ALGEBRAIC MULTIGRID BY SMOOTHED AGGREGATION FOR SECOND AND FOURTH ORDER ELLIPTIC PROBLEMS

PETR VANĚK , JAN MANDEL AND MARIAN BREZINA*

January 1995, revised August 1995

Abstract. An algebraic multigrid algorithm for symmetric, positive definite linear systems is developed based on the concept of prolongation by smoothed aggregation. Coarse levels are generated automatically. We present a set of requirements motivated heuristically by a convergence theory. The algorithm then attempts to satisfy the requirements. Input to the method are the coefficient matrix and zero energy modes, which are determined from nodal coordinates and knowledge of the differential equation. Efficiency of the resulting algorithm is demonstrated by computational results on real world problems from solid elasticity, plate bending, and shells.

Key words. Algebraic multigrid, unstructured meshes, automatic coarsening, biharmonic equation, elasticity, plates and shells

AMS(MOS) subject classifications. 65N55, 65F10

1. Introduction. Multigrid methods are very efficient iterative solvers for systems of algebraic equations arising from finite element and finite difference discretizations of elliptic boundary value problems. The main principle of multigrid methods is to complement the local exchange of information in point-wise iterative methods by a global one utilizing several related systems, called *coarse levels,* with a smaller number of variables. The coarse levels are often obtained as a hierarchy of discretizations with different characteristic meshsizes, but this requires that the discretization be controlled by the iterative method. To solve linear systems produced by existing finite element software, one needs to create an artificial hierarchy of coarse problems. The principal issue is then to obtain computational complexity and approximation properties similar to those for nested meshes, using only the matrix of the system and as little additional input as possible.

An algebraic multigrid method that uses the system matrix only was developed by Brandt, McCormick, Ruge and Stüben [13, 17, 5, 14, 15]. The prolongations were based on the matrix of the system by partial solution from given values at selected coarse points [1]. The coarse grid points were selected so that each point could be interpolated to via so-called strong connections.

Our approach is based on *smoothed aggregation* introduced recently by Vaněk [19, 18]. First the set of nodes is decomposed into small mutually disjoint subsets. A tentative interpolation is then defined on those subsets as piecewise constant (in the discrete sense) for second order problems, and piecewise linear for fourth order problems. For the second order problems, the tentative prolongator can be constructed based solely on the information contained in the matrix of the system. For problems of fourth or-

* Center for Computational Mathematics, University of Colorado at Denver, Denver, CO 80217-3364. The first author is visiting from University of West Bohemia, Americká 42, 306 14 Plzeň, Czech Republic

der, additional user input is required in the form of the nodal values of the basis of linear functions over the entire domain Ω. The prolongation operator is then obtained by smoothing the output of the tentative prolongation and coarse level operators are defined variationally. The proposed multigrid method based on such prolongations converges very fast for a wide range of problems, including those with strongly anisotropic and discontinuous coefficients, plates, shells and thin solids. In addition, it has a low computational complexity per iteration since the typical coarsening ratio is about three in each dimension.

The aim of this paper is practical. We introduce objectives for the coarsening, motivated by the abstract multigrid theory [3, 4] and our practical experience. Then we develop a coarsening algorithm that attempts to satisfy those objectives. Like previous AMG methods, we strive to use the knowledge of the origin of the linear system implicitly. The algorithm is of course applicable to general symmetric, positive definite linear systems, but it is designed and work best for systems arising from finite elements.

The new variant of the method presented in this paper is distinguished by the additional user input of nodal coordinates to generate linear functions, which made practical treatment of fourth order problems possible. Similarly, rigid body motions are generated to improve convergence for elasticity problems that are ill-conditioned due to boundary conditions. In an earlier paper [20], we have developed an analysis of a simplified version of the method without input of linear functions and for second order problems only. The analysis from [20] does not apply here, so we rely on heuristic arguments and practical experience instead. Numerical experiments reported here confirm that the new variant of the method is superior to the old one for the biharmonic equation and for elasticity, particularly for shells.

For another approach to multigrid on unstructured meshes, see [7] and references therein. For other multigrid approaches to the biharmonic equation, see [6, 12, 21, 11]. For a multigrid theory for the biharmonic equation with non-nested finite element spaces, see [2].

2. Basic Multigrid Algorithm. For reference, we state the basic multigrid algorithm for the solution of a system of linear algebraic equations $A_1 x = b$. First, a preprocessing stage creates full rank prolongation matrix P_l of size $n_l \times n_{l+1}$. The coarse level matrx is defined by $A_{l+1} = P_l^T A_l P_l$. The coarsening is repeated until the order of A_{l+1} is small enough to allow an efficient direct solution. The number of levels will be denoted by L. The iterations then proceed as follows.

ALGORITHM 2.1 (BASIC MULTIGRID). *To solve the system $A_l x^l = b^l$, do:*
Pre-smoothing: *do ν_1 times $x^l \leftarrow \mathcal{S}^l(x^l, b^l)$*
Coarse grid correction:

- *let $b^{l+1} \leftarrow P_l^T (b^l - A_l x^l)$*
- *If $l+1 = L$, solve $A_{l+1} x^{l+1} = b^{l+1}$ by a direct method, otherwise apply γ iterations of this algorithm on level $l+1$, starting with initial guess $x^{l+1} = 0$*
- *correct the solution on level l by $x^l \leftarrow x^l + P_l x^{l+1}$*

Post-smoothing: *do ν_2 times $x^l \leftarrow \mathcal{S}^l(x^l, b^l)$.*

We use $\nu_1 = \nu_2 = \gamma = 1$ with the pre-smoothing iteration consisting of SOR with the relaxation parameter $\omega = 1.0$ in the forward sweep and $\omega = 1.85$ in the backward sweep. Post-smoothing is symmetric, consisting of a backward sweep $\omega = 1.85$ followed by a forward sweep with $\omega = 1.0$. Although SOR with $\omega > 1$ alone is known not to be a good smother, we have found that this combination of relaxation parameters gives very good results, apparently because the sweeps with different ω reduce different components of the error.

There is a set of basis functions $\{\varphi_i^l\}_{i=1}^{n_l}$ associated with each level, $V_l = \operatorname{span}\{\varphi_i^l\}$. The basis functions on the finest level are given as finite element shape functions, while the coarse level basis functions are determined from the prolongations by

$$\begin{bmatrix} \varphi_1^{k+1} \\ \vdots \\ \varphi_{n_{k+1}}^{k+1} \end{bmatrix} = P_k^T \begin{bmatrix} \varphi_1^{k} \\ \vdots \\ \varphi_{n_k}^{k} \end{bmatrix} \qquad k = 1, \ldots, L-1. \tag{1}$$

3. Requirements on Prolongations for Elliptic Problems of Order $2K$. We shall attempt to build coarse spaces that meet the following objectives. Though the input to our algorithm is algebraic, we find in convenient to formulate the requirements in terms of the basis functions (1).

