



ELSEVIER

Comput. Methods Appl. Mech. Engrg. 149 (1997) 267–287

**Computer methods
in applied
mechanics and
engineering**

Automated adaptive multilevel solver

J. Fish*, A. Suvorov, V. Belsky

Departments of Civil, Mechanical and Aerospace Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180, USA

Abstract

This paper presents an automated adaptive multilevel solver for linear (or linearized) system of equations. The multilevel aspect of the solver is aimed at securing an optimal rate of convergence, while keeping the size of the coarsest problem sufficiently small to ensure that the direct portion of the solution does not dominate the total computational cost. Adaptivity in terms of a priori selection of the number of levels (one or more) and construction of the optimal multilevel preconditioner is the key to the robustness of the method. The number of levels is selected on the basis of estimated conditioning, sparsity of the factor and available memory. The auxiliary coarse models (if required) are automatically constructed on the basis of spectral characteristics of individual aggregates (groups of neighboring elements).

An obstacle test consisting of twenty industry and model problems was designed to (i) determine the optimal values of computational parameters and to (ii) compare the adaptive multilevel scheme with existing state-of-the-art equation solvers including the Multifrontal solver [17] with the MMD reordering scheme, and the PCG solver with the nearly optimal Modified Incomplete Cholesky factorization preconditioner.

1. Introduction

Solution of linearized (or linear) systems of equations resulting from the finite element discretization of large-scale systems may consume over 50% and often up to 95% of total computational cost, and yet, the quest for an optimal general purpose equation solver is still in its embryonic stage. The ultimate goal is to develop an optimal solution scheme that will possess the following characteristics:

- (i) *Robustness*: Sensitivity to problem conditioning should be minimal.
- (ii) *Efficiency*: Speed and storage should be nearly proportional to the problem size.
- (iii) *Automation*: Black-box algorithm requiring as input matrix information only.
- (iv) *Predictability*: CPU time, disk and in-core storage should be a priori assessed.
- (v) *Generality*: Characteristics (i)–(iv) should hold for a wide range of problems ranging from indefinite systems to various problems in multiphysics.
- (vi) *Scalability*: Speed should be nearly proportional to the number of processors.

In the following we briefly summarize three categories of equation solution methods, including the direct solvers, the iterative solvers with a single-level preconditioner, and the iterative solvers with multi-level preconditioners. We will analyze their strengths and limitations and in light of what we perceive as an optimal solution scheme, we will attempt to construct a framework for such a solver. Attention will be restricted to symmetric positive definite systems in the present manuscript.

For the purpose of discussion below we will assume that the performance of linear solvers in terms of CPU time for symmetric positive definite systems can be approximated as CN^β , where N is the number of degrees-of-freedom, and C, β are solution method dependent parameters.

* Corresponding author.

The major advantage of direct solvers is their robustness, which is manifested by the fact that parameters C and β are independent of problem conditioning (except for nearly singular systems). Direct solvers are ideal for solving small to medium size problems since the constant C for direct methods is significantly smaller than for iterative solvers, but become prohibitively expensive for large-scale problems since the value of the exponent for direct solvers is higher than for iterative methods. To make direct solvers more efficient, various modifications of Gaussian elimination, which store and compute only the logical nonzeros of the factor matrix [1], have been developed. Nevertheless, fill-in cannot be avoided but only minimized, and serious consideration of iterative methods for large problems is increasingly accepted.

Single-level iterative methods with SSOR, Modified Incomplete Cholesky (MIC), or Element-By-Element (EBE) preconditioners are considered by many commercial finite element code developers (ANSYS, COSMOS, ALGOR) as most suitable for commercial applications. The value of exponent β for CG type methods with a single level preconditioner typically ranges between 1.17 to 1.33 [2] depending on the preconditioner, while the value of constant C increases with degradation in problem conditioning.

Multigrid-like methods, on the other hand, possess an optimal rate of convergence ($\beta = 1$), i.e. computational work required to obtain fixed accuracy is proportional to the number of discrete unknowns. The principal idea of a multigrid method consists of capturing the oscillatory response of the system by means of smoothing, whereas remaining lower frequency response is resolved on the auxiliary coarse grid. Nevertheless, multigrid methods (or multigrid preconditioners within the CG method) thus far have had only very little impact in computational mechanics. There seem to be two basic reasons:

(i) Commercial software packages must be able to automatically produce a full sequence of auxiliary discretizations (finite element or boundary element meshes) that are gradual coarsenings of the source discretization.

