Computing 62, 205–228 (1999)

Computing

© Springer-Verlag 1999
Printed in Austria

Energy Optimization of Algebraic Multigrid Bases

J. Mandel, Denver, CO, **M. Brezina,** Boulder, CO,
and **P. Vaněk,** Los Angeles, CA

Received March 9, 1998; revised January 25, 1999

Abstract

We propose a fast iterative method to optimize coarse basis functions in algebraic multigrid by minimizing the sum of their energies, subject to the condition that linear combinations of the basis functions equal to given zero energy modes, and subject to restrictions on the supports of the coarse basis functions. For a particular selection of the supports, the first iteration gives exactly the same basis functions as our earlier method using smoothed aggregation. The convergence rate of the minimization algorithm is bounded independently of the mesh size under usual assumptions on finite elements. The construction is presented for scalar problems as well as for linear elasticity. Computational results on difficult industrial problems demonstrate that the use of energy minimal basis functions improves algebraic multigrid performance and yields a more robust multigrid algorithm than smoothed aggregation.

AMS Subject Classifications: 65N55, 65F10.

Key Words: Algebraic multigrid, constrained optimization.

1. Introduction

This paper is concerned with some aspects of the design of Algebraic Multigrid Methods (AMG) for the solution of symmetric, positive definite linear algebraic systems arising from finite element discretization of elliptic boundary value problems. Multigrid methods achieve their efficiency through the complementary effects of smoothing by relaxation and the coarse level correction, using a hierarchy of coarse meshes. Unlike classical, geometrical multigrid, where the fine meshes are obtained by successive refinement of a given coarse mesh, algebraic multigrid strives to build artificial coarse spaces and the associated operators from a matrix associated with a given fine mesh. For further general information, see, e.g., [3, 11, 17, 22, 24, 34].

First generation multigrid convergence theories [1, 2, 10] were based mainly on the concept of approximation properties between the grids and elliptic regularity. Early in multigrid history, it was recognized that a class of multigrid algorithms can be seen as a successive minimization of the energy in a set of directions consisting of coordinate vectors on the finest grid and of directions given by coarse grid coordinate vectors, transferred via interpolation onto the finest grid [19]. A similar idea has become the background for a new generation of multigrid theory based on energy estimates for successive subspace corrections [4]; see also [3, 35] for overview of related results and relations to domain decomposition methods. In this approach, the role of approximation properties is lessened and an important place is taken by an upper bound on the energy of the basis functions in the grids hierarchy.

The classical approach to the design of algebraic multigrid methods [5, 8, 23, 24] has been to attempt to have the intergrid transfer operators possess approximation properties similar to geometric multigrid. In [29, 30], we have proposed an alternative set of objectives, which includes

- *minimization of energy:* the basis functions on the coarse levels should have as small energy as possible;

- *preservation of null space:* the span of basis functions on each coarse level should contain zero energy modes, at least away from the boundary; and

- *limited overlap:* the supports of the basis functions on the coarse levels should overlap as little as possible, or, equivalently, the system matrices on the coarse levels should have as few nonzero entries as possible.

These objectives are then used to build coarse space basis functions, or, equivalently, the prolongation operators. The objectives were motivated by the multigrid theory of [4], recognition of the need to represent zero energy modes in the coarse space exactly [14, 16], and early work on algebraic multigrid with prolongation by smoothed aggregation [25, 26].

In [6, 29–31], we have proposed the prolongation by smoothed aggregation as an attempt to satisfy the above objectives approximately. The energy of the basis functions – and convergence of the resulting multigrid method – can be further improved by more smoothing of the coarse basis functions, but at the cost of increasing their supports, and thus the number of non zeros in coarse level matrices and the computational complexity of the multigrid algorithm [6, 27, 30, 31]. An efficient method to reduce the energy of the coarse basis functions without increasing their supports has been therefore of interest.

Another way to satisfy the above objectives is to set up a minimization problem and to define coarse basis functions as its solution. The issues are then the properties of such coarse basis functions, and how to compute them efficiently. Construction of basis functions by minimization of the quadratic function equal to the sum of their energies, subject to the constraint that the basis functions sum up to a constant, was proposed in [32] and further developed in [7, 33]. The constrained minimization problem was solved by Lagrange multipliers. In [33], it was proved that in one dimension, the resulting multigrid method has convergence rate independent of jumps in coefficients and mesh size, the performance of such method for oscillatory and discontinuous coeffients in two dimensions was investigated numerically, and it was demonstrated that, in some cases, the coarse basis functions obtained from energy minimization coincide with standard finite element basis functions.

In this paper, we present a general approach to building coarse basis functions by the minimization of the sum of their energies, subject to the condition that their linear combinations equal to given zero energy modes, and subject to restrictions on their supports. We develop a fast projected gradient descent algorithm for the solution of the minimization problem, and prove that its convergence rate is bounded independently of the mesh size under usual assumptions on finite elements. In the case when the initial approximation to the basis functions is obtained by aggregation, the first iteration of the projected gradient descent algorithm coincides with our earlier smoothed aggregation method [29]. The construction and the algorithm are presented here for scalar as well as vector problems, such as linearized elasticity. In the scalar case, our abstract minimization problem essentially reduces to that of [32]. Our algorithm for its solution as well its application to basis functions derived from aggregation are different.

Our computational results show that in many situations, the first iteration of the minimization algorithm, that is, smoothed aggregation, results in a quite acceptable algebraic multigrid method already, and additional iterations have only marginal effect. But additional iterations of the energy minimization algorithm make the multigrid method more robust and result in an improvement in multigrid performance for difficult problems, especially in the case of vector problems, such as elasticity. To approximate minimization of the maximum of the energies of the coarse basis functions, we have tried to minimize also the sum of powers of order s of the energies, $s > 1$. This, however, did not yield an additional benefit.

In Section 2, we recall the matrix notation used here. Algebraic multigrid with coarse bases by smoothed aggregation is summarized in Section 3. For ease of exposition, the new method is first presented in Section 5 for the scalar case and with some simplifications. The minimization algorithm in the general case, including vector problems and preconditioning, is given in Section 4. The choice of the initial approximation and application to algebraic multigrid in specific situations is discussed in Section 6. Finally, Section 7 contains computational results and Section 8 is the conclusion.

2. Notation

The symbol $\mathbb{M}^{m\times n}$ will denote the space of all $m \times n$ real matrices. The canonical unit vectors in $\Re^n$ are denoted by e_i and $\mathbf{1}$ is the vector of all ones. For a matrix $M = (m_{ij})$, denote by m_{*k} and m_{k*} the k-th column and row, respectively. If the matrix M has a block structure, M_{*k} and M_{k*} are the k-th block column and row, respectively. For a square matrix $M \in \mathbb{M}^{n\times n}$, $\mathrm{tr}(M)$ is its trace and

$\operatorname{diag}(M) = \operatorname{diag}(m_{ii})$ is the diagonal matrix with the same diagonal elements as M. For a collection of matrices $\{D_k\}$, $\operatorname{diag}(D_k)$ is the block diagonal matrix with D_k as diagonal blocks.

For two matrices $X, Y \in \mathbb{M}^{m\times n}$, define the term-by-term product by $X * Y = (z_{ij})$, $z_{ij} = x_{ij} y_{ij}$. If M is $m \times n$ block matrix and N is a scalar $m \times n$ matrix, we use the notation $N * M = (n_{ij} M_{ij})$. We will use the Frobenius inner product on $\mathbb{M}^{m\times n}$, defined by

$$(X, Y) = \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij} y_{ij} = \operatorname{tr}(X^T Y).$$

We will make use of the fact that this inner product of matrices reduces to the usual Euclidean inner product for vectors, and it is the sum of the inner products of the respective rows or columns,

$$(X, Y) = \sum_{i=1}^{m} (x_{i*}, y_{i*}) = \sum_{j=1}^{n} (x_{*j}, y_{*,j}). \tag{1}$$

The same identity holds for block matrices $X = (X_{ij})$, $Y = (Y_{ij})$, obtained by replacing scalar entries of the matrices by matrix blocks. The associated Frobenius matrix norm is

$$\|X\| = (X, X)^{1/2} = \left(\sum_{i=1}^{m} \sum_{j=1}^{m} x_{ij}^2 \right)^{1/2}. \tag{2}$$

For a symmetric and positive definite matrix $D \in \mathbb{M}^{m\times m}$, define another matrix inner product on $\mathbb{M}^{m\times n}$ by $(X, Y)_D = (DX, Y)$, and note that

$$(X, Y)_D = (DX, Y) = \sum_{j=1}^{n} (Dx_{*j}, y_{*j}) = \sum_{j=1}^{n} (x_{*j}, Dy_{*j}) = (X, DY). \tag{3}$$

The spectral radius of M is $\rho(M)$, and M^+ denotes the Moore-Penrose pseudoinverse of M.