(AMG1) Slow coarsening. The support of each basis function on the level l should be the union of a small number of supports of basis functions on level $l-1$.

(AMG2) Coarse supports should follow strong couplings. Each two nodes in the support of a coarse basis function should be connected by a path of strong couplings. Two nodes i and j on level l are strongly connected if the $|a_{ij}^l|$ is large compared with $\sqrt{a_{ii}^l a_{jj}^l}$. (For extension to systems, see Section 5.2.)

(AMG3) Reasonable geometry of supports. If the problem is not anisotropic, then we require that each support $\operatorname{supp}\varphi_i^l$ can be mapped to a reference domain (or one of few reference domains) $\hat{\Omega}$ using a mapping J_i^l with Jacobian satisfying the uniform spectral equivalence with respect to l, i

$$\partial J_i^l \approx \delta^{1-l}/h(x) I, \tag{2}$$

where $h(x)$ is the local meshsize of the finite element mesh at the point x and δ is the characteristic coarsening ratio. In particular, coarsening is done by about a factor of δ in each direction.

(AMG4) Bounded intersection. The support of each basis function intersects only a bounded number of supports of other basis functions on the same level. The maximum number of intersections does not depend on the level.

(AMG5) Polynomial (kernel) preserving property. If the energy norm of the problem is equivalent to H^K- norm or seminorm with sufficiently close constants of equivalence, then each coarse space V_l should contain polynomials up to degree $K-1$ away from the essential boundary conditions. If the equivalence is violated (e.g., in the case of insufficient essential boundary conditions in elasticity), then each coarse space should contain the kernels of the unconstrained bilinear form of the problem to be solved, that is, the zero energy modes. For elasticity, these are rigid body motions.

(AMG6) Small energy of coarse basis functions. For each coarse basis function φ_i^l, its energy should be minimal, up to a constant factor, among the set of all functions with the same L^2 norm and support contained in the support of φ_i^l.

(AMG7) Uniform l^2 equivalence. Discrete l^2 norms on all spaces V_l should be up to diagonal scaling uniformly equivalent with the continuous L^2 norm:

$$||u||^2_{L^2(\Omega)} \approx \sum_{i=1}^{n_l} \alpha_i^2 ||\varphi_i^l||^2_{L^2(\Omega)} \text{ for every } u = \sum_{i=1}^{n_l} \alpha_i \varphi_i^l. \tag{3}$$

REMARK 3.1. *When solving a scalar H^1-elliptic problem discretized by finite element basis satisfying the decomposition*

$$\sum_{i=1}^{n_1} \varphi_i^1 = constant \tag{4}$$

on all elements which are not adjacent to any node constrained by essential boundary condition, the requirement (AMG5) means that

$$\sum_{j=1}^{n_{l+1}} P_{ij} = 1, \qquad l = 1, \ldots, L-1, \tag{5}$$

for all rows i that do not correspond to degrees of freedom adjacent to an essential boundary condition.

REMARK 3.2. *In [20], we have proved that (AMG7) follows from a stronger version of (AMG3), and from (AMG4), and (AMG6), for H^1 elliptic problems discretized by shape regular linear finite elements. We also shown that for such problem, prolongation by smoothed aggregation produces coarse basis functions with bounded energy according to (AMG6).*

4. Motivation of the Requirements. The requirements (AMG1) and (AMG2) attempt to assure semi-coarsening when solving anisotropic problems [8, 16]. Algebraically, anisotropy is reflected in the coefficients of the stiffness matrix in the sense that the neighboring nodes are strongly coupled in the direction of anisotropy. Therefore, the supports of the coarse shape functions will be elongated in this direction. Similarly, for an isotropic problem discretized by elements with large aspect ratios (which happens in engineering practice due to mesh refinement and causes a numerical anisotropy), these requirements result in coarse suports with low aspect ratios. In either case, coarse levels tend not to be numerically anisotropic any more.

The requirement (AMG4) guarantees sparsity of the resulting coarse level matrices and thus low computatonal complexity.

The requirements (AMG3), (AMG4), and (AMG6) attempt to assure the standard inverse inequality on our coarse spaces. Together with the L^2 equivalence (AMG7), this means that our artificially created hierarchy of coarse spaces will behave much like a hierarchy of nested finite element spaces, even if our coarse basis functions are some unknown linear combinations of the original (fine level) finite element basis functions rather than mapped shape functions from a reference element.

In the rest of this section, we outline the main ideas of an (admittedly incomplete) argument how our requirements may lead to bounds that imply almost optimal multi-level estimates using the theory of [3]. Note that we do not attempt to present a real proof, but rather seek a heuristic confirmation of the choice of our requirements.

Let $a(\cdot,\cdot)$ be the bilinear form corresponding to an elliptic problem of order $2K$, $||.||_a = a(.,.)^{1/2}$. Further, let $(\cdot,\cdot)_{d,l}$ denote the diagonally weighted discrete inner product on level l defined by $(x,y)_{d,l} = \sum_{i=1}^{n_l} \alpha_i\beta_i||\varphi_i^l||_a^2$, where $x = \sum_{i=1}^{n_l} \alpha_i\varphi_i^l$, $y = \sum_{i=1}^{n_l} \beta_i\varphi_i^l$ and the associated norm $||.||_{d,l} = (.,.)_{d,l}^{1/2}$.

The essential assumption of the multigrid convergence theory of [3] is the existence of linear operators $Q_l : V_1 \rightarrow V_l$, $l = 2,\ldots,L$, $Q_1 = I_{V_1}$, which satisfy the *approximation property*

$$(6) \qquad ||Q_{l-1}u - Q_lu||_{d,l-1} \le C||u||_a, \qquad \text{for all } u \in V_{l-1}, l = 2,\ldots,L$$

and are *stable in energy,*

$$(7) \qquad ||Q_lu||_a \le C||u||_a, \qquad \forall u \in V_1.$$

We will define the desired linear mappings Q_l locally. Denote by Z the space of functions with zero energy from (AMG5), and by $|_{\operatorname{supp}\varphi_j^l}$ the restrictions of functions and functional spaces on the support of φ_j^l.