(ii) For optimal multigrid convergence, smooth solution components relative to a given discretization must be well approximated by subsequent coarser grids. Conventional or geometric multigrid methods cannot guarantee that a sequence of auxiliary discretizations will possess this approximation property for general applications. For example, what is a good coarse discretization for a frame structure or a wing structure, where each panel in the source mesh consists of a single or very few shell elements?

These difficulties motivated the development of the *Algebraic Multigrid* (AMG) [3] with the intent of providing a black box algebraic solver based on multigrid principles and exhibiting multigrid efficiency. While the geometric multigrid approach constructs its discretization sequence using auxiliary coarser grids, the AMG accomplishes the same goal on the basis of the information available in the source matrix of equations only. By this technique the coarse level variables are selected so as to satisfy certain criteria based on the source grid matrix. The most basic criterion is typically that each fine level degree-of-freedom should be strongly connected to some coarse level variable. However, the fact that the algebraic multigrid method uses information available in the source matrix only in constructing auxiliary discretizations results in a suboptimal rate of convergence.

The *aggregation based iterative solver* is a hybrid scheme where some minor extra information (depending on the type of aggregation scheme) might be used to construct a hierarchy of coarser problems, but no sequence of coarser discretization is required. The concept of aggregation was introduced by Leontief in 1951 [4] in the context input–output economics, where commodities in large-scale systems were aggregated to produce smaller systems.

The concept of aggregation has been utilized within the context of the multigrid method by Bulgakov [5,6] and Vanek [7]. In [5], aggregates consisting of non-intersecting groups of neighboring nodes were chosen to have translational degrees of freedom only, and consequently, the auxiliary coarse model was constructed without knowledge of nodal coordinates. Unfortunately, convergence was guaranteed only for scalar problems such as heat conduction. This algorithm has been improved in [6] by enriching the kinematics of the aggregate with rotational degrees of freedom (three in 3D, one in 2D) and by constructing the prolongation operator on the basis of nodal coordinates. In general this approach does not guarantee that the coarse model captures the entire null space of the aggregate, such as in the case of pinned connections in frames, or continuum problems where elements within an aggregate are connected at a single node. Furthermore, the convergence characteristics of this approach have been found to be unsatisfactory for poorly conditioned problems. These drawbacks motivated development of the smoothed aggregation concept [7]. By this technique a tentative piecewise interpolation field consisting of a null space of individual aggregates is first constructed and then corrected using a Jacobi smoother in an attempt to reduce the energy of coarse space basis functions. Our numerical experiments indicate that

although smoothed aggregation markedly improves the rate of convergence in well-conditioned continuum problems, computational efficiency in poor-conditioned problems such as a thin shell is not improved and in some cases degrades.

In the aggregation schemes [6,7] a typical coarsening ratio was about 3^{nsd} for the Laplace operator, where nsd is a number of space dimensions. For well-conditioned problems this is a nearly optimal ratio resulting in methods with remarkably low computational complexity. Unfortunately, for poorly conditioned systems, such as thin domain problems, problems with distorted geometries, severe anisotropies and heterogeneities, the coarse problem may fail to adequately capture the lower frequency response of the source problem, which in turn may significantly degrade the performance of aggregation-based schemes.

Based on the preceding analysis, it is evident that a nearly optimal solution scheme in the sense described earlier can be constructed within a framework of an adaptive multilevel scheme. The multilevel aspect of the solver is crucial to secure an optimal rate of convergence, while keeping the size of the coarsest problem sufficiently small to ensure that the direct portion of the solution does not dominate the total solution cost. Adaptivity in terms of a priori selection of number of levels (one or more) as well as construction of the optimal multilevel preconditioner is critical to ensure robustness of the method. Within such a framework the number of levels will be a priori selected on the basis of estimated conditioning, sparsity of the factor and available memory. The concept of the aggregation scheme developed in [6,7] will be generalized to permit automatic construction of a nearly optimal auxiliary coarse model based on the spectral characteristics of individual aggregates. We will denote such an adaptive multilevel scheme as a Generalized Aggregation Method (GAM).