We will use matrix-to-matrix mappings, with the notation like $\tilde{Z} : P \mapsto Q$ or $\tilde{Z}(P) = Q$. We will always enclose the argument of a matrix-to-matrix mappings in parentheses because application of a matrix-of-matrix map and multiplication of matrices do not associate, $\tilde{Z}(AB) \neq \tilde{Z}(A)B$. Composition of matrix-to-matrix mappings will be denoted by $\circ$, as in $(\tilde{W} \circ \tilde{Z})(P) = \tilde{W}(\tilde{Z}(P))$. The operator norm of a linear matrix-to-matrix mapping, induced by the Frobenius norm of matrices (2), will be denoted also by $\|\cdot\|$,

$$\|\tilde{Z}\| = \max_{A \in \mathbb{M}^{m\times n}, A \neq 0} \frac{\|\tilde{Z}(A)\|}{\|A\|}. \tag{4}$$

The spectral radius of a linear matrix-to-matrix map $\tilde{A}$ will be denoted also by $\varrho(\tilde{A})$.

We believe that the matrix-to-matrix mapping notation makes the paper easier to read than if the tensor notation were used.

3. Algebraic Multigrid by Smoothed Aggregation

We briefly recall the concept of smoothed aggregation as we have previously applied it to solving a system of algebraic equations

$$Au = b, \tag{5}$$

where $A \in \mathbb{M}^{n\times n}$ is a symmetric positive definite matrix arising from a finite element discretization of a second order elliptic problem. For simplicity, consider the model variational problem

$$v \in V : \qquad a(v, w) = \int_\Omega f w \, dx + \int_{\partial\Gamma_N} g w \, ds, \qquad \forall w \in V, \tag{6}$$

where

$$\begin{aligned} V &= \{v \in H^1(\Omega) \mid v = 0 \text{ on } \partial\Omega \setminus \Gamma_N\}, \\ a(v, w) &= \int_\Omega \alpha(x) \operatorname{grad} v \operatorname{grad} w \, dx, \end{aligned}$$

and $\alpha(x) > 0$ in Ω. The problem (6) is discretized by P1 or Q1 conforming Lagrange elements. The value of the quadratic form $a(v, v)$ is called the *energy* of $v \in H^1(\Omega)$. Let $\Re^n$ be the space of the values of the degrees of freedom of the discretization, and $\Pi_n : \Re^n \to H^1(\Omega)$ the finite element interpolation operator. The images $\Pi_n e_i$ of the natural basis vectors of $\Re^n$ are called *shape functions* or *basis functions*.

On input, our method requires the system of algebraic equations (5), discrete representations of zero energy functions, and the system $\{\mathcal{A}_i\}_{i=1}^m$ of node aggregates forming a disjoint covering of the index set,

$$\bigcup_{i=1}^m \mathcal{A}_i = \{1, \ldots, n\}, \quad \mathcal{A}_i \cap \mathcal{A}_j = \emptyset \text{ for } i \neq j. \tag{7}$$

Let the set of functions we want to be represented exactly by the coarse space functions be denoted by $\{b^i\}_{i=1}^r$. These are typically all zero energy modes of the principal part of the differential operator, without any boundary conditions applied. For the model problem, $r = 1$ and b^1 is the constant function. Because we have assumed Lagrange elements in the model problem, the discrete representation of b^1 with respect to the finite element basis is the vector of all ones, $b^1 = \Pi_n \mathbf{1}$.

The *tentative prolongator* is now defined by

$$\hat{P}_{ij} = \begin{cases} 1 & \text{if } i \in \mathcal{A}_j \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

See Section 6 of this paper and [28] for the construction in the general case. In order to eliminate high energy components from the range of $\hat{P}$, we introduce a *prolongator smoother* $q(D^{-1}A)$, where $D \in \mathbb{M}^{n \times n}$ is symmetric positive definite matrix acting as a preconditioner for A, and q is a polynomial such that $q(0) = 1$. We then define the final prolongator P by

$$P = q(D^{-1}A)\hat{P}. \tag{9}$$

The final prolongator is used in a multigrid method in the usual fashion. A full presentation of the method requires the introduction of a large amount of notation, which is not relevant to the purpose of this paper. Therefore, we present only the basic method and describe its extensions informally. See [29] for details.

Let E be another preconditioner for A, for example, $E = \operatorname{diag}(A)$.

Algorithm 1 (Basic two-level multigrid). Given A, P, and initial approximation u to the solution of $Au = b$, do:

1. *pre-smoothing:* $u \leftarrow u - E^{-1}(Au - b)$,
2. *coarse grid correction:* find v from the *coarse problem*

$$(P^T AP)v = P^T(Au - b) \tag{10}$$

   and correct the solution by $u \leftarrow u - Pv$,

3. *post-smoothing:* $u \leftarrow u - (E^T)^{-1}(Au - b)$.

Recall that the prolongator P acts from $\Re^m$ to $\Re^n$, $m < n$. The space $\Re^m$ is called the coarse space, $\Re^n$ is the fine space, and $P^T AP$ is called the coarse matrix.

In the actual computational method, the coarse problem is solved iteratively by one recursive application of Algorithm 1, starting from the initial approximation $v = 0$, resulting in so-called V-cycle [2]. To apply Algorithm 1 recursively, we construct a decomposition of the index set $\{1, \ldots, m\}$ into aggregates similarly as in (7), and build the tentative and final prolongator from a yet coarser space similarly as in (8) and (9). Eventually, when the coarse problem is small enough, it is solved by a direct method. To increase robustness of the method, this recursive algorithm is used as a preconditioner for the method of Conjugate Gradients for the solution of (5).

The coarse space $\Re^m$ is embedded in the fine space $\Re^n$ via the prolongator P, and we define an interpolation operator $\Pi_m : \Re^m \to H^1(\Omega)$ by composition,

$$\Pi_m u = \Pi_n(Pu). \tag{11}$$

The functions $\Pi_m e_j$, $j = 1, \ldots, m$, are *coarse basis functions*. Naturally, the vectors Pe_j are called *discrete coarse basis functions*.

The quadratic functional

$$u^T Au = \|u\|_A^2 = a(\Pi_n u, \Pi_n u), \qquad u \in \Re^n,$$

has the interpretation of energy of the vector u. The energy of a coarse space vector is defined via the embedding P,

$$\|Pv\|_A^2 = a(\Pi_n Pv, \Pi_n Pv) = a(\Pi_m v, \Pi_m v), \qquad v \in \Re^m. \tag{12}$$

The energy of a discrete coarse basis function Pe_j can be bounded by

$$\begin{aligned}(APe_j, Pe_j) &= (Aq(D^{-1}A)\hat{P}e_j, q(D^{-1}A)\hat{P}e_j)\\ &= (D^{-1/2}AD^{-1/2}q(D^{-1/2}AD^{-1/2})D^{1/2}\hat{P}e_j,\\ &\qquad q(D^{-1/2}AD^{-1/2})D^{1/2}\hat{P}e_j) \qquad (13)\\ &\le \varrho(D^{-1/2}AD^{-1/2}q^2(D^{-1/2}AD^{-1/2}))(D\hat{P}e_j, \hat{P}e_j)\\ &\le \max_{0\le t\le \varrho(D^{-1/2}AD^{-1/2})} tq^2(t)\,(D\hat{P}e_j, \hat{P}e_j), \qquad (14)\end{aligned}$$

using the Cauchy-Schwarz inequality and the spectral mapping theorem, cf. [28]. The polynomial q is then chosen to minimize the right hand side of (14), cf. [27], which is equivalent to minimizing an upper bound on the maximum of the energies of the coarse basis functions.

