüben. Algebraic multigrid (AMG). In S. McCormick, editor, *Multigrid Methods*, volume 3 of *Frontiers in Applied Mathematics*, pages 73–130. SIAM, Philadelhia, PA, 1987.

[7] M. Shashkov. *Conservative Finite-Difference Methods on General Grids*. CRC Press, Boca Raton, 1996.

[8] K. Stüben. Algebraic multigrid (AMG): experiences and comparisons. *Appl. Math. Comput.*, 13:419–452, 1983.

[9] P. Vanêk, M. Brezina, and J. Mandel. Convergence of algebraic multigrid based on smoothed aggregation. *NM*, 88:559–579, 2001.

[10] P. Vanêk, J. Mandel, and M. Brezina. Algebraic multigrid on unstructured meshes. Technical Report UCD/CCM Report 34, Center for Computational Mathematics, University of Colorado at Denver, 1994.

[11] J. Wang and T. Mathew. Mixed finite element methods over quadrilaterals. In I. Dimov, B. Sedov, and P. Vassilevski, editors, *Third International Conference on Advances in Numerical Methods and Applications*, pages 203–214. World Scientific, 1994.

# Appendix

In this section we describe an efficient implementation of Algorithm 1. Let $Q_l \equiv \Re^{\tilde{m}_l}$ and $\mathcal{K}_l \colon V_l \to Q_l$ be a bijective mapping between spaces $Q_l$ and $V_l$. We shall use small letters for vectors in the space $V_l$ (e.g., $\xi$, $r$, $z$, etc) and calligraphic capital letters for their counterparts in the space $Q_l$ (resp., $\mathcal{E}$, $\mathcal{R}$, $\mathcal{Z}$, etc). Then, the characterization property (17) implies:

$$z_l = \mathcal{K}_l^{-1}(\mathcal{Z}_l) = \sum_{i=1}^{\tilde{m}_l} \mathcal{Z}_{l,i} \mathcal{N}_{l,i} M_{l,i} w_{l,i}^{(1)}$$

where $\mathcal{Z}_{l,i}$ is the $i$-th component of $\mathcal{Z}_l \in Q_l$. The first two steps in Algorithm 1 are implemented simultaneously. Let $\mathcal{E}_l^0 = \mathcal{K}_l(\xi_l^0) \in Q_0$ be the counterpart of $\xi_l^0$. We have to consider two separate cases: $l = 1$ and $l > 1$. Let $l = 1$, i.e. $r \in V_0$. Then

$$\xi_1^0 = -\sum_{i=1}^{\tilde{m}_1} \mathcal{N}_{1,i} M_{1,i} \mathcal{P}_{1,i} M_{1,i} \mathcal{N}_{1,i}^T M^{-1} r = \sum_{i=1}^{\tilde{m}_l} \mathcal{N}_{1,i} M_{1,i} w_{1,i}^{(1)} \mathcal{E}_{1,i}^0$$

where

$$\mathcal{E}_{1,i}^0 = -[w_{1,i}^{(1)}]^T M_{1,i} \mathcal{N}_{1,i}^T M^{-1} r. \tag{19}$$

This step requires $O(\tilde{m}_1)$ arithmetical operations. Let $l > 1$, i.e. $r \in V_{l-1}$. The vector $r$ is numerically represented by a vector $\mathcal{R}_{l-1} \in Q_{l-1}$,

$$r = \sum_{i=1}^{\tilde{m}_{l-1}} \mathcal{N}_{l-1,i} M_{l-1,i} w_{l-1,i}^{(1)} \mathcal{R}_{l-1,i}.$$

Note that we can not use the straightforward approach as in (19) to compute the residual $\mathcal{E}_l^0$ because it is too expensive. Instead of that, we use the vector $\mathcal{R}_{l-1}$:

$$\begin{aligned} \mathcal{E}_{l,i}^0 &= -[w_{l,i}^{(1)}]^T M_{l,i} \mathcal{N}_{l,i}^T M^{-1} \sum_{j=1}^{\tilde{m}_{l-1}} \mathcal{N}_{l-1,j} M_{l-1,j} w_{l-1,j}^{(1)} \mathcal{R}_{l-1,j} \\ &= -[w_{l,i}^{(1)}]^T \sum_{k \in \mathfrak{S}_l(i)} \hat{\mathcal{N}}_{l-1,k} M_{l-1,k} \mathcal{N}_{l-1,k}^T M^{-1} \sum_{j=1}^{\tilde{m}_{l-1}} \mathcal{N}_{l-1,j} M_{l-1,j} w_{l-1,j}^{(1)} \mathcal{R}_{l-1,j}. \end{aligned}$$

Suppose that we have a recursive procedure for computing coefficients

$$\Upsilon_{kl}^{l-1} = [w_{l-1,k}^{(1)}]^T M_{l-1,k} \mathcal{N}_{l-1,k}^T M^{-1} \mathcal{N}_{l-1,j} M_{l-1,j} w_{l-1,j}^{(1)}. \tag{20}$$