The paper is organized as follows. Section 2 reviews the basic multigrid principles. Description of the Generalized Aggregation Method (GAM) is given in Section 3. Adaptive features including selection of number of levels, based on a priori estimation of condition number and sparsity of the factor, automated construction of an aggregated model by the tracing ‘stiff’ and ‘soft’ elements, adaptive construction of intergrid transfer operators, and various smoothing procedures are described in Section 4. In Section 5 we conduct numerical studies on 3D industry problems, such as a ring-strut-ring structure, casting setup in airfoil, nozzle for turbines, turbine blade and diffuser casing as well as on poorly conditioned shell problems, such as High Speed Civil Transport (HSCT), submarine, canoe and automobile body. Comparisons to the state-of-the-art direct [1] and iterative (PCG with Incomplete Cholesky preconditioner) solvers are also included in Section 5.

2. Multigrid principles

Consider a linear or linearized system of equations within a Newton–Raphson or related scheme

$$Ku = f \quad u \in R^n \quad f \in R^n \quad (1)$$

where K is an $n \times n$ symmetric positive definite and sparse matrix.

The following notation is adopted. Auxiliary grid functions are denoted with subscript 0. For example, u_0 denotes the nodal values of the solution in the auxiliary grid, where $u_0 \in R^m$, $m < n$. We also denote the prolongation operator from the coarse grid to the fine grid by Q :

$$Q : R^m \rightarrow R^n \quad (2)$$

The restriction operator Q^T from the fine-to-coarse grid is conjugated with the prolongation operator, i.e.:

$$Q^T : R^n \rightarrow R^m \quad (3)$$

In this section superscripts are reserved to indicate the iteration count. Let r^i be the residual vector in the i th iteration defined by

$$r^i = f - Ku^i \quad (4)$$

where u^i is the current approximation of the solution in the i th iteration.

The problem of the coarse grid correction for positive definite systems consists of the minimization of the energy functional on the subspace R^m , i.e.:

$$\frac{1}{2}(\mathbf{K}(\mathbf{u}^i + \mathbf{Q}\mathbf{u}_0^i), \mathbf{u}^i + \mathbf{Q}\mathbf{u}_0^i) - (\mathbf{f}, \mathbf{u}^i + \mathbf{Q}\mathbf{u}_0^i) \Rightarrow \min \quad \mathbf{u}_0^i \in \mathbf{R}^m \quad (5)$$

where (\cdot, \cdot) denotes the bilinear form defined by

$$(\mathbf{u}, \mathbf{v}) = \sum_{j=1}^n u_j v_j \quad \mathbf{u}, \mathbf{v} \in \mathbf{R}^n \quad (6)$$

A direct solution of the minimization problem (5) yields a classical two-grid procedure. Alternatively, one may introduce an additional auxiliary grid for \mathbf{u}_0 and so forth, leading to a natural multi-grid sequence. For the purpose of discussion here we will consider a two-grid process resulting from the direct minimization of (5) which yields

$$\mathbf{K}_0 \mathbf{u}_0^i = \mathbf{Q}^T (\mathbf{f} - \mathbf{K}\mathbf{u}^i) \quad (7)$$

where $\mathbf{K}_0 = \mathbf{Q}^T \mathbf{K} \mathbf{Q}$ is the restriction of the matrix \mathbf{K} . The resulting classical two-grid algorithm can be viewed as a two-step procedure:

(a) *Coarse grid correction*

$$\begin{aligned} \mathbf{r}^i &= \mathbf{f} - \mathbf{K}\mathbf{u}^i \\ \mathbf{u}_0^i &= \mathbf{K}_0^{-1} \mathbf{Q}^T \mathbf{r}^i \\ \tilde{\mathbf{u}}^i &= \mathbf{u}^i + \mathbf{Q}\mathbf{u}_0^i \end{aligned} \quad (8)$$

where $\tilde{\mathbf{u}}^i$ is a partial solution obtained after the coarse grid correction.

(b) *Smoothing*

$$\mathbf{u}^{i+1} = \tilde{\mathbf{u}}^i + \mathbf{D}^{-1} (\mathbf{f} - \mathbf{K}\tilde{\mathbf{u}}^i) \quad (9)$$

where \mathbf{D} is a preconditioner for smoothing. Any preconditioned iterative procedure which has good smoothing properties and requires little computational work per iteration step can, in principle, be used as a smoother in the multigrid process. In particular, various incomplete factorizations have been found to possess good smoothing characteristics.