In this paper, we advocate minimization of the quadratic functional equal to the sum of the energies of the coarse basis functions. This functional is easier to minimize, and in the practically important case $\deg q = 1$, the first step of the projected descent method for its minimization will be seen to give exactly the same result as (9). We will also investigate the minimization of $\sum_{j=1}^m (APe_j, Pe_j)^s$, $s > 1$, to approximate the minimization of $\max\{(APe_j, Pe_j) \mid j = 1, \ldots, m\}$.

The prolongation operator based on the technique of smoothed aggregation yields very good convergence properties of the multigrid method, but it leads to increased computational complexity for small aggregates or a high degree of the polynomial q. In the case of coarsening by a factor of about 3 in each direction [30], one is forced to use a polynomial q of degree 1 in order to assure sparsity of the coarse level matrices. Minimizing energy of the coarse space basis functions without the penalty of increased computational complexity is the subject of this paper.

4. Coarse Bases by Minimization of Energy: Simple Case

In this section, we consider scalar problems such as (6). Although Section 5 also covers this case, this will allow us to explain more clearly the main idea unobscured by technical details.

Let $N = (n_{ij}) \in \mathbb{M}^{n\times m}$ be a given 0-1 matrix corresponding to the allowed nonzero structure of the sought prolongator P, that is, $p_{ij} = 0$ whenever $n_{ij} = 0$. Define the space

$$\mathcal{N} = \{P \in \mathbb{M}^{n\times m} : p_{ij} = 0 \text{ if } n_{ij} = 0\},$$

equipped with the Frobenius matrix inner product (P, Q). Note that

$$\mathcal{N} = \{P \in \mathbb{M}^{n\times m} : N * P = P\},$$

the mapping $P \mapsto N * P$ is the orthogonal projection in $\mathbb{M}^{n\times m}$ onto $\mathcal{N}$, and, for all $S, Q \in \mathbb{M}^{n\times m}$, it holds that

$$(S, N * Q) = (N * S, Q). \tag{15}$$

We will construct our prolongator to satisfy the general requirements postulated in [30], cf. the summary in Section 1.

Our first requirement is that the coarse space basis reproduces zero modes of the bilinear form. In the scalar case, the modes are constants, which means that $\sum_{j=1}^{m} p_{ij} = 1$ for all i such that the node i and its neighbors do not have a Dirichlet constraint. The selection is implemented by a $0-1$ vector $f = (f_i)$,

$$\left.\begin{array}{l} f_i = 0 \text{ if node } i \text{ belongs to element with Dirichlet constraint,} \\ f_i = 1 \text{ otherwise.} \end{array}\right\} \tag{16}$$

The energy of the j-th coarse-space basis function equals $(APe_j, Pe_j) = p_{*j}^T Ap_{*j}$, cf. (12). We will minimize the sum of the energies,

$$J(P) = \frac{1}{2}\sum_{j=1}^{m} p_{*j}^T Ap_{*j} = \frac{1}{2}(P, AP) \to \min \tag{17}$$

subject to

$$\left.\begin{aligned} &P \in \mathcal{N}, \\ &\sum_{j=1}^{m} p_{ij} = 1 \text{ if } f_i = 1, \quad i = 1, \dots, n, \end{aligned}\right\} \tag{18}$$

cf. [32]. Since A is symmetric positive definite, we have for some $\alpha > 0$ that $v^T A v \geq \alpha v^T v$, for all v. By setting $v = p_{*j}$ and by summation over j, we get

$$(P, AP) = \sum_{j=1}^{m} p_{*j}^T A p_{*j} \geq \alpha \sum_{j=1}^{m} p_{*j}^T p_{*j} = \alpha(P, P).$$

Consequently, the quadratic form (P, AP) is strictly convex. It follows that the problem (17)–(18) has a unique solution, which we denote by P^*.

Define the subspace $\mathcal{Z} \subset \mathcal{N}$ by

$$\begin{aligned} \mathcal{Z} &= \{Q_1 - Q_2 \mid Q_1, Q_2 \text{ satisfy (18)}\} \\ &= \{Q \in \mathcal{N} \mid \sum_{j=1}^{m} q_{ij} = 0 \quad \text{if } f_i = 1, \text{ for all } i = 1, \dots, n\}. \end{aligned}$$

To write an iterative method for the solution of the problem (17)–(18), we will need the gradient of $J(P)$ in the space $\mathcal{N}$ and the orthogonal projection onto $\mathcal{Z}$.

Lemma 1. *The gradient of the functional J in the space $\mathcal{N}$ is*

$$\operatorname{grad}_{\mathcal{N}} J(P) = N * (AP).$$

Proof: The gradient is defined by

$$J(P + tQ) = J(P) + t(Q, \operatorname{grad}_{\mathcal{N}} J(P)) + o(t), \quad t \to 0, \quad \forall Q \in \mathcal{N}.$$

Since

$$\frac{1}{2}(P + tQ, A(P + tQ)) = \frac{1}{2}(P, AP) + t\ (Q, AP) + o(t), \quad t \to 0,$$

it holds that

$$(Q, \operatorname{grad}_{\mathcal{N}} J(P)) = (Q, AP) = (Q, N * (AP)),$$

for all $Q \in \mathcal{N}$, by (15). □

Lemma 2. *The orthogonal projection in $\mathcal{N}$ onto $\mathcal{Z}$ is $\tilde{Z} : S \mapsto Q$, where*

$$q_{ij} = s_{ij} - n_{ij} f_i \frac{(s_{i*}, n_{i*})}{(n_{i*}, n_{i*})}. \tag{19}$$

Proof: Let $S \in \mathcal{N}$ and $Q = \tilde{Z}(S)$ be given by (19). To prove that Z is the orthogonal projection onto $\mathcal{Z}$, we need to show that $Q \in \mathcal{Z}$ and $S - Q$ is orthogonal to $\mathcal{Z}$.

It follows from (19) and the fact that $S \in \mathcal{N}$ that $q_{ij} = 0$ if $n_{ij} = 0$, hence, $Q \in \mathcal{N}$. Now fix $i = 1, \dots, n$ such that $f_i = 1$. Then

$$\begin{aligned} \sum_{j=1}^{m} q_{ij} &= \sum_{j=1}^{m} \left(s_{ij} - n_{ij} \frac{(s_{i*}, n_{i*})}{(n_{i*}, n_{i*})} \right) \\ &= \left(\sum_{j=1}^{m} s_{ij} \right) - (s_{i*}, n_{i*}) = 0. \end{aligned}$$

Hence, $Q \in \mathcal{Z}$.

It remains to show that $(S - Q, Z) = 0$ for all $Z \in \mathcal{Z}$. Let $Z = (z_{ij}) \in \mathcal{Z}$. Using the facts that $z_{ij} n_{ij} = z_{ij}$ and $\sum_{j=1}^{m} z_{ij} = 0$ if $f_i = 1$, we have

$$\begin{aligned} (S - Q, Z) &= \sum_{j=1}^{m} \sum_{i=1}^{n} z_{ij} n_{ij} f_i \frac{(s_{i*}, n_{i*})}{(n_{i*}, n_{i*})} \\ &= \sum_{i=1}^{n} \left(f_i \sum_{j=1}^{m} z_{ij} \right) \frac{(s_{i*}, n_{i*})}{(n_{i*}, n_{i*})} = 0. \quad \square \end{aligned}$$

Given P_0 satisfying (18), the problem (17)–(18) is now equivalent to finding P such that

$$P - P_0 \in \mathcal{Z} \quad \text{and} \quad \tilde{Z}(\text{grad}_{\mathcal{N}} J(P)) = \tilde{Z}(N * (AP)) = 0. \tag{20}$$

We compute the prolongation operators by the projected steepest descent method.