Let $u \in V_1$. For a basis function φ_j^l, let v_j be the the L^2 orthogonal projection of $u|_{\operatorname{supp}\varphi_j^l}$ on $Z|_{\operatorname{supp}\varphi_j^l}$. (Note that by (AMG5), $Z|_{\operatorname{supp}\varphi_j^l} \subset V_l|_{\operatorname{supp}\varphi_j^l}$.) Expand v_j in terms of shape functions restricted on a subset, $v_j = \sum_i \alpha_{ij}\varphi_i^l|_{\operatorname{supp}\varphi_j^l}$. After doing this for all φ_j^l, define $Q_lu = \sum_j \alpha_{jj}\varphi_j^l$.

The essential property of the this construction is that $(I - Q_l)u|_{\operatorname{supp}\varphi_j^l} = 0$ if u coincides with a function from Z on a patch consisting of $\operatorname{supp}\varphi_j^l$ and all intersecting supports. The approximation property (6) then follows by mapping the patch onto one of a finite number of configurations according to (AMG3), using a standard scaling argument on the factorspace modulo Z and the equivalence of norms (AMG7). The energy stability (7) may be obtained also by scaling on a factorspace (cf., (AMG5)), and the use of (AMG6).

5. Coarsening for Scalar Second Order Problems.

5.1. Basic Algorithm. We now construct prolongations P_l based on the matrix A_l. Assume that the finest level base functions satisfy (4). First, we create a tentative piecewise constant prolongator satisfying all of the above properties except for the energy bound in (AMG6). This prolongator will then be smoothed to satisfy (AMG6), while preserving the other properties.

We start by specifying a disjoint decomposition of the set of nodes on level l. Every component of the decomposition on level l (so-called *aggregate*) gives rise to one degree of freedom on level $l+1$.

Motivated by the requirement (AMG2) above, for a given ε we define the *strongly-coupled neighborhood of node i* as

$$N_i^l(\varepsilon) = \{j : \ |a_{ij}| \geq \varepsilon\sqrt{a_{ii}a_{jj}}\} \tag{8}$$

ALGORITHM 5.1 (AGGREGATION). *Let the matrix A_l of order n_l and $\varepsilon \in [0,1)$ be given. Generate a disjoint covering $\{C_i^l\}_{i=1}^{n_{l+1}}$ of the set $\{1,\ldots,n_l\}$ as follows.*

Initialization *Set $R = \{i \in 1,\ldots,n_l : N_i^l(0) \neq \{i\}\}$ and $j = 0$ (isolated nodes are not aggregated).*

Step 1 - Startup aggregation.

Select disjoint strongly coupled neighborhoods as the initial approximation of covering:

- *For all nodes $i \in R$:*

  if $N_i^l(\varepsilon) \subset R$, set $j \leftarrow j+1$, $C_j^l \leftarrow N_i^l(\varepsilon)$, $R \leftarrow R \setminus C_j^l$.

(Note that this approximation does not necessarily cover entire R).

Step 2 - Enlarging the decomposition sets.

Add remaining $i \in R$ to one of the sets C_k^l to which node i is strongly connected if any such set exists:

- *Copy $\tilde{C}_k^l \leftarrow C_k^l$, $k = 1,\ldots,j$*
- *For all nodes $i \in R$:*

  if there exists k such that $N_i^l(\varepsilon) \cap \tilde{C}_k^l \neq \emptyset$, set $C_k^l \leftarrow C_k^l \cup \{i\}$, $R \leftarrow R \setminus \{i\}$.

  (if more than one such set exists, choose the one to which node i has strongest coupling)

Step 3 - Handling the remnants.

Make the remaining $i \in R$ into aggregates that consist of subsets of strongly coupled neighborhoods:

- *For all nodes $i \in R$:*

  set $j \leftarrow j+1$, $\quad C_j^l \leftarrow R \cap N_i^l(\varepsilon)$, $\quad R \leftarrow R \setminus C_j^l$.

Typically, Step 3 is not performed, because $R = \emptyset$ after Step 2.

Define the tentative prolongation $\tilde{P}_l$ by the aggregates C_l^i:

$$(\tilde{P}_l)_{ij} = \begin{cases} 1 \text{ if } i \in C_j^l \\ 0 \text{ otherwise} \end{cases} \tag{9}$$

The piecewise constant prolongation $\tilde{P}_l$ will now be improved by a smoothing to get the final prolongation matrix P_l. We choose a simple damped Jacobi smoother, yielding the prolongation matrix

$$P_l = (I - \omega D^{-1} A_l^F)\tilde{P}_l \tag{10}$$

where $A_l^F = (a_{ij}^F)$ is the *filtered matrix* given by

$$a_{ij}^F = \left\{ \begin{array}{cc} a_{ij} \text{ if } & j \in N_i^l(\varepsilon) \\ 0 & \text{otherwise} \end{array} \right\} \text{ if } i \neq j, \qquad a_{ii}^F = a_{ii} - \sum_{j=1, j\neq i}^{n_l} (a_{ij} - a_{ij}^F), \tag{11}$$

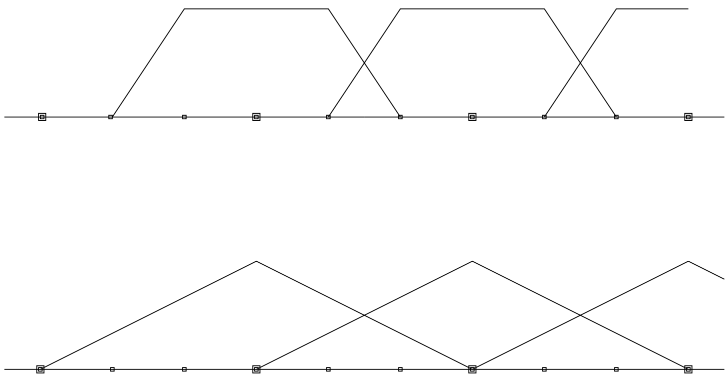

FIG. 1. *The basis functions given by aggregation and the corresponding smoothed basis for 1D Laplacian, using the smoother* $I - 2/3D^{-1}A$

and D denotes the diagonal of A_l.

When applying Algorithm 5.1 to uniformly elliptic problems, one usually obtains coarsening by about a factor of 3 in each dimension and the resulting coarse level matrix A_{l+1} tends to follow the nonzero pattern of the 9-point stencil. The filtration (11) has little or no effect in this case.

In the case of anisotropic problems, however, the application of the smoother with the unfiltered matrix would make the supports of basis functions overlap extensively in the direction of weak connections. Here the filtration prevents the undesired overlaps of the coarse space basis functions. By construction, A_l^F typically makes the nonzero pattern of A_{l+1} follow the 9-point stencil as in the uniformly V-elliptic case. The special treatment of the diagonal entries in (11) assures that the sum of the entries in a row of A_l^F is zero whenever the sum of the entries in the correspondig row of A is zero. Thus, a constant remains the local kernel of A_l^F at every point where constant is the local kernel of A_l. Consequently, for problems without zero-order term the final prolongator P_l satisfies the decomposition of unity (5) away from the essential boundary conditions which, together with (1), implies the same property for the coarse level basis functions.