Let \mathbf{u} be the exact solution of the source problem, then the error resulting from the coarse grid correction (8) can be cast into the following form

$$\tilde{\mathbf{e}}^i = \mathbf{u} - \tilde{\mathbf{u}}^i = (\mathbf{I} - \mathbf{Q}\mathbf{K}_0^{-1}\mathbf{Q}^T\mathbf{K})\mathbf{e}^i = (\mathbf{I} - \mathbf{C}^{-1}\mathbf{K})\mathbf{e}^i \quad (10)$$

where \mathbf{I} is the identity $n \times n$ matrix and $\mathbf{C}^{-1} = \mathbf{Q}\mathbf{K}_0^{-1}\mathbf{Q}^T$ is a coarse grid preconditioner. Likewise, the influence of smoothing on error reduction is given by

$$\mathbf{e}^{i+1} = \mathbf{u} - \mathbf{u}^{i+1} = (\mathbf{I} - \mathbf{D}^{-1}\mathbf{K})\tilde{\mathbf{e}}^i \quad (11)$$

and from Eqs. (9), (10) the error vector of the two-grid process with one post-smoothing iteration can be expressed as

$$\mathbf{e}^{i+1} = (\mathbf{I} - \mathbf{D}^{-1}\mathbf{K})(\mathbf{I} - \mathbf{C}^{-1}\mathbf{K})\mathbf{e}^i \quad (12)$$

Further denoting

$$\begin{aligned} \mathbf{M} &= \mathbf{I} - \mathbf{D}^{-1}\mathbf{K} \\ \mathbf{T} &= \mathbf{I} - \mathbf{Q}\mathbf{K}_0^{-1}\mathbf{Q}^T\mathbf{K} \end{aligned} \quad (13)$$

Eq. (12) with ν_1 post-smoothing and one ν_2 pre-smoothing iterations can be cast into the following concise form

$$\mathbf{e}^{i+1} = \mathbf{M}^{\nu_1} \mathbf{T} \mathbf{M}^{\nu_2} \mathbf{e}^i \quad (14)$$

Based on Eq. (13) it can be easily shown that \mathbf{T} is a projection operator, i.e. $\mathbf{T} = \mathbf{T}^2$, and hence $\|\mathbf{T}\|_{\mathbf{K}} = 1$.

Eq. (14) represents the sufficiency condition for the convergence of the multigrid method provided that the iterative procedure employed for smoothing is convergent, i.e. $\|M\|_k < 1$. For recent advances on convergence analysis for multigrid like methods we refer to [8].

In practice, however, the solution increment $u^{k+1} - u^k = P^{-1}r^k$ obtained from a single multigrid cycle is used in the determination of the search direction within the framework of the Conjugate Gradient method.

The inverse of the two-grid preconditioner, P^{-1} , with $v_1 = 1, v_2 = 0$ can be obtained from Eq. (12)

$$P^{-1} = D^{-1}(I - KC^{-1}) + C^{-1} \tag{15}$$

for which the closed form expression is given as

$$P = [I + (K - D)Q(Q^T D Q)^{-1} Q^T] D. \tag{16}$$

3. Generalized Aggregation Method (GAM)

In an aggregation scheme the coarse model is directly constructed from the source grid by decomposing the whole set of nodes into non-intersecting groups to be referred to as aggregates, and then for each aggregate assigning a reduced number of degrees of freedom. By doing so one reduces dimensionality of the source problem, while maintaining the compatibility of the solution. The key issue is how to approximate the solution on each aggregate so that the coarse model, to be referred to as an aggregated model, will effectively capture the lower frequency response of the source system.

We start by relating (Assertion 1) the optimal characteristics of the aggregated mesh to the intergrid transfer operator properties of individual aggregates and interface regions between the aggregates.