Algorithm 2 (Energy minimization, simple case). Given A, N, f, and P_0 satisfying (18),

1. choose $\omega \in (0, 2/\varrho(A))$,

2. $P_{k+1} = P_k - \omega \tilde{Z}(N * (AP_k))$, $k = 0, 1, 2, \dots$, where $\tilde{Z}$ is defined in Lemma 2.

We now derive a bound on the convergence of Algorithm 2.

Theorem 1. *Let A_j denote the square matrix obtained by selecting rows and columns of A with indices $\{i \mid n_{ij} = 1\}$. Further let $c_1 = \min_j \lambda_{\min}(A_j)$, $c_2 = \max_j \lambda_{\max}(A_j)$. Then Algorithm 2 converges for all $\omega \in (0, 2/c_2)$, and*

$$\|P_{n+1} - P^*\| \leq \max\{|1 - \omega c_1|, |1 - \omega c_2|\}\|P_n - P^*\|.$$

Proof: From the definition of c_1, c_2, we have $c_1 p_{*j}^T p_{*j} \leq p_{*j}^T A p_{*j} \leq c_2 p_{*j}^T p_{*j}$, for all $P \in \mathcal{N}$ and all $j = 1, \dots, m$. Hence, by summation over j, we obtain

$$c_1(P, P) \leq (P, AP) \leq c_2(P, P), \quad \forall P \in \mathcal{N}. \tag{21}$$

Let P_n satisfy (18). Then $P_n - P^* \in \mathcal{Z}$. Using this, the facts that $\tilde{Z}$ is projection onto $\mathcal{Z}$, and that P^* solves (20), hence $\tilde{Z}(N * (AP^*)) = 0$, we have

$$\begin{aligned} P_{n+1} - P^* &= P_n - P^* - \omega\tilde{Z}(N * (AP_n)) \\ &= \tilde{Z}(P_n - P^*) - \omega\tilde{Z}(N * (A(P_n - P^*))) \\ &= (\tilde{Z} \circ (I - \omega\tilde{A}) \circ \tilde{Z})(P_n - P^*), \end{aligned}$$

where $\tilde{A} : \mathcal{N} \to \mathcal{N}, \quad \tilde{A}(P) = \operatorname{grad}_{\mathcal{N}} J(P) = N * (AP)$, cf. (4). Since $\tilde{Z}$ is an orthogonal projection, we have $\|\tilde{Z}\| = 1$, and it follows that

$$\|P_{n+1} - P^*\| \leq \|I - \omega\tilde{A}\|\|P_n - P^*\|.$$

Because from (1) and (9),

$$\begin{aligned} (\tilde{A}(P), Q) &= (N * (AP), Q) = (AP, N * Q) = (A(P), Q) \\ &= (N * (AP), Q)(AP, N * Q) = (AP, Q) = (P, \tilde{A}(Q)), \end{aligned}$$

it follows that the operator $\tilde{A}$ is symmetric on $\mathcal{N}$, and, using (21), $c_1 \leq \lambda_{\min}(\tilde{A}) \leq \lambda_{\max}(\tilde{A}) \leq c_2$. Hence,

$$\|I - \omega\tilde{A}\| = \rho(I - \omega\tilde{A}) \leq \max\{|1 - \omega c_1|, |1 - \omega c_2|\}. \qquad \square$$

Remark 1. (Convergence independent of mesh size). Under the assumption that the fine mesh is locally quasi-uniform, and the coefficients of the differential operator are uniformly elliptic and bounded, the convergence rate of Algorithm 2 does not deteriorate with decreasing mesh size. Indeed, each A_j is a (small) submatrix of A corresponding to a Dirichlet boundary value problem on a mesh with nodes i such that $n_{ij} = 1$, thus we have $\lambda_{\max}(A)/\lambda_{\min}(A_j) \leq C$. Obviously, $c_2 \leq \lambda_{\max}(A)$. Choosing $\omega = \frac{1}{\bar{\lambda}}$, where $\bar{\lambda} \geq \lambda_{\max}(A)$ is an easy to compute estimate, we obtain a fast and practical algorithm.

Remark 2. (Relation to smoothed aggregation). Let $P_0 = \hat{P}$ be obtained from aggregation, cf. (8), and choose N so that nonzeros are allowed only at the nodes of the aggregates and at their immediate neighbors. In terms of the underlying finite element mesh, we allow $p_{ij} \neq 0$ only on nodes forming the aggregate plus one strip of nodes around. Then the prolongator P_1 resulting from one iteration of Algorithm 2 is identical to the smoothed aggregation prolongator from [30], described in Section 3 of this paper. Indeed, with our selection of N, we have $P_0 \in \mathcal{N}$ and $AP_0 \in \mathcal{N}$. Recalling that in (16), f was chosen so that $f_i = 1$ except at the Dirichlet boundary and at its immediate neighbors, we have

$$\sum_{j=1}^{m} a_{ij} = 0 \text{ if } f_i = 1.$$

Since $\sum_{j=1}^{m}(P_0)_{ij} = 1$ if $f_i = 1$, we have $(AP_0\mathbf{1})_i = (A\mathbf{1})_i = 0$ if $f_i = 1$, hence $AP_0 \in \mathcal{Z}$. Consequently, $ZAP_0 = AP_0$, and $P_1 = P_0 - \omega AP_0 = (I - \omega A)P_0$, the original smoothed aggregation prolongator for the smoother of degree 1.

Remark 3 (Construction of initial P_0). If only the nozero pattern N of P is given, one can still use the initial $P_0 = \hat{P}$ from (8), using a system of aggregates $\{\mathcal{A}_j\}$ such that (7) holds and $\mathcal{A}_j \subset \{i \mid n_{ij} = 1\}$ for all j. Such system $\{\mathcal{A}_j\}$ always exists if N does not have a zero row, which is reasonable to assume. A better choice in this case may be to define P_0 as a prolongator given by one of the existing methods [8, 24]. The prolongator is then improved by application of Algorithm 2.

5. Coarse Bases by Minimization of Energy: General Case

We will now discuss an extension of the method described in Section 4, suitable in the case of systems of equations such as three-dimensional linear elasticity.

We will find it convenient to work in terms of blocks corresponding to degrees of freedom associated with a single node. Let n denote the number of nodes on the fine level, and d be the number of degrees of freedom per node. The matrix A will be understood as a block matrix with $n \times n$ blocks of size $d \times d$. Let B be a given $(nd) \times r$ matrix whose columns are discrete representation of the zero energy modes of the problem. That is, the columns of B span the null space of the stiffness matrix for the related problem with no essential boundary conditions imposed. For example, in the case of 3D elasticity, columns of B are formed by discrete representation of the 6 rigid body modes with respect to the current fine level basis. The number r will play the role of the number of degrees of freedom per node on the coarse level.

Note that we may well have $r \neq d$. For example, in a $3D$ finite element model with displacement degrees of freedom, we have $d = 3$ because there are 3 degrees of freedom per node, but $r = 6$ because the dimension of the space of rigid body modes is 6.

The discrete representation of rigid body modes is the basis of many other efficient algorithms [9, 13–16, 18, 20, 21], and it is commonly available in finite element packages.

We will view the prolongator P as a block matrix $P = (P_{ij})$ with $n \times m$ blocks of size $d \times r$. The matrix B will be considered a block column formed by blocks of size $d \times r$,

$$B = \begin{pmatrix} B_1 \\ B_2 \\ \vdots \\ B_n \end{pmatrix}.$$

The matrix B is supplied as data on the finest level.

Let

$$R = \begin{pmatrix} R_1 \\ R_2 \\ \vdots \\ R_m \end{pmatrix}$$

be a given block column matrix with blocks of size $r \times r$. We will require that $PR = B$ with a modification in rows that correspond to points near the boundary as follows. Assume that for each node i, there is given an $r \times r$ matrix F_i. The purpose of the matrix F_i is to select the subspace of zero energy modes that are to be captured by the range of P at the node i: we will require that the prolongation P satisfies $(P_{i*}R - B_i)F_i = 0$, for all $i = 1, \ldots, n$.