Fig. 1 shows the 1D coarse basis functions resulting from smoothed aggregation. Note that for the 1D Laplace operator and the choice of $\omega = 2/3$ in (10), the smoothed coarse space basis is exactly that of P1 finite elements. Fig. 2 shows the typical aggregates obtained on an unstructured grid. The corresponding supports are formed by adding one belt of elements to the aggregates. The smoothing adds at most one more belt of adjacent elements.

Based on numerical experiments, we choose

$$\varepsilon = 0.08\,(\frac{1}{2})^{l-1}, \qquad \omega = \frac{2}{3}.$$

5.2. Block Approach for Systems. When there is more than one degree of freedom per node, the coarsening algorithm as described in Section 5.1 is likely to produce aggregates of physically incompatible degrees of freedom causing deterioration of convergence. This happens, e.g., for elements with derivatives as degrees of freedom,

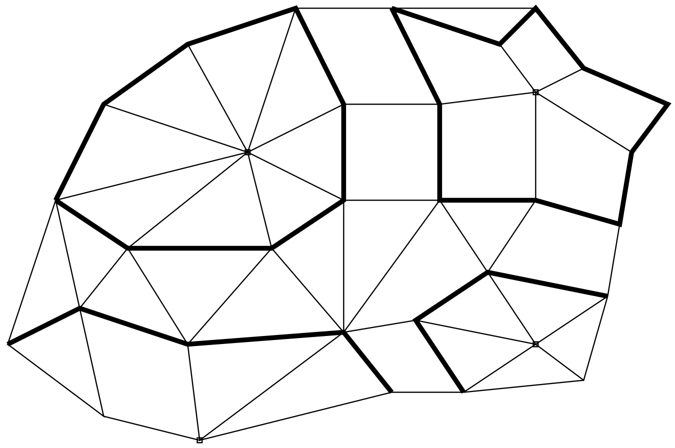

FIG. 2. *Typical 2D aggregates*

and for systems such as elasticity. The obvious remedy is to treat all degrees of freedom associated with a node as a block and the scalar operations on degree of freedom by their block counterparts for nodes. Let N_d denotes the number of degrees of freedom per node (assumed to be constant, for programming convencience) and $df(i)$ be the list of degrees of freedom associated with the node i. The communication between the neighboring nodes i, j can now be expressed in the form of a matrix selection $A_{(ij)}$ of order N_d,

$$A_{(ij)} = A(df(i), df(j)). \tag{12}$$

The definition of strongly coupled neighborhood of node i (8) is now replaced by

$$N_i^l(\epsilon) = \{\, j : \ \varrho(A_{(ii)}^{-1/2} A_{(ij)} A_{(jj)}^{-1/2}) \geq \epsilon\}. \tag{13}$$

Note that the value $\varrho(A_{(ii)}^{-1/2} A_{(ij)} A_{(jj)}^{-1/2})$ is the cosine of the spaces spanned by the basis functions associated with nodes i, j. Also, in the definition of auxiliary prolongations (9), we replace the numbers 1 and 0 by identity and zero matrices of order N_d, respectively. This is used in experiments No. 1 and 5–11 in Section 7.

5.3. Absolute Term. For a second order problem with positive absolute term $a(u,v) = \int_\Omega \sum_{i,j} a_{ij}\partial_i u \partial_j v + quv$, $[a_{ij}]$ positive definite, $q > 0$, the prolongation smoothers lose their constant-preserving property because the constant is no longer locally in the kernel of A_l. Fortunately, the presence of the absolute term improves the condition number of A_l, thus compensating for the loss of the preservation of a constant. Cf., experiments 4a - 4c.

6. Coarsening for High Order Problems and Elasticity.

6.1. Basic Algorithm for High Order Problems. The construction of prolongators resulting in the coarse spaces satisfying (AMG5) for $K > 1$ is not possible without additional user input. In order to be able to approximate the polynomials with

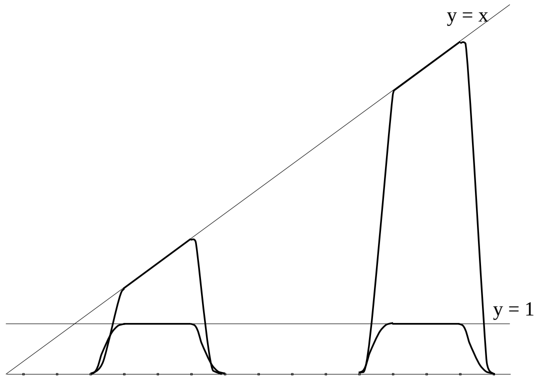

FIG. 3. *The coarse-space basis given by the restriction of p^0 and p^{11} onto aggregates of nodes*

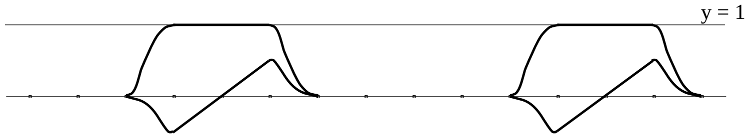

FIG. 4. *The coarse-space basis after l^2 Gram-Smidt modification*

degrees of up to $K-1$ by coarse space functions exactly, we need their representation with respect to the finest level basis $\{\varphi_i^1\}_{i=1}^{n_1}$.

For the elliptic problem of order $2K$ on the domain $\Omega \subset \mathbb{R}^d$, we need vectors $p^{(0)}, p^{(i,j)} \in \mathbb{R}^{n_1}$, $i = 1\ldots, K-1$, $j = 1, \ldots d$ satisfying

$$\sum_{k=1}^{n_1} p_k^{(0)} \varphi_k^1(\mathrm{x}) = 1, \quad \sum_{k=1}^{n_1} p_k^{(i,j)} \varphi_k^1(\mathrm{x}) = x_j^i, \quad \text{for every } \ \mathrm{x} = [x_1, \ldots, x_d] \in \Omega. \tag{14}$$

For example, to solve the biharmonic equation in 2D, we need $p^{(0)}$, $p^{(1,1)}$, and $p^{(1,2)}$, the representations with respect to the fine level basis of the planes $z=1$, $z=x$, $z=y$, respectively.