ASSERTION 1. The prolongation operator $Q : R^m \rightarrow R^n$ is considered optimal for fixed $m \leq n$ if $\|Q^T K Q\|_2$ is minimal for all Q satisfying $\|Q\|_2 = 1$ and $\text{rank } Q = m$. Furthermore, among all the block diagonal prolongation operators, where each block corresponds to the prolongation operator of individual aggregate, the optimal prolongation operator is such that

$$\begin{aligned} \max_{e,a} \{ \|K_0^a\|_2, \|K_0^e\|_2 \} &\Rightarrow \min(Q^a) \\ \text{Subjected to } \|Q^a\|_2 &= 1 \quad \text{rank } Q^a = m_a \quad \forall a \in G \end{aligned} \tag{17}$$

where superscripts a and e denote the aggregates and interface elements between the aggregates, respectively. N_A and N_E are the total numbers of aggregates and interface elements, respectively; $Q^a : R^{m_a} \rightarrow R^{n_a}$ and $Q^e : R^{m_e} \rightarrow R^{n_e}$ are the prolongation operators for aggregate a and interface element e ; $K_0^a = (Q^a)^T K^a Q^a$ and $K_0^e = (Q^e)^T K^e Q^e$ are the corresponding restricted stiffness matrices.

Note that minimization is carried out with respect to the prolongation operator for the aggregates only, that the prolongation operator Q^e for each element in the interface region is uniquely determined from $\{Q^a\}_{a=1}^{N_A}$.

DISCUSSION. Let Φ and Λ be an $n \times n$ matrix of unitary eigenvectors and a diagonal $n \times n$ matrix of eigenvalues of the stiffness matrix K , respectively, partitioned as

$$[\Phi_0 \quad \Phi_1] \quad \text{and} \quad \begin{bmatrix} \Lambda_0 & 0 \\ 0 & \Lambda_1 \end{bmatrix}$$

so that $\|\Lambda_0\|_2 < \tilde{\gamma}$ and $Q = \Phi_0 \alpha$ where Φ_0 consists of m unitary eigenvectors and $\Phi^T \Phi = I$.

The spectral norm of stiffness matrix of the auxiliary model $\|K_0\|_2$ can be bounded utilizing the consistency condition [10]:

$$\|K_0\|_2 = \|Q^T K Q\|_2 = \|\alpha^T \Lambda_0 \alpha\|_2 \leq \|\alpha\|_2^2 \|\Lambda_0\|_2 \tag{18}$$

Furthermore, since $\|Q\|_2 = \|Q^T Q\|_2^{1/2} = \|\alpha^T \Phi_0^T \Phi_0 \alpha\|_2^{1/2} = \|\alpha\|_2 = 1$ we obtain

$$\|K_0\|_2 \leq \tilde{\gamma} \tag{19}$$

which completes the first part of the assertion. For the second part we bound the maximum eigenvalue of the system [9] by the maximum eigenvalue of the subdomain (aggregate or interface element)

$$\|\mathbf{K}_0\|_2 \leq \max_{e,a} \{\|\mathbf{K}_0^e\|_2, \|\mathbf{K}_0^a\|_2\} \quad (20)$$

Assertion 1 states that the quality of the aggregated model is governed by the maximum spectral radius of individual subdomains. The next assertion formulates certain minimum requirements for the construction of \mathbf{Q}^a aimed at ensuring the lower bound of the minimal eigenvalue of the two-grid preconditioned system $\mathbf{P}^{-1}\mathbf{K}$. It assumes the worst case scenario where smoothing does not affect lower frequency response errors.

ASSERTION 2. Consider the two-level method with $v_1 = 1$, $v_2 = 0$ and smoothing affecting only high frequency modes of error. Then the lower frequency response of the two-level system characterized by the lower bound of the Rayleigh quotient $\rho(\mathbf{x}) = (\mathbf{x}^T \mathbf{K} \mathbf{x}) / (\mathbf{x}^T \mathbf{P} \mathbf{x})$ is governed by the lowest eigenvalue among all the aggregates provided that the prolongation operator of each aggregate is spanned by the space, which at a minimum contains the null space of that aggregate.