Then

$$\mathcal{E}_{l,i}^0 = \sum_{k \in \mathfrak{S}_l(i)} \sum_{j \in \mathfrak{O}_{l-1}(k)} \sqrt{\frac{\sigma_{l,i}}{\sigma_{l-1,k}}} \Upsilon_{kj}^{l-1} \mathcal{R}_{l-1,j}$$

where $\mathfrak{O}_{l-1}(k)$ denotes a collection of indexes of the subdomains adjacent to subdomain $\Omega_{l-1,k}$ including the subdomain itself. The arithmetical complexity of computing $\mathcal{E}_l^0$ is $O(\tilde{m}_l)$. Efficient algorithms for recursive computing coefficients $\sigma_{l,i}$ and $\Upsilon_{kj}^l$ are discussed at the end of this section.

The third step in Algorithm 1 is implemented in the space $Q_l$. Since $y_l^0 \in V_l$ and $V_l$ is the image of matrix $C_l$, we have $y_l^i \in V_l$. From the last step in Algorithm 1, we derive that $MB_{l+1}\xi_l^{i-1} \in V_l$. Since $A_l = M - C_l$, all the residuals $\xi_l^{i-1}$, $i = 0, \ldots, \beta$, belong to $V_l$. Using the bijective mapping $\mathcal{K}_l$, we define

$$\mathcal{Z}_l^i = \mathcal{K}_l(C_l B_{l+1} \xi_l^i) \quad \text{and} \quad \mathcal{G}_l^i = \mathcal{K}_l(MB_{l+1}\xi_l^i).$$

Then, the third step in Algorithm 1 can be rewritten as follows:

$$\begin{array}{rcl} \mathcal{Y}_l^i & = & \mathcal{Y}_l^{i-1} - \gamma_i \mathcal{Z}_l^{i-1}, \\ \mathcal{E}_l^i & = & \mathcal{E}_l^{i-1} - \gamma_i (\mathcal{G}_l^{i-1} - \mathcal{Z}_l^{i-1}) \end{array} \tag{21}$$

where $\mathcal{Y}_l^i = \mathcal{K}(y_l^i)$. The arithmetical complexity of (21) is $O(\tilde{m}_l)$.

The last step in Algorithm 1 is implemented such that to provide vectors $\mathcal{Z}_l^{i-1}$ and $\mathcal{G}_l^{i-1}$ which are used in formula (21). We have to consider two separate cases: $l = 1$ and $l > 1$. Let $l = 1$, i.e. $r \in V_0$. Then

$$\tilde{r} = M^{-1} \left( \sum_{i=1}^{\tilde{m}_1} \mathcal{N}_{1,i} M_{1,i} w_{1,i}^{(1)} \mathcal{Y}_{1,i}^{\beta} + r \right).$$

The arithmetical complexity of this step is $O(\tilde{m}_1)$. Let $l > 1$, i.e. $r \in V_{l-1}$. Then, we have to compute vectors

$$\mathcal{Z}_{l-1} = \mathcal{K}_{l-1}(C_{l-1} M^{-1}(y_l^{\beta} + r)) \quad \text{and} \quad \mathcal{G}_{l-1} = \mathcal{K}_{l-1}(y_l^{\beta} + r).$$

Let $g_{l-1} = \mathcal{K}_{l-1}^{-1}(\mathcal{G}_{l-1})$ and $z_{l-1} = \mathcal{K}_{l-1}^{-1}(\mathcal{Z}_{l-1})$. Then

$$\begin{array}{rcl} g_{l-1} & = & \displaystyle\sum_{i=1}^{\tilde{m}_{l-1}} \mathcal{N}_{l-1,i} M_{l-1,i} w_{l-1,i}^{(1)} \mathcal{R}_{l-1,i} + \sum_{i=1}^{\tilde{m}_l} \mathcal{N}_{l,i} M_{l,i} w_{l,i}^{(1)} \mathcal{Y}_{l,i}^{\beta} \\ & = & \displaystyle\sum_{i=1}^{\tilde{m}_{l-1}} \mathcal{N}_{l-1,i} M_{l-1,i} w_{l-1,i}^{(1)} \mathcal{R}_{l-1,i} + \sum_{i=1}^{\tilde{m}_l} \mathcal{N}_{l,i} \left( \sum_{j \in \mathfrak{S}_l(i)} \hat{\mathcal{N}}_{l-1,j} M_{l-1,j} \hat{\mathcal{N}}_{l-1,j}^T \right) w_{l,i}^{(1)} \mathcal{Y}_{l,i}^{\beta} \\ & = & \displaystyle\sum_{i=1}^{\tilde{m}_{l-1}} \mathcal{N}_{l-1,i} M_{l-1,i} w_{l-1,i}^{(1)} \mathcal{R}_{l-1,i} + \sum_{i=1}^{\tilde{m}_{l-1}} \mathcal{N}_{l-1,i} M_{l-1,i} w_{l-1,i}^{(1)} \sqrt{\frac{\sigma_{l,j}}{\sigma_{l-1,i}}} \mathcal{Y}_{l,j}^{\beta}. \end{array}$$