The proposed coarsening technique is a natural generalization of the concept of smoothed aggregation described in Section 5.1. The aggregation step (9) can be understood as restriction of the unit vector to aggregates C_i^l, which, in the scalar case, gave rise to one degree of freedom on level $l+1$ for each C_i^l. Here, tentative prolongators will be generated by restricting all the vectors $p^{(0)}, p^{(j,k)}$ to the aggregates C_i^1. Each aggregate will be represented by a set of degrees of freedom, where every degree of freedom corresponds to one of the vectors $p^{(0)}$, $p^{(jk)}$ (see Fig. 3). The shape of the basis functions derived from the nonconstant polynomials depends on the position of the aggregate. When far away from the origin, basis functions derived from polynomials of higher degree contain a large low degree polynomial component which results in the violation of the uniform equivalence of discrete and continuous L^2-norms. This undesirable effect is suppressed by a local l^2 Gram-Schmidt orthogonalization process performed on each aggregate C_i^1. (see Fig. 4). Again, the resulting prolongator will be smoothed by the Jacobi smoother (see Fig. 5).

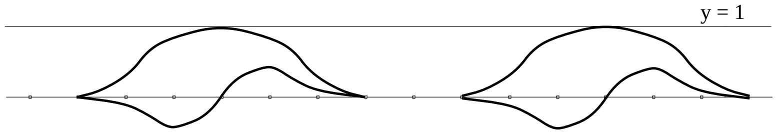

FIG. 5. *The final smoothed basis*

ALGORITHM 6.1 (COARSENING FOR HIGH-ORDER PROBLEMS). *Assume the number of degrees of freedom per node on the finest level to be constant. Let n_1 be the number of nodes on the finest level, and $df^1(i)$ be the list of degrees of freedom associated with the node i. Set $p^{1,(0)} = p^{(0)}, p^{1,(i,j)} = p^{(i,j)}, \quad i = 1, \ldots, K-1, \quad j = 1, \ldots, d$ (cf., (14)).*

Step 1 - Decomposition. *Generate a disjoint covering $\{C_i^l\}_{i=1}^{n_{l+1}}$ of the set of nodes $\{1, \ldots, n_l\}$ using Algorithm 5.1, where the strongly coupled neighborhood of i is defined by (13).*

Step 2 - Restriction. *For each aggregate C_i^l define the index set $\mathcal{D}_i^l$ of all degrees of freedom associated with nodes in C_i^l, i.e.*

$$\mathcal{D}_i^l = \bigcup_{j \in C_i^l} df^l(j).$$

For every $\mathcal{D}_i^l$ generate auxiliary sparse vectors $v^{l,i,1}, \ldots, v^{l,i,N_p}$ by

$$v^{l,i,1} = p^{l,(0)}|_{\mathcal{D}_i^l}, \; v^{l,i,2} = p^{l,(1,1)}|_{\mathcal{D}_i^l}, \quad v^{l,i,3} = p^{l,(1,2)}|_{\mathcal{D}_i^l}, \ldots, v^{l,i,N_p} = p^{l,(K-1,d)}|_{\mathcal{D}_i^l},$$

where $2K$ is the order of equation, d is the number of space variables ($\Omega \subset \mathbb{R}^d$), and $N_p = (K-1)d + 1$ is the number of the user supplied polynomials. $v|_I$ denotes the restriction of the vector to the index set in the sense that $(v|_I)_i = v_i$ iff $i \in I$, zero otherwise.

Step 3 - Gram Schmidt orthogonalization. *For each aggregate C_i^l update the set of associated sparse vectors generated in Step 2 by l^2 Gram-Schmitd orthogonalization process in the ordering $v^{l,i,1}, \; v^{l,i,2}, \ldots, \; v^{l,i,N_p}$ (i.e., vectors derived from low-degree polynomials are processed first). Note that the representation of the unity $v^{l,i,1}$ remains unchanged by the process.*

Step 4 - Building of auxiliary prolongators. *Generate the auxiliary prolongator $\tilde{P}_l$ whose $N_p(i-1)+j$-th column consists of the vector $v^{l,i,j}$ and create the corresponding coarse level list of degrees of freedom associated with node i*

$$df^{l+1}(i) = \{N_p(i-1)+1, \; N_p(i-1)+2, \ldots, N_p i\}, \quad i = 1, \ldots, n_{l+1}.$$

Step 5 - Representation of polynomials on the coarse level. *Generate vectors $p^{l+1,(0)}, p^{l+1,(11)}, \ldots, p^{l+1,(K-1,d)}$ satisfying*

$$p^{l,(0)} = \tilde{P}_l p^{l+1,(0)}, \quad p^{l,(1,1)} = \tilde{P}_l p^{l+1,(1,1)}, \ldots, \; p^{l,(K-1,d)} = \tilde{P}_l p^{l+1,(K-1,d)}.$$

As $\{C_i^l\}$ is a disjoint covering, the columns corresponding to different aggregates are l^2-orthogonal and consequently, the global Gram matrix given by columns *of $\tilde{P}_l$ is a block matrix. Therefore, $p^{l+1,(0)}$, $p^{l+1,(1,1)}, \ldots, p^{l+1,(K-1,d)}$ can be computed by solving the local problems with Gram matrices $G_i^l = \{(v^{l,i,j}, v^{l,i,k})_{l^2}\}_{j,k=1}^{N_p}$ generated by the columns of prolongator $\tilde{P}_l$ associated with C_i^l.*

Step 6 - Prolongator smoothing. *Improve the prolongator $\tilde{P}_l$ by smoothing step (10), (11), where scalar entries a_{ij} are replaced by blocks $A_{(ij)} = A_l(df^l(i), df^l(j))$.*

Step 7 - Polynomial preservation correction. *The polynomial preserving property potentially disturbed by the filtration of the smoother in (10) is restored as follows:*

- *Compute the error of polynomial preservation property by*
  $e^{l,(0)} = (I - \omega D^{-1} A_l) p^{l,(0)} - P_l p^{l+1,(0)}$,
  $e^{l,(1,1)} = (I - \omega D^{-1} A_l) p^{l,(1,1)} - P_l p^{l+1,(1,1)}$,
  $\vdots$
  $e^{l,(K-1,d)} = (I - \omega D^{-1} A_l) p^{l,(K-1,d)} - P_l p^{l+1,(K-1,d)}$,
  where ω is the parameter from (10).
- *For each aggregate of degrees of freedom D_i^l generated in Step 2, create the auxiliary sparse vectors*
  $w^{l,i,1} = e^{l,(0)}|_{\mathcal{D}_i^l}$, $w^{l,i,2} = e^{l,(1,1)}|_{\mathcal{D}_i^l}$, $\ldots$, $w^{l,i,N_p} = e^{l,(K-1,d)}|_{\mathcal{D}_i^l}$.
- *Generate the correction prolongator P_l^{corr} whose $N_p(i-1)+j$-th column is a sparse vector $w^{l,i,j}$.*
- *Compute the $N_p \times N_p$ block-diagonal matrix Q with $N_p \times N_p$ blocks $Q_{(ii)}$, whose columns consist of vectors*
  $p^{l+1,(0)}(df^{l+1}(i))$, $p^{l+1,(1,1)}(df^{l+1}(i))$, $\ldots$, $p^{l+1,(K-1,d)}(df^{l+1}(i))$, *where $df^{l+1}(i)$ is the list of degrees of freedom associated with node i on the level $l+1$ (see step 4).*
- *Set $P_l \leftarrow P_l + P_l^{corr} Q^+$, Q^+ denoting the pseudo-inverse of matrix Q.*