DISCUSSION. Let ϕ^a and λ^a be n_a eigenvectors and eigenvalues of the aggregate a . Nodal solution u^a on each aggregate can be expressed as a linear combination of its eigenvectors

$$\mathbf{u}^a = \phi^a \hat{\mathbf{u}}^a \quad (21)$$

whereas the global solution vector, denoted as $\mathbf{u} = \phi \hat{\mathbf{u}}$ can be assembled from its aggregates. Let

$$[\phi_0 \quad \phi_1] \quad \text{and} \quad \begin{bmatrix} \lambda_0 & \mathbf{0} \\ \mathbf{0} & \lambda_1 \end{bmatrix}$$

be the partitions of ϕ and λ , respectively, such that $\|\lambda_0\|_2 < \hat{\gamma}$. Then the system of equations can be transformed into hierarchical form:

$$\begin{bmatrix} \hat{\mathbf{K}}_{00} & \hat{\mathbf{K}}_{01} \\ \hat{\mathbf{K}}_{10} & \hat{\mathbf{K}}_{11} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{u}}_0 \\ \hat{\mathbf{u}}_1 \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{f}}_0 \\ \hat{\mathbf{f}}_1 \end{bmatrix} \quad (22)$$

where

$$\hat{\mathbf{u}}_i = \phi_i^T \mathbf{u}_i \quad \hat{\mathbf{f}}_i = \phi_i^T \mathbf{f}_i \quad \hat{\mathbf{K}}_{ij} = \phi_i^T \mathbf{K} \phi_j \quad i, j = 0, 1 \quad (23)$$

Similarly, any smoothing preconditioner \mathbf{D} can be transformed into hierarchical form $\hat{\mathbf{D}}$ as follows:

$$\hat{\mathbf{D}}_{ij} = \phi_i^T \mathbf{D} \phi_j \quad (24)$$

Let $\mathbf{Q} = \phi_0$; then the prolongation operator $\hat{\mathbf{Q}}$ defined in hierarchical basis is given

$$\hat{\mathbf{Q}} = \begin{bmatrix} \phi_0^T \\ \phi_1^T \end{bmatrix} \mathbf{Q} = \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix} \quad (25)$$

Substituting Eqs. (22)–(25) into (16) yields the two grid preconditioner $\hat{\mathbf{P}}$ ($v_1 = 1$, $v_2 = 0$) defined in the hierarchical basis

$$\hat{\mathbf{P}} = \begin{bmatrix} \hat{\mathbf{K}}_{00} & \mathbf{0} \\ \hat{\mathbf{K}}_{10} & \hat{\mathbf{D}}_{11} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \hat{\mathbf{K}}_{00} \\ \mathbf{0} & \hat{\mathbf{K}}_{10} - \hat{\mathbf{D}}_{10} \end{bmatrix} \hat{\mathbf{D}}_{00}^{-1} \hat{\mathbf{D}}_{01} \quad (26)$$

If we further assume that smoothing affects higher frequency response only in the sense that

$$\hat{\mathbf{D}}_{00}^{-1} \hat{\mathbf{u}}_0 = \mathbf{0} \quad \forall \hat{\mathbf{u}}_0 \quad (27)$$

then the resulting two-level preconditioner can be cast into the block Gauss–Seidel form:

$$\hat{\mathbf{P}} = \begin{bmatrix} \hat{\mathbf{K}}_{00} & \mathbf{0} \\ \hat{\mathbf{K}}_{10} & \hat{\mathbf{D}}_{11} \end{bmatrix} \quad (28)$$

To estimate the lower bound of Rayleigh quotient of the two-level preconditioned system, we utilize again the theorem that bounds the lower eigenvalue of the system (aggregated model) by the minimum eigenvalue of any subdomain i consisting of either aggregates a or interface elements e :

$$\min_{i, \hat{x}} \{\rho^i(\hat{x}^i)\} \leq \min_{\hat{x}} \{\rho(\hat{x})\} \tag{29}$$

It remains for us to examine under which condition the Rayleigh quotient on each aggregate or interface element is bounded from below. For the two-grid preconditioner given in (28) the Rayleigh quotient for each aggregate or interface element is given as

$$\rho^i(\hat{x}^i) = \frac{(\hat{x}_0^i)^T \hat{K}_{00}^i \hat{x}_0^i + 2(\hat{x}_0^i)^T \hat{K}_{01}^i \hat{x}_1^i + (\hat{x}_1^i)^T \hat{K}_{11}^i \hat{x}_1^i}{(\hat{x}_0^i)^T \hat{K}_{00}^i \hat{x}_0^i + (\hat{x}_0^i)^T \hat{K}_{01}^i \hat{x}_1^i + (\hat{x}_1^i)^T \hat{D}_{11}^i \hat{x}_1^i} \tag{30}$$