It follows that the colums of R specify the discrete representation of the zero energy modes on the coarse level, i.e., $(P^T AP)R = 0$ away from boundary nodes with constraints.

The energy minimization algorithm presented in this section assumes the matrices R and F_i as input. We describe their construction separately in Section 6 below.

Let $N \in \mathbb{M}^{n\times m}$ be a given $0 - 1$ matrix with 1 where nonzero blocks of P are allowed, and define

$$\mathcal{N} = \{P \in \mathbb{M}^{nd\times mr} : P_{ij} = 0 \text{ if } n_{ij} = 0, \quad i = 1, \ldots, n, \quad j = 1, \ldots, m\}.$$

Next, assume there is given a block diagonal symmetric positive definite matrix $D = \operatorname{diag}(D_{ii})$ with $d \times d$ diagonal blocks D_i. The matrix D will play the role of a preconditioner for A in the energy minimization process. Note that D preserves the nonzero pattern of P, i.e., $D : \mathcal{N} \to \mathcal{N}$.

Let $s \geq 1$ and consider the problem

$$J(P) = \frac{1}{s} \sum_{k=1}^{nd} (p_{*k}^T A p_{*k})^s = \frac{1}{s} \operatorname{tr}(\operatorname{diag}(P^T A P))^s \to \min \tag{22}$$

subject to

$$\left.\begin{aligned} P &\in \mathcal{N} \\ (P_{i*}R - B_i)F_i = 0 \quad &\text{for all } i. \end{aligned}\right\} \tag{23}$$

The simple case, discussed in Section 5, is obtained by setting $s = 1$, $d = 1$, $r = 1$, $B = b_{*1} = \mathbf{1}$, and $F_i = f_i = 0$ or 1.

Again, denote

$$\begin{aligned} \mathcal{Z} &= \{Q_1 - Q_2 \mid Q_1, Q_2 \text{ satisfy (23) }\} \\ &= \{Q \in \mathcal{N} \mid Q_{i*}RF_i = 0 \text{ for all } i\}, \end{aligned} \tag{24}$$

and compute the gradient of $J(P)$ in $\mathcal{N}$ and the orthogonal projection onto $\mathcal{Z}$, now in the matrix inner product $(\cdot, \cdot)_D$.

Lemma 3. *The gradient of J in the space $\mathcal{N}$ with the inner product $(\cdot, \cdot)_D$ is*

$$\operatorname{grad}_D J(P) = D^{-1}(N * (AP((\operatorname{diag}(P^T AP))^{s-1}). \tag{25}$$

Proof: By a direct computation,

$$\begin{aligned} (Q, \operatorname{grad}_D J(P))_D &= \frac{d}{dt} J(P + tQ)|_{t=0} \\ &= \frac{1}{2s} \sum_{k=1}^{nd} s(p_{*k}^T p_{*k})^{s-1} 2q_{*k}^T A p_{*k} \\ &= (Q, AP((\operatorname{diag}(P^T AP))^{s-1}) \\ &= (Q, D^{-1}(N * (AP((\operatorname{diag}(P^T AP))^{s-1}))_D, \end{aligned}$$

for all $Q \in \mathcal{N}$. □

Lemma 4. *The $(\cdot, \cdot)_D$–orthogonal projection in $\mathcal{N}$ onto the subspace $\mathcal{Z}$ is $\tilde{Z} : S \mapsto Q$, defined by*

$$Q_{ij} = S_{ij} - V_{ij}, \tag{26}$$

where

$$V_{ij} = \left(\sum_{k=1}^{m} S_{ik} U_{ki}\right)\left(\sum_{k=1}^{m} U_{ki}^T U_{ki}\right)^+ U_{ji}^T, \tag{27}$$

$$U_{ji} = n_{ij} R_j F_i. \tag{28}$$

Proof: Write (24) and (27) as

$$\mathcal{Z} = \{Q \in \mathcal{N} \mid Q_{i*} U_{*i} = 0, \text{ for all } i = 1, \dots, n\}, \tag{29}$$

$$V_{i*} = S_{i*} U_{*i} (U_{*i}^T U_{*i})^+ U_{*i}^T, \tag{30}$$

respectively. Let $S \in \mathcal{N}$ and Q be given by (26). We need to verify that $Q \in \mathcal{Z}$ and $S - Q = V$ is orthogonal to $\mathcal{Z}$.

It follows from (27) and (28) that $Q_{ij} = 0$ whenever $n_{ij} = 0$, hence $Q \in \mathcal{N}$. To show that $Q \in \mathcal{Z}$, note that the mapping $u \mapsto U_{*i}(U_{*i}^T U_{*i})^+ U_{*i}^T u$ is a projection onto the span of the columns of U_{*i}; hence, using (30),

$$Q_{i*} U_{*i} = S_{i*}(I - U_{*i}(U_{*i}^T U_{*i})^+ U_{*i}^T) U_{*i} = 0,$$

which proves that $Q \in \mathcal{Z}$.

Let $Z \in \mathcal{Z}$. We need to show that $(V, Z)_D = 0$. Let r be a row of S_{i*} and

$$v = r U_{*i} (U_{*i}^T U_{*i})^+ U_{*i}^T$$

the corresponding row of V_{i*}, cf. (30). Since v is the orthogonal projection of r onto the span of the rows of U_{*i}^T, it holds that $(v, y) = 0$ for all row vectors y orthogonal to the span of the rows of U_{*i}^T, that is, for all row vectors y such that $yU_{*i} = 0$. By summation for r running over all rows of S_{i*} and using (1), we obtain that

$$(V_{i*}, Y_{i*}) = 0 \qquad \text{for all } Y_{i*} \text{ such that } Y_{i*} U_{*i} = 0.$$

But $Z_{i*} U_{*i} = 0$ implies that $D_i Z_{i*} U_{*i} = 0$, hence, with $Y_{i*} = D_i Z_{i*}$ and using (3), we obtain

$$0 = (V_{i*}, D_i Z_{i*}) = (D_i V_{i*}, Z_{i*}).$$

By summation, we finally get

$$\sum_{i=1}^{n} (D_i V_{i*}, Z_{i*}) = (V, Z)_D = 0. \quad \square$$

Remark 4. Lemma 4 implies that the projection $\tilde{Z}$ does not depend on the preconditioning matrix D as long as D is symmetric positive definite and has the assumed block diagonal structure.

We are now ready to write the general algorithm for computation of the improved prolongator.

Algorithm 3 (Energy minimization, general case). Given $s \geq 1$, A, D, N, B, R, F_i, and P_0 satisfying (23), for $k = 0, 1, 2, \ldots$, choose ω_k and set

$$P_{k+1} = P_k - \omega_k \tilde{Z}(D^{-1}(N * AP_k(\operatorname{diag}(P_k^T AP_k)^{s-1}))),$$

where $\tilde{Z}$ is defined in Lemma 4.

In Algorithm 3, the values ω_k are to be chosen sufficiently small so that $J(P_{k+1}) < J(P_k)$ except at the solution. In the practically important case $s = 1$, we have the following direct generalization of Algorithm 2.

Algorithm 4 (Energy minimization, quadratic case). Given A, D, N, B, R, F_i, and P_0 satisfying (23),

1. choose $\omega \in (0, 2/\varrho(D^{-1}A))$,
2. $P_{k+1} = P_k - \omega \tilde{Z} D^{-1}(N * AP_k)$, $k = 0, 1, 2, \ldots$, where $\tilde{Z}$ is defined in Lemma 4.

We then have a straightforward generalization of Theorem 1.