REMARK 6.2. *Algorithm 6.1 is invariant to the selection of basis within the span of basis functions associated with one node, since the group of degrees of freedom is always treated as a subspace, and the input vectors $p_k^{(i,j)}$ are expressed with respect to the current basis. This is an advantage even for second order problems, where the constant function may not be always represented by the constant vector due to scaling, cf., Experiment No. 5 below.*

6.2. Coarse Shape Functions. The final smoothed coarse basis functions resemble the standard shape functions for the Hermitean element with one degree of freedom for the value at the node and one degree of freedom for each derivative, cf., Fig. 5. This is true regardless of the choice of basis functions in the original problem (finest level), and makes the algebraic approach to coarsening possible.

6.3. Zero Energy Functions. For elasticity in 2D or 3D, the energy norm is not equivalent to $(H^1)^3$-seminorm on the factorspace modulo constants, so the approximation properties of coarse spaces built from constant functions depend on the constant from Korn's inequality. This results in deterioration of convergence if, for example, Dirichlet boundary conditions are prescribed only on a small part of the boundary. Therefore, it is reasonable to require that the coarse space contains locally functions with zero energy, which are rigid body modes. We then build the coarse space from these rigid body modes in exactly the same way as from the polynomials above.

A related technique that builds the coarse space from local generators of the nullspace is used in the so-called Balancing Domain Decomposition [10].

6.4. Linearly Dependent Coarse Functions. In some cases, e.g., when solving 3D elasticity using coarse spaces built from rigid body modes, the number of degrees of freedom per node on the finest level can be smaller than on coarse levels. Then, when the aggregate is small, tentative basis functions generated by step 2 of Algorithm 6.1 can be linearly dependent and some basis functions are dropped during the Gram-Schmidt modification in the step 3. In such a case, pseudoinverse is be used when representing polynomials on the coarse level in the step 5 and when inverting D in (10).

6.5. Complexity of the V-cycle. Following [15], define *algebraic complexity* Ω_A to be the number of nonzero entries in the matrices on all levels divided by the number of nonzero entries in the matrix A on the first level. The total cost of all smoothing is $2(\nu_1+\nu_2)\Omega_A\,\mathrm{nonz}(A_1)$ multiply/add operations. Computations of defects costs $\Omega_A\,\mathrm{nonz}(A_1)$ operations. A pessimistic estimate of the cost of prolongation or restriction between levels l and $l-1$ is $\mathrm{nonz}(P_l)\leq \mathrm{nonz}(A_l)\frac{\mathrm{ndof}(l+1)}{\mathrm{ndof}(l)}$, where $\mathrm{ndof}(l)$ is the number of degrees of freedom per node on level l. Note that $\mathrm{ndof}(l+1)=\mathrm{ndof}(l)$ for $l\geq 2$, but we may have $\mathrm{ndof}(2)\neq\mathrm{ndof}(1)$. Putting this all together, the number of operations for one V-cycle can be estimated as

$$(1+2(\nu_1+\nu_2)\Omega_A\,\mathrm{nonz}(A_1)+2\frac{\mathrm{ndof}(l+1)}{\mathrm{ndof}(l)}\mathrm{nonz}(A_1)+2\,\mathrm{nonz}(A_1)(\Omega_A-1).$$

7. Numerical Experiments . The experiments in this section demonstrate the favorable behavior of the method. The code is available by anonymous ftp to tiger.denver.colorado.edu in `/pub/people/pvanek`. Experiments 1-10 were performed on an IBM RS-6000/360 with 128 MBytes of memory. Experiment 11 was performed on one processor of SGI Power Challenge. In Experiments 7 - 11, the stiffness matrix was generated using UAI/NASTRAN. Rigid body modes were computed using the list of coordinates of nodes obtained from NASTRAN data structures and passed to *AMG* code as polynomials, cf., Section 6.3. In Experiments No. 6 – 11, we have run for comparison AMG with the coarse space built from constants, and from rigid body motions for elasticity and bilinear functions for the biharmonic equation. The results demostrate that the large coarse space improves convergence significantly.

The residual was measured in the l^2 norm. The iteration process was stopped once the relative residual became smaller than 10^{-5}. The rate of convergence is computed as an average reduction of l^2-norm of the residual per iteration. The coarsest problem order was 50 or smaller. Results of experiments are summed up in Tab. 7. The description of testing problems follows.

Experiment No. 1: Planar elasticity problem with the bilinear form

$$a(\mathbf{u},\mathbf{v})=\int_\Omega \lambda \mathrm{div}\,\mathbf{u}\mathrm{div}\,\mathbf{v}+2\mu\sum_{i,j}\epsilon_{ij}(\mathbf{u})\epsilon_{ij}(\mathbf{v})\,d\mathbf{x}, \tag{15}$$