Let $\mathcal{N}(\hat{K}_{00}^i)$ be the null space of \hat{K}_{00}^i defined as

$$\mathcal{N}(\hat{K}_{00}^i) = \{(\hat{x}_0^i)^T \hat{K}_{00}^i \hat{x}_0^i = 0 \quad \forall \hat{x}_0^i \in \text{SPAN}(\phi_0^i)\} \tag{31}$$

Then, the Rayleigh quotient is bounded from below $\rho^i(\hat{x}^i) > 0$ if \hat{K}_{00}^i contains all the rigid body modes of \hat{K}^i , i.e. $\mathcal{N}(\hat{K}_{00}^i) = \mathcal{N}(\hat{K}^i)$. This condition can be easily satisfied if the prolongation operator for each aggregate is spanned by the space containing the rigid body modes of that aggregate. In addition, for all interface elements $\mathcal{N}(\hat{K}_c^e) = \emptyset$, where \hat{K}_c^e is the interface element stiffness matrix constrained along the boundary between interface elements and aggregates. Loosely speaking, each interface element should be connected to aggregates at a number of degrees of freedom greater or equal than the dimension of the null space of that element $\dim\{\mathcal{N}(\hat{K}^e)\}$.

So far, we have proposed how to assess the quality of intergrid transfer operators (Assertion 1) and what are the properties that it should maintain (Assertion 2). In the subsequent proposition we describe a heuristic approach, which on the basis of the two assertions, attempts to construct a nearly optimal aggregated model.

PROPOSITION. For given $\{m_a\}_{a=1}^{N_a}$ a nearly optimal aggregation model can be constructed if (i) the prolongation operator $Q^a : \mathbf{R}^{m_a} \rightarrow \mathbf{R}^{n_a}$ on aggregate a is spanned by $m_a \leq n_a$ eigenvectors corresponding to m_a lowest eigenvalues on aggregate a , where $m_a \geq \dim\{\mathcal{N}(\hat{K}^a)\}$, and (ii) in forming the aggregated model soft elements determined by the Gerschgorin upper bound of their maximal eigenvalue $\max_i(\sum_j |k_{ij}^e|)$ are placed at the interface, where $K^e = [k_{ij}^e]$.

DISCUSSION. We first show that for fixed m_a the prolongation operator Q^a that minimizes $\|K_0^a\|_2$ is obtained as a linear combination of m_a lowest eigenvectors of K^a . This fact directly follows from Eq. (19) in the context of individual aggregates

$$\|K_0^a\|_2 \leq \gamma^a \quad \forall Q^a = \{\|Q^a\|_2 = 1 \quad \text{rank } Q^a = m_a\} \tag{32}$$

where γ^a is the maximum eigenvalue of eigenvectors spanning the space of Q^a . Furthermore, if we select $\gamma^a \leq \gamma \quad \forall a$, then the spectral norm of individual aggregates does not exceed user prescribed tolerance γ .

The spectral radius of the restricted interface element stiffness matrix K_0^e is given as

$$\|K_0^e\|_2 = \|(Q^e)^T K^e Q^e\|_2 \leq \|Q^e\|_2^2 \|K^e\|_2 \tag{33}$$

Since Q^e is a diagonal block of Q it can be easily shown that $\|Q^e\|_2 \leq \|Q\|_2 = 1$ and thus using Gerschgorin theorem for the maximal eigenvalue of K follows that

$$\|K_0^e\|_2 \leq \max_i \left(\sum_j |k_{ij}^e| \right) \tag{34}$$

where $K^e = [k_{ij}^e]$.

4. Adaptive features

This section describes four features of adaptivity built into the adaptive equation solver: (i) Selection of number of levels; (ii) Automated model aggregation; (iii) Construction of prolongation operator; (iv) Adaptive smoothing.

Section 4.1 describes a decision graph based methodology aimed at selecting an optimal number of levels on the basis of estimated conditioning, sparsity and memory requirements. Attention is restricted to one- and two-level methods, single right-hand side vector and single processor computer architectures. The lowest eigenvalue of the source problem is estimated using the Lanczos method applied to the aggregated model constructed by utilizing only rigid body modes of each aggregate. The estimate of the lowest eigenmode is then improved by using a Gauss–Seidel smoother in an attempt to reduce the energy absorbed at the interface between the aggregates. A model problem is constructed to investigate the effectivity of the estimator.