Theorem 2. *Let $\|\cdot\|_D = (\cdot,\cdot)_D$ and A_j^D be the square matrix obtained by selecting the rows and columns of $D^{-1/2}AD^{-1/2}$ with indices $\{i : n_{ij} = 1\}$. Further, let $c_1 = \min_j \lambda_{\min}(A_j^D)$, $c_2 = \max_j \lambda_{\max}(A_j^D)$. Then, for any initial guess P_0 satisfying the constraint* (23) *and $\omega \in (0, 2/c_2)$, Algorithm 4 creates a sequence $P_1, P_2 \ldots$ such that*

$$\|P_{k+1} - P^*\|_D \leq \max\{|1 - \omega c_1|, |1 - \omega c_2|\} \|P_k - P^*\|_D.$$

Proof: From the definition of c_1, c_2, we have for any $P \in \mathcal{N}$ that

$$c_1 \|P_{*j}\|^2 \leq (P_{*j}, D^{-1/2}AD^{-1/2}P_{*j}) \leq c_2 \|P_{*j}\|^2.$$

The rest of the proof follows by replacing in the proof of Theorem 1 $(\cdot,\cdot)$, $\|\cdot\|$, A, by $(\cdot,\cdot)_D$, $\|\cdot\|_D$, $D^{-1}A$, respectively, cf. (13). □

Similarly as in the simple case, this implies that the convergence of Algorithm 4 is independent of mesh size in the case of quasi uniform mesh and a uniformly elliptic problem.

6. Specification of Algorithm Components and Implementation

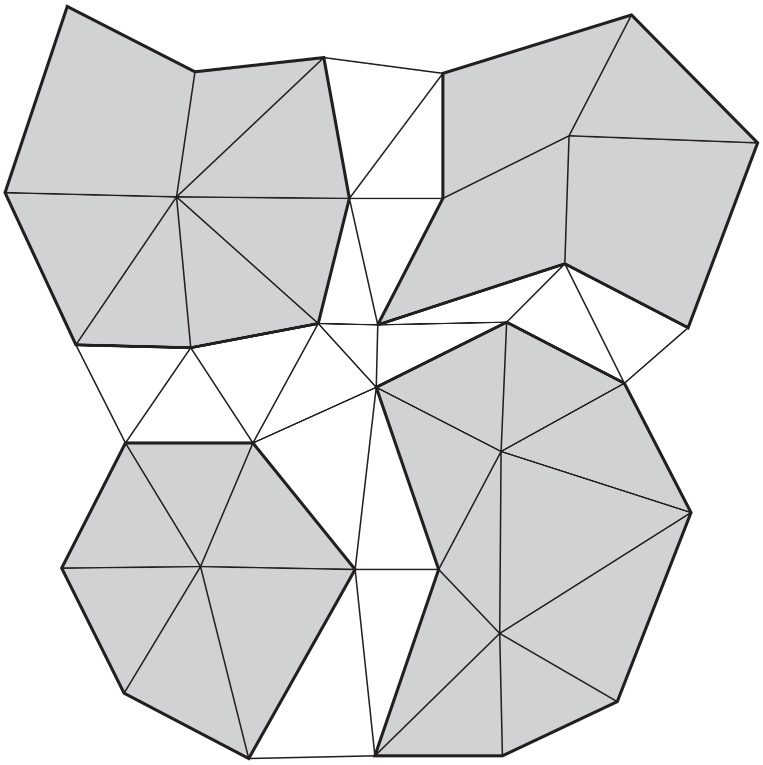

Figure 1. Example of node aggregation in 2D

To use the method of this paper, one needs to specify the choice of the nonzero structure N of P, as well as the matrices F_i, R, B, and P_0. The matrices N and B are given by the user, while F_i, R, and P_0 are computed as described below.

First let us specify the blocks F_i in the constraint (23). In our experiments, we have used the following simple construction. The notation $(AB)_i$ means the i-th block row of the block column vector AB. The size of $(AB)_i$ is $d \times r$. The blocks F_i are of size $r \times r$.

Algorithm 5 (Construction of F_i). Given A and B, for each node $i = 1, \ldots, n$, set $F_i = I$ if $(AB)_i = 0$ and $F_i = 0$ otherwise.

The use of this simple construction was made possible by the fact that our test

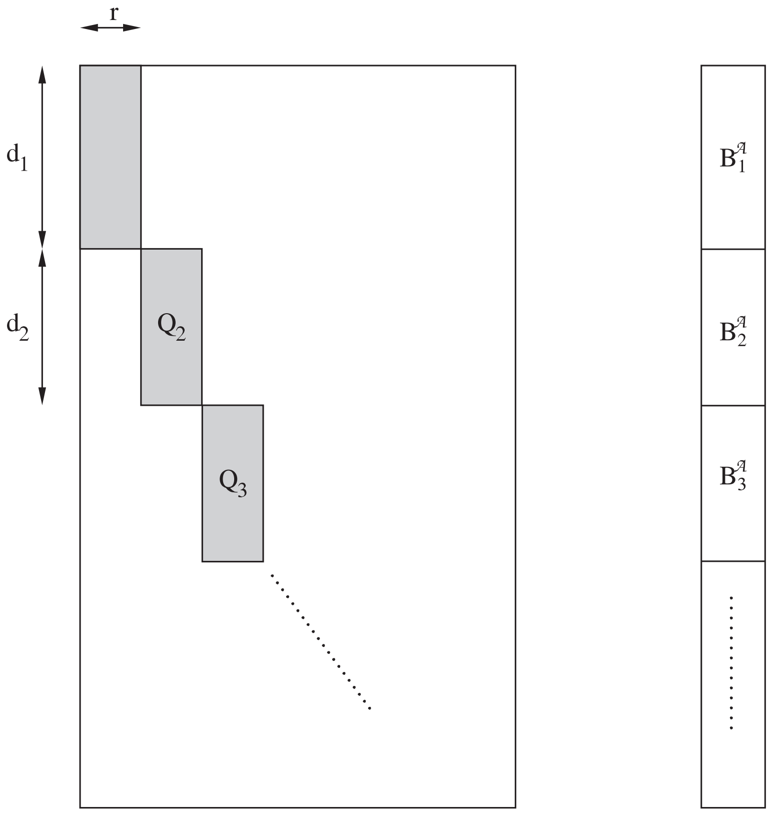

Figure 2. The structure of P_0 and B corresponding to aggregate degrees of freedom

problems had on every node either none or all degrees of freedom constrained. For more general boundary conditions, one needs to choose F_i to be an $r \times r$ matrix such that

$$\mathrm{Range}(F_i) = \mathrm{Null}(AB)_i. \tag{31}$$

For nodes i away from Dirichlet boundary conditions, that is, such that $(AB)_i = 0$, one can then again choose $F_i = I$.

If (31) does not hold, then $\mathrm{Range}(F_i) \subset \mathrm{Null}(AB)_i$ results in loosing approximation propertes that rely on exact representation of the local kernel in the coarse space, cf. [12], while $\mathrm{Range}(F_i) \supset \mathrm{Null}(AB)_i$ may result in coarse basis functions with higher energy than necessary.

We now present a construction of P_0. Let $\{\mathcal{A}_j\}_{j=1}^m$ be the set of aggregates forming disjoint covering of the set of nodes $\{1, \dots, n\}$, such that $\mathcal{A}_j \subset \{i \mid n_{ij} = 1\}$. Cf. (7) and Remark 3.

Let d_j denote the number of degrees of freedom in aggregate j. Without loss of generality, assume that the fine level nodes are numbered by consecutive numbers within each aggregate. Recall that the columns of B are the discrete representation of the zero energy modes with respect to the fine grid basis, and r denotes the number of columns of B.

Algorithm 6 (Construction of P_0).

1. Partition B into sets of rows $B_k^{\mathcal{A}}$, $k = 1, \ldots, m$, each corresponding to the set of degrees of freedom on an aggregate $\mathcal{A}_k$, cf. Fig. 1.

2. Use the QR algorithm to decompose matrices $B_k^{\mathcal{A}} = Q_k^{\mathcal{A}} R_k^{\mathcal{A}}$, where $Q_k^{\mathcal{A}}$ is $d_k \times r$ orthogonal, and $R_k^{\mathcal{A}}$ is $r \times r$ upper triangular square matrix.

3. Set
$$R = \begin{pmatrix} R_1^{\mathcal{A}} \\ R_2^{\mathcal{A}} \\ \vdots \\ R_m^{\mathcal{A}} \end{pmatrix},$$
and $P_0 = \operatorname{diag}(Q_k^{\mathcal{A}})$, cf. Fig. 2.