							Setup time		Iteration time	
#	ord(A_1)	Problem	Coarse sp.	ρ	σ_A	L	CPU	REAL	CPU	REAL
1	21,358	Fig. 6	const.	0.08	1.23	3	2s	2s	3s	3s
2	10^6	(16), $q=0$	const.	0.10	1.56	5	310s	7262s*	458s	630s
3	68,921	(17)	const.	0.21	1.14	4	52s	142s	82s	91s
4a	160,000	(16), $q=0.1$	const.	0.11	1.65	5	44s	49s	41s	45s
4b	160,000	(16), $q=1.0$	const.	0.10	1.65	5	44s	51s	41s	43s
4c	160,000	(16), $q=10$	const.	0.10	1.65	5	44s	48s	41s	42s
5	21,358	Fig. 6	const.$^+$	0.09	1.24	3	5s	5s	8s	8s
6a	48,400	2d biharm.	bilin.$^+$	0.26	1.37	4	21s	30s	43s	47s
6b	48,400	2d biharm.	const	0.88	1.36	4	15s	34s	838s	917s
7a	48,400	(18)	bilin.$^+$	0.31	1.48	5	34s	36s	80s	85s
7b	48,400	(18)	const	0.79	1.48	5	18s	19s	618s	636s
8a	102,966	plate	RBM$^+$	0.02	1.12	4	50s	261s	80s	83s
8b	102,966	plate	const	0.27	1.12	4	37s	158s	180s	190s
9a	33,489	thin solid	RBM$^+$	0.12	1.85	5	51s	53s	55s	57s
9b	33,489	thin solid	const	0.88	1.21	5	25s	33s	1419s	1446s
10a	77,286	shell, Fig. 7	RBM$^+$	0.4	1.12	4	56s	78s	273s	324s
10b	77,286	shell, Fig. 7	const	0.89	1.12	4	45s	73s	1419s	1446s
11a	417,867	indust. solid	RBM$^+$	0.45	1.17	4	330s	440s	1025s	1210s
11b	417,867	indust. solid	const	0.71	1.06	4	190s	220s	1510s	1711s

*) Extensive swapping. +) Additional user input (14) used.
σ_A = algebraic complexity, ρ = rate of convergence, L = number of levels. Symbols const./bilin./RBM mean coarse space built from vectors of constants in each field/bilinear functions/rigid body modes.

TABLE 1
Results of the numerical experiments.

where ϵ is the strain tensor, $\epsilon_{ij}(\mathbf{u}) = \frac{1}{2}(\partial_i u_j + \partial_j u_i)$, $\lambda > 0$, $\mu > 0$ are the Lamé coefficients, and the Poisson ratio is $\kappa = \frac{\lambda}{2(\mu+\lambda)}$. Discretized on an unstructured mesh (Fig. 6). Constant coefficients, Poisson ratio 0.3. Discretized by P1 finite elements, number of nodes 10610, number of degrees of freedom 21358. Boundary conditions : Dirichlet and Neumann.

EXPERIMENT NO. 2: Anisotropic problem with jumps in coefficients (16), $q = 0$. Number of nodes 10^6. P1 elements, regular grid.

EXPERIMENT NO. 3: 3D problem (17) with random coefficients Number of nodes 68921. P1 elements, regular grid.

$$
\begin{gathered}
-\frac{\partial}{\partial x}\left(a\frac{\partial u}{\partial x}\right) - \frac{\partial}{\partial y}\left(a^{-1}\frac{\partial u}{\partial y}\right) + q\,u = f(x,y) \text{ on } \Omega = (0,1)\times(0,1), \\
u = 0 \text{ on } \partial\Omega, \\
a = 10^{-2} \text{ on } (0,0.5)\times(0,0.5),\ 1 \text{ on } (0,0.5)\times(0.5,1) \text{ and} \\
10^2 \text{ on } (0.5,1)\times(0,1).
\end{gathered}
\tag{16}
$$

$$
(17)\qquad \begin{array}{c} -\sum_{i=1}^{3} \frac{\partial}{\partial x_i}(\exp(r_i)\frac{\partial u}{\partial x_i}) = f(x,y), \quad \text{on } \Omega = (0,1)\times(0,1), \\ u = 0 \quad \text{on} \quad \partial\Omega, \\ r_i \text{ is a random number uniformly distributed in } [\ln(10^{-2}), \ln(10^{2})]. \end{array}
$$

EXPERIMENT NO. 4: 2D problem (16), a) $q = 0.1$, b) $q = 1$, c) $q = 10$. Number of nodes 160000. P1 elements, regular grid.

EXPERIMENT NO. 5: Elasticity problem (15) discretized on an unstructured mesh (Fig. 6) by P1 finite elements. The finite element basis was randomly scaled to show that performace does not deteriorate since the algorithm is invariant to scaling, due to the input of the constant function as a vector. Cf., Remark 6.2. Constant coefficients, Poisson ratio 0.3, number of nodes 10610, number of degrees of freedom 21358. Boundary conditions Dirichlet and Neumann.

EXPERIMENT NO. 6: Biharmonic problem discretized on a rectangular square grid by the standard 13-point difference scheme. Clusters of 4 points are considered one node with four degrees of freedom. Number of degrees of freedom 48400. Boundary conditions essential.

EXPERIMENT NO. 7: Fourth order problem (18) discretized on a regular square grid by the standard 13-point difference scheme. Clusters of 4 points are considered one node with four degrees of freedom. Number of degrees of freedom 48400. Boundary conditions essential.

$$
(18)\qquad \frac{\partial^4 u}{\partial x^4} + 2\frac{\partial^2}{\partial x^2}\left(e^{16xy}\frac{\partial^2 u}{\partial y^2}\right) + \frac{\partial^2}{\partial y^2}\left(e^{16xy}\frac{\partial^2 u}{\partial y^2}\right) = f(x,y) \;\text{ on } (0,1)\times(0,1)
$$

EXPERIMENT NO. 8: Plate 1 by 1 m, 0.0005 m thick. `QUAD4` UAI/NASTRAN elements. Number of nodes 17161. Number of degrees of freedom 102966. Boundary conditions one quarter of one side fixed (both displacements and rotations). Poisson ratio 0.3.

EXPERIMENT NO. 9: Thin solid 1 by 1 m, 0.0005 m thick. Discretized by 2 layers of UAI/NASTRAN `HEXA` elements. Number of nodes 11163. Number of degrees of freedom 33489. Boundary conditions one quarter of one thin face fixed. Elasticity (15) in 3D, constant coefficients, Poisson ratio 0.3.

EXPERIMENT NO. 10: Spherical shell (half sphere, see Fig. 7) with the diameter of 2 m, 0.0005 m thick. UAI/NASTRAN `QUAD4` elements. Number of nodes 12881. Number of degrees of freedom 77286. Constant coefficients, Poisson ratio 0.3.

EXPERIMENT NO. 11: An industrial problem, 3D elasticity, unstructured mesh of UAI/NASTRAN `HEXA` elements. Number of nodes 139289. Number of degrees of freedom 417867.

8. Acknowledgements. This research was supported by the National Science Foundation under grants ASC-9121431, ASC-9404734, and ASC-9217394. The authors would like to thank Universal Analytics, Inc., for providing the UAI/NASTRAN software and real-life testing data. The use of SGI Power Challenge was provided by National Center for Supercomputing Applications.

REFERENCES

[1] R. E. Alcouffe, A. Brandt, J. E. Dendy, and J. W. Painter, *The multi-grid methods for the diffusion equation with strongly discontinuous coefficients*, SIAM J. Sci. Stat. Comput., 2 (1981), pp. 430–454.