The definition of the bijective mapping $\mathcal{K}_{l-1}$ gives

$$\mathcal{G}_{l-1,i} = \mathcal{R}_{l-1,i} + \sqrt{\frac{\sigma_{l,j}}{\sigma_{l-1,i}}}\mathcal{Y}^{\beta}_{l,j}.$$

The computation of $\mathcal{Z}_{l-1}$ is associated with a matrix-vector multiplication. Indeed,

$$\begin{aligned} z_{l-1} &= C_{l-1}M^{-1}g_{l-1} \\ &= \sum_{i=1}^{\tilde{m}_{l-1}} \mathcal{N}_{l-1,i}M_{l-1,i}\mathcal{P}_{l-1,i}M_{l-1,i}\mathcal{N}^T_{l-1,i}M^{-1}\sum_{j=1}^{\tilde{m}_{l-1}} \mathcal{N}_{l-1,j}M_{l-1,j}w^{(1)}_{l-1,j}\mathcal{G}_{l-1,j} \end{aligned}$$

and

$$\mathcal{Z}_{l-1,i} = \sum_{j\in\mathfrak{O}_{l-1}(i)} \Upsilon^{l-1}_{ij}\mathcal{G}_{l-1,j}.$$

The arithmetical complexity of computing $\mathcal{G}_{l-1}$ and $\mathcal{Z}_{l-1}$ is $O(\tilde{m}_{l-1})$.

Let us consider the recursive algorithms for computing coefficients $\Upsilon^l_{ij}$ and $\sigma_{l,i}$. We recall that $w^{(1)}_{l,i} = \sqrt{\sigma_{l,i}}\varepsilon_l$ where $\varepsilon_l = (1,\ldots,1)^T$ and $[w^{(1)}_{l,i}]^T M_{l,i}w^{(1)}_{l,i} = 1$. Then, formula (15) leads to the following recursion:

$$\sigma_{l,i} = \frac{1}{\varepsilon_l^T M_{l,i}\varepsilon_l} = \frac{1}{\varepsilon_l^T\left(\sum\limits_{j\in\mathfrak{S}_l(i)} \hat{\mathcal{N}}_{l-1,j}M_{l-1,j}\hat{\mathcal{N}}^T_{l-1,j}\right)\varepsilon_l} = \frac{1}{\sum\limits_{j\in\mathfrak{S}_l(i)}\dfrac{1}{\sigma_{l-1,j}}}. \tag{22}$$

The coefficients $\Upsilon^1_{ij}$ are computed with $O(\tilde{m}_1)$ arithmetical operations. Combining formulas (15) and (20) we get the following recursive procedure:

$$\begin{aligned} \Upsilon^l_{kj} &= [w^{(1)}_{l,k}]^T M_{l,k}\mathcal{N}^T_{l,k}M^{-1}\mathcal{N}_{l,j}M_{l,j}w^{(1)}_{l,j} \\ &= [w^{(1)}_{l,k}]^T\left[\sum_{r\in\mathfrak{S}_l(k)} \hat{\mathcal{N}}_{l-1,r}M_{l-1,r}\hat{\mathcal{N}}^T_{l-1,r}\right]\mathcal{N}^T_{l,k}M^{-1}\mathcal{N}_{l,j}\left[\sum_{t\in\mathfrak{S}_l(j)} \hat{\mathcal{N}}_{l-1,t}M_{l-1,t}\hat{\mathcal{N}}^T_{l-1,t}\right]w^{(1)}_{l,j} \\ &= \left[\sum_{r\in\mathfrak{S}_l(k)} \sqrt{\frac{\sigma_{l,k}}{\sigma_{l-1,r}}}[w^{(1)}_{l-1,r}]^T M_{l-1,r}\mathcal{N}^T_{l-1,r}\right]M^{-1}\left[\sum_{t\in\mathfrak{S}_l(j)} \sqrt{\frac{\sigma_{l,j}}{\sigma_{l-1,t}}}\mathcal{N}_{l-1,t}M_{l-1,t}w^{(1)}_{l-1,t}\right] \\ &= \sum_{r\in\mathfrak{S}_l(k)}\sum_{t\in\mathfrak{S}_l(j)} \sqrt{\frac{\sigma_{l,k}}{\sigma_{l-1,r}}\frac{\sigma_{l,j}}{\sigma_{l-1,t}}}\Upsilon^{l-1}_{rt}. \end{aligned} \tag{23}$$

The arithmetical complexity of algorithms (22) and (23) is $O(\sum_{l=1}^{L}\tilde{m}_l)$. Assuming that the numbers $\tilde{m}_1,\ldots,\tilde{m}_L$ form a decreasing geometric series, we get the upper bound $O(\tilde{m}_1)$ for the arithmetical complexity.

Finally, formula (11) implies that coefficient $\Upsilon^L_{ij}$ are all what is needed for deriving the coarse grid problem.