Step 2 is an implementation of Gram-Schmidt orthogonalization of the coarse space basis of rigid body modes, which was previously suggested in [30].

Remark 5 (Relation to smoothed aggregation). In the case when the aggregates and the nonzero structure of P are same as in the method of smoothed aggregation, we again recover the prolongator from smoothed aggregation as P_1, similarly as in Remark 2.

Remark 6 (Semi-coarsening). The fill-in in the coarse matrix $P^T AP$, associated with the application of smoothed aggregation method, becomes critical when many of the aggregates are “strips” only one element wide. These aggregate types will be encountered when semi-coarsening is used to deal with anisotropic problems. To reduce fill, the supports of coarse basis functions should not extend beyond the aggregate in the short direction of the strips. In [30], we have proposed to modify the result of smoothed aggregation by a heuristic “filtering” process to achieve this. Using the present approach, we can simply specify the desired support of the basis functions, and the present method performs the “filtering” in a systematic way.

7. Computational Experiments

The purpose of this section is to demonstrate the performance of the proposed method and compare it to the smoothed aggregation technique of [30]. The sparsity structure N was set to be the nonzero structure of AP_0, so the two methods coincide if only one minimization step is used in the present method.

The system of aggregates was constructed as described in [30]. The coarsening process was stopped when the number of nodes on the coarsest level reached 1300. This has resulted in two to three levels on the problems reported here. The smoother in steps 1 and 3 of Algorithm 1 was block Gauss-Seidel, with the blocks same as the aggregates. The order of the blocks was determined by a simple coloring algorithm so that the Gauss-Seidel steps within one color are independent and can be performed in parallel.

Table 1. The body of an automobile

unstructured shell, QUAD4 elements, 43,104 dofs.						
min.	setup	iter.	total	num. of	cond.	time/
steps	time	time	time	iters.	estim.	time sm. agg.
1	1.56	18.05	19.61	37	119.5	1.00
2	1.93	15.56	17.49	32	92.27	0.89
3	2.11	14.23	16.34	29	81.70	0.83
4	2.21	13.83	16.04	28	78.78	0.81
5	2.33	14.28	16.61	28	78.15	0.84
6	2.45	12.82	16.27	28	77.89	0.82
7	2.74	13.83	16.57	28	77.77	0.84

Table 2. Shell model of an automobile wheel

unstructured shell, QUAD4 elements, 59,490 dofs.						
min.	setup	iter.	total	num. of	cond.	time/
steps	time	time	time	iters.	estim.	time sm. agg.
1	1.30	5.00	6.30	9	4.55	1.00
2	1.47	4.53	6.00	8	3.81	0.95
3	2.00	3.98	5.98	7	2.91	0.94
4	2.08	3.56	5.64	6	2.22	0.89
5	2.28	3.57	5.85	6	2.00	0.92
6	2.47	3.46	5.93	6	1.91	0.94

Table 3. Large unstructured solid

large unstructured solid, 407,277 dofs.						
min.	setup	iter.	total	num. of	cond.	time/
steps	time	time	time	iters.	estim.	time sm. agg.
1	12.79	36.37	49.16	13	6.16	1.00
2	15.43	34.05	49.48	12	5.04	1.01
3	18.11	32.06	50.17	11	4.45	1.02
4	20.28	31.19	51.47	11	4.23	1.04
5	22.44	28.43	50.87	10	4.00	1.03
6	24.95	28.56	53.51	10	3.90	1.09

In all the experiments below, the method is Algorithm 1 with modifications for multiple levels and used as a preconditioner for the conjugate gradient method, as described in Section 3. The stopping criterion in preconditioned conjugate gradients was

Table 4. An automobile steering knuckle

unstructured solid, TETRA elements, 75,174 dofs.

min. steps	setup time	iter. time	total time	num. of iters.	cond. estim.	time/ time sm. agg.
1	8.13	5.36	13.49	8	2.58	1.00
2	8.60	5.37	13.97	8	2.55	1.03
3	8.85	5.40	14.25	8	2.50	1.06
4	8.86	4.79	13.79	7	2.20	1.02
5	9.07	4.78	13.85	7	2.21	1.03
6	9.52	4.78	14.30	7	2.17	1.06

$$\left(\frac{(Mr^i, r^i)}{(Mr^0, r^0)}\operatorname{cond}(M, A)\right)^{1/2} \leq 10^{-5},$$

where M is the preconditioner, r^i denotes the residual after i steps of the iteration, and cond(M, A) is a condition number estimate computed at run time from the Lanczos coefficients obtained in Conjugate Gradients.

The experiments were run on 15 R10000 processors of a 16-processor SGI Origin/2000. The results of the experiments are presented in Tables 1–4. The first line of each table corresponds to the smoothed aggregation technique with a smoother of degree 1. The parameters were $\omega = 0.53$ and $s = 1$. The CPU times are seconds of the master thread.

Sensitivity of the multigrid rate of convergence to the choice of ω in Algorithm 4 is investigated in Table 5. As expected, the minimization problem diverges for large ω.

Finally, we have tested the performance of AMG for minimization with $s > 1$ to approximate the minimization of the maximal energy of basis functions. Our algorithm used an (admittedly crude) choice of ω found by replacing ω by $\omega/2$ until the minimization algorithm decreased the value of the objective function. We have found little if any improvement of AMG convergence with increasing s. A sample result is in Table 6. The test problem was the same as for Table 5.

8. Conclusion

We have formulated and tested an algorithm for building algebraic multigrid bases from minimization of their energies. For minimization of the sum of the energies, the first step of the minimization gives the same result as our previous method, prolongation by smoothed aggregation. Numerical experiments indicate that, in general, adding more minimization steps improves the convergence properties of the solver, but the benefit is partially offset by the expense of the additional minimization steps. In the case of more complicated problems, such as shells, performing more minimization steps results in better computational times. For less complicat-

Table 5. Dependence of the AMG convergence (cond/num. iters.) on ω in Algorithm 4. Unstructured solid discretized using PENTA elements. 12125 nodes, 36375 degrees of freedom. Stopping condition: 10^{-6}

min. steps	2	3	4	5	8	16
$\omega = 0.20$	31.88/31	31.13/30	30.48/30	29.89/29	28.36/28	25.14/27
0.30	31.08/30	30.13/30	29.30/29	28.55/29	26.57/28	22.73/26
0.40	30.37/30	29.24/29	28.25/28	27.35/28	25.02/27	20.96/25
0.45	30.04/30	28.85/29	27.76/28	26.67/28	24.33/27	20.23/25
0.50	29.73/29	28.50/28	27.31/28	26.12/28	23.72/28	19.53/25
0.53	29.57/29	28.33/28	27.07/28	26.03/27	23.33/27	19.25/26
0.55	29.47/29	28.25/28	26.93/28	25.71/28	23.21/27	diverg*
0.60	29.25/29	28.15/29	26.67/28	26.0/29	diverg*	diverg*

* Divergence of the minimization process

Table 6. Convergence of the method for different values of s. Unstructured solid discretized using PENTA elements. 12125 nodes, 36375 degrees of freedom. Stopping condition: 10^{-6}

s	min. steps	AMG iters	cond. est.
1	4	28	27.07
1	8	27	23.33
2	4	31	27.17
2	8	30	26.04
3	4	29	26.13
3	8	29	26.06
4	4	29	26.17
4	8	29	26.12
5	4	29	26.18
5	8	29	26.15
10	4	29	26.18
10	8	29	26.18
10	20	29	26.15
10	40	29	26.14
10	80	28	26.02
smoothed agregs.		32	29.11

ed problems, the computational time grows, but only insignificantly, because the minimization process is relatively cheap.

Minimization of the sums of the energies to a power of s, $s > 1$, is more complicated and computationally expensive. It did not improve the multigrid convergence enough to offset the added cost, and in fact may make it worse in some cases. Our explanation of this fact is that although the maximum of the energies of basis functions appears in theoretical estimates, these estimates are far from being sharp. Therefore, while the principle of minimization of the energies of the basis functions is a useful tool for constructing the coarse basis functions, the actual behavior of AMG is not predicted exactly by the maximum of the energies.