[2] J. Bramble and X. Zhang, *Multigrid methods for the biharmonic problem discretized by conforming C^1 finite elements on nonnested meshes.* Submitted.

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

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

[5] A. Brandt, S. F. McCormick, and J. W. Ruge, *Algebraic multigrid (AMG) for sparse matrix equations*, in Sparsity and Its Applications, D. J. Evans, ed., Cambridge Univ. Press, Cambridge, 1984.

[6] S. C. Brenner, *An optimal order nonconforming multigrid method for the biharmonic equation*, SIAM J. Numer. Anal., 26 (1989), pp. 1124–1138.

[7] T. F. Chan, S. Go, and J. Zou, *Multilevel domain decomposition and multigrid methods for unstructured meshes: Algorithms and theory*, Tech. Report CAM 95-24, Dept. Math., UCLA, May 1995.

[8] J. E. Dendy, M. P. Ida, and J. M. Rutledge, *A semicoarsening multigrid algorithm for SIMD machines*, SIAM J. Sci. Stat. Comput., 13 (1992), pp. 1460–1469.

[9] W. Hackbusch, *Multigrid Methods and Applications*, vol. 4 of Computational Mathematics, Springer–Verlag, Berlin, 1985.

[10] J. Mandel, *Balancing domain decomposition*, Comm. Numer. Meth. Engrg., 9 (1993), pp. 233–241.

[11] P. Oswald, *Hierarchical conforming finite element methods for the biharmonic equation*, SIAM J. Numer. Anal., 29 (1992), pp. 1610–1625.

[12] P. Peisker and D. Braess, *A conjugate gradient method and a multigrid method for Morley's finite element approximation of the biharmonic equation*, Numer. Math., 50 (1987), pp. 567–586.

[13] J. W. Ruge, *Algebraic multigrid (AMG) for geodetic survey problems*, in Prelimary Proc. Internat. Multigrid Conference, Fort Collins, CO, 1983, Institute for Computational Studies at Colorado State University.

[14] J. W. Ruge and K. Stüben, *Efficient solution of finite difference and finite element equations by algebraic multigrid (AMG)*, in Multigrid Methods for Integral and Differential Equations, D. J. Paddon and H. Holstein, eds., The Institute of Mathematics and its Applications Conference Series, Clarendon Press, Oxford, 1985, pp. 169–212.

[15] J. W. Ruge and K. Stüben, *Algebraic multigrid (AMG)*, in Multigrid Methods, S. F. McCormick, ed., vol. 3 of Frontiers in Applied Mathematics, SIAM, Philadelphia, PA, 1987, pp. 73–130.

[16] R. A. Smith and A. Weiser, *Semicoarsening multigrid on a hypercube*, SIAM J. Sci. Stat. Comput., 13 (1992), pp. 1314–1329.

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

[18] P. Vaněk, *Fast multigrid solver.* Applications of Mathematics, to appear.

[19] ———, *Acceleration of convergence of a two-level algorithm by smoothing trarnsfer operator*, Applications of Mathematics, 37 (1992), pp. 265–274.

[20] P. Vaněk, J. Mandel, and M. Brezina, *Algebraic multigrid on unstructured meshes*, SIAM J. Sci. Comp., Submitted.

[21] X. Zhang, *Multilevel Schwarz methods for the biharmonic Dirichlet problem*, SIAM J. Sci. Comput., 15 (1994), pp. 621–644.

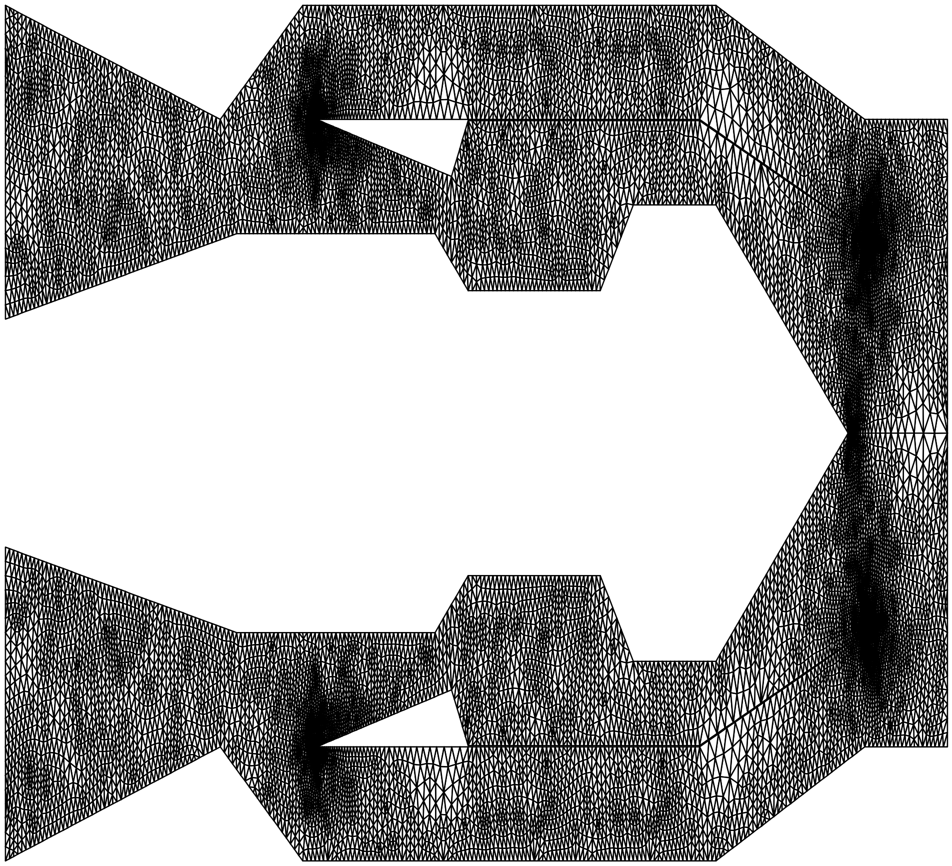

FIG. 6. *Jet engine, 2d elasticity. (Courtesy of Charbel Farhat, Center for Aerospace Engineering, University of Colorado, Boulder)*

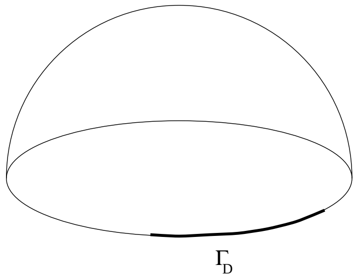

FIG. 7. *Spherical shell with Dirichlet values prescribed on* Γ_D