The major memory requirements of the multigrid iteration are standard: the storage of the system matrix A, the prolongation operator P, the rigid body modes B, the corresponding data structures on the coarse levels, and the LU decomposition for the direct solution on coarsest level. The design of the prolongator by energy minimization requires another copy of P. Because minimization of the energy of the basic functions by projected conjugate gradients would require additional memory equivalent to three more copies of P, we did not pursue that possibility. The need to store an additional copy of P could be avoided by employing a projected Gauss-Seidel method, which will be explored elsewhere. We did not report memory usage in this paper, because our prototype code was not optimized for minimal memory use.

Acknowledgements

This research was supported in part by NSF grant ECS-9725504 and the Czech academic grant VS 97156.

References

[1] Bnak, R. E., Dupont, T.: An optimal order process for solving elliptic finite element equations. Math. Comp. *36*, 35–51 (1981).

[2] Braess, D., Hackbusch, W.: A new convergence proof for the multigrid method including the V cycle. SIAM J. Numer. Anal. *20*, 967–975 (1983).

[3] Bramble, J. H.: Multigrid methods. Pitman Research Notes in Mathematical Sciences, Vol. 294. Essex: Longman 1993.

[4] Bramble, J. H., Pasciak, J. E., Wang, J., Xu, J.: Convergence estimates for multigrid algorithms without regularity assumptions. Math. Comp. *57*, 23–45 (1991).

[5] Brandt, A.: Algebraic multigrid theory: the symmetric case. Appl. Math. Comput. *19*, 23–56 (1986).

[6] Brezina, M., Vaněk, P.: One black-box iterative solver. UCD/CCM Report 106, Center for Computational Mathematics, University of Colorado at Denver, 1997. http://www-math.cudenver.edu/ccmreports/rep106.ps.gz. Submitted to Computing.

[7] Chan, T. F., Smith, B., Wan, W. L.: An energy-minimizing interpolation for multigrid methods. Presentation at the 10th International Conference on Domain Decomposition, Boulder, CO, August 1997.

[8] Chan, T. F., Smith, B. F.: Domain decomposition and multigrid algorithms for elliptic problems on unstructured meshes. In: Domain Decomposition Methods in Scientific and Engineering Computing: Proceedings of the Seventh International Conference on Domain Decomposition, vol. 180 of Contemporary Mathematics, Providence, Rhode Island, 1994, American Mathematical Society, pp. 175–189.

[9] Farhat, C., Roux, F.-X.: A method of finite element tearing and interconnecting and its parallel solution algorithm. Int. J. Numer. Meth. Eng. *32*, 1205–1227 (1991).

[10] Hackbusch, W.: On the multigrid method applied to difference equations. Computing *20*, 291–306 (1978).

[11] Hackbusch, W.: Multigrid methods and applications. Computational Mathematics, Vol. 4. Berlin: Springer-Verlag, 1985.

[12] Janka, A.: Smoothed aggregation overlapping domain decomposition method for structural mechanics problems. Tech. Rep., University of West Bohemia. Submitted to Numer. Math.

[13] Le Tallec, P., Mandel, J., Vidrascu, M.: A Neumann-Neumann domain decomposition algorithm for solving plate and shell problems. SIAM J. Numer. Anal. *35*, 836–867 (1998).

[14] Mandel, J.: Iterative solvers by substructuring for the p-version finite element method. Comput. Meth. Appl. Mech. Eng. *80*, 117–128 (1990).

[15] Mandel, J.: Two-level domain decomposition preconditioning for the p-version finite element method in three dimensions. Int. J. Numer. Methods Eng. *29*, 1095–1108 (1990).

[16] Mandel, J.: Balancing domain decomposition. Comm. Numer. Meth. Eng. *9*, 233–241 (1993).

[17] Mandel, J., McCormick, S. F., Bank, R. E.: Variational multigrid theory. In: Multigrid methods, (McCormick, S. F. ed.), pp. 131–177. Frontiers in Applied Mathematics, Vol. 3. Philadephia: SIAM Books 1987.

[18] Mandel, J., Tezaur, R., Farhat, C.: A scalable substructuring method by Lagrange multipliers for plate bending problems. SIAM J. Numer. Anal. (in press).

[19] McCormick, S. F., Ruge, J.: Unigrid for multigrid simulation. Math. Comp. *19*, 924–929 (1983).

[20] Park, K. C., Justino, M. R., Jr., Felippa, C. A.: An algebraically partitioned FETI method for parallel structural analysis: algorithm description. Int. J. Numer. Meth. Eng. *40*, 2717–2737 (1997).

[21] Pavarino, L. F., Windlund, O. B.: Iterative substructuring methods for spectral element discretizations of elliptic systems. I. Compressible linear elasticity. SIAM J. Numer. Anal. (in press).

[22] Rüde, U.: Mathematical and computational techniques for multilevel adaptive methods. Frontiers in Applied Mathematics, Vol. 3. Philadelphia: SIAM 1993.

[23] Ruge, J. W., Stüben, K.: Efficient solution of finite difference and finite element equations by algebraic multigrid (AMG). In: Multigrid methods for integral and differential equations (Paddon, D. J., Holstein, H., eds.), pp. 169–212. Oxford: Clarendon Press 1985.

[24] Ruge, J. W., Stüben, K.: Algebraic multigrid (AMG). In: Multigrid methods (McCormick, S. F., ed.), pp. 73–130. Frontiers in Applied Mathematics, Vol. 3. Philadelphia: SIAM 1987.

[25] Vaněk, P.: Acceleration of convergence of a two level algorithm by smoothing transfer operators. Appl. Math. *37*, 265–274 (1992).

[26] Vaněk, P.: Fast multigrid solver. Appl. Math. *40*, 1–20 (1995).

[27] Vaněk, P., Brezina, M., Mandel, J.: Algebraic multigrid for problems with jumps in coefficients. In preparation.

[28] Vaněk, P., Brezina, M., Mandel, J.: Convergence analysis of algebraic multigrid based on smoothed aggregation. UCD/CCM Report 126, Center for Computational Mathematics, University of Colorado at Denver, February 1998. Numer. Math. (in press).
http://www-math.cudenver.edu/ccmreports/rep126.ps.gz.

[29] Vaněk, P., Mandel, J., Brezina, M.: Algebraic multigrid on unstructured meshes. UCD/CCM Report 34, Center for Computational Mathematics, University of Colorado at Denver, December 1994.
http://www-math.cudenver.edu/ccmreports/rep34.ps.gz.

[30] Vaněk, P., Mandel, J., Brezina, M.: Algebraic multigrid based on smoothed aggregation for second and fourth order problems. Computing *56*, 179–196 (1996).

[31] Vaněk, P., Mandel, J., Brezina, M.: Two-level algebraic multigrid for the Helmholtz problem. Cont. Math. *218*, 349–356 (1998).

[32] Wan, W. L.: An energy-minimizing interpolation for multigrid methods. UCLA CAM Report 97-18, Department of Mathematics, UCLA, April 1997.

[33] Wan, W. L., Chan, T. F., Smith, B.: An energy-minimizing interpolation for robust multigrid methods. UCLA CAM Report 98-6, Department of Mathematics, UCLA, February 1998.

[34] Wesseling, P.: An introduction to multigrid methods. Chichester: Wiley 1992.

[35] Xu, J.: Iterative methods by space decomposition and subspace correction: A unifying approach. SIAM Rev. *34*, 581–613 (1992).

J. Mandel
Department of Mathematics
University of Colorado at Denver
Denver, CO 80217-3364, USA
e-mail: jmandel@math.cudenver.edu

M. Brezina
Department of Applied Mathematics
University of Colorado at Boulder
Boulder, CO 80309-0526, USA
e-mail: Marian.Brezina@colorado.edu

P. Vaněk
Department of Mathematics
University of California at Los Angeles
Los Angeles, CA 90095-1555, USA
e-mail: vanek@math.ucla.edu
(On leave from
University of West Bohemia,
Plzeň, Czech Republic)