Next, we present the algorithm for automated construction of aggregates on the element-by-element basis as opposed to the node-by-node procedure employed in [6,7]. We present two versions of the aggregate formation algorithm: the basic version which utilizes topological information only, and the adaptive version which in addition to the topological information utilizes elemental stiffness matrices in the process of the aggregated model construction.

The third adaptive feature is related to the selection of parameter γ , which plays a central role in constructing the prolongation operator. This parameter has a direct effect on the restriction of the stiffness matrix, the sparsity pattern of the resulting auxiliary stiffness matrix and the effectiveness of the auxiliary model to capture the lower frequency response.

For optimal performance in terms of CPU time we adopt the Modified Incomplete Cholesky (MIC) factorization for pre- and post-smoothing. The number of fill-ins as well as a diagonal-scaling needed to preserve the positive definiteness of the system and to provide the fastest rate of convergence of the iterative process are determined based on the parametric study. For optimal performance in terms of memory requirements we utilize the SSOR smoother, which does not require additional storage as opposed to MIC based methods.

4.1. Selection of number of levels

In this section we describe condition number and sparsity estimators. Decision graphs aimed at selecting optimal number of levels are given in Section 5.

4.1.1. Condition number estimates

The exact calculation of the condition number is not practical and thus only an estimate is sought. The following predictor–corrector approach is employed for estimation of the lowest eigenmode.

(i) *Predictor*. Compute the lowest eigenmode of the source grid on the subspace $\mathbf{R}^{\hat{m}_a}$ defined as

$$\hat{\mathbf{Q}}^a : \mathbf{R}^{\hat{m}_a} \rightarrow \mathbf{R}^{\hat{n}_a} \quad \forall a \quad (35)$$

where $\hat{m}_a = \dim\{\mathcal{N}(\mathbf{K}^a)\}$ is the dimension of the null space of \mathbf{K}^a or the number of rigid body modes in the aggregate a . The corresponding constrained minimization problem can be stated as follows:

Find the vector $\hat{\mathbf{v}}_0 \in \mathbf{R}^{\hat{m}}$, which minimizes $(\mathbf{K}\hat{\mathbf{Q}}\hat{\mathbf{v}}_0, \hat{\mathbf{Q}}\hat{\mathbf{v}}_0)$ subjected to the constraint $(\hat{\mathbf{Q}}\hat{\mathbf{v}}_0, \hat{\mathbf{Q}}\hat{\mathbf{v}}_0) = 1$, where $\hat{\mathbf{Q}} = \{\hat{\mathbf{Q}}^a\}_{a=1}^{\hat{m}_a}$ is a prolongation operator of the smallest rank satisfying Assertion 2. The resulting eigenvalue problem directly follows from the constrained minimization problem and is given below:

$$(\hat{\mathbf{Q}}^T \mathbf{K} \hat{\mathbf{Q}} - \lambda_{\min}^0 \hat{\mathbf{Q}}^T \hat{\mathbf{Q}}) \hat{\mathbf{v}}_0 = 0 \quad (36)$$

(ii) *Corrector*. Compute the ‘smoothed’ eigenpair, $(\hat{\mathbf{w}}, \hat{\lambda}_{\min})$, by prolongating $\hat{\mathbf{v}}_0$ onto the source grid and smoothing it using either Gauss–Seidel or Jacobi preconditioner $\hat{\mathbf{D}}$:

$$\hat{\mathbf{w}} = \frac{(\mathbf{I} - \hat{\mathbf{D}}^{-1} \mathbf{K}) \hat{\mathbf{Q}} \hat{\mathbf{v}}_0}{\|(\mathbf{I} - \hat{\mathbf{D}}^{-1} \mathbf{K}) \hat{\mathbf{Q}} \hat{\mathbf{v}}_0\|} \quad \hat{\lambda}_{\min} = \hat{\mathbf{w}}^T \mathbf{K} \hat{\mathbf{w}} \quad (37)$$

The maximum eigenvalue of $\mathbf{K} = [k_{ij}]$ is estimated using the Gerschgorin theorem:

$$\hat{\lambda}_{\max} \leq \max_i \left(\sum_j |k_{ij}| \right) \tag{38}$$

and the estimated condition number, κ , is given by $\kappa = \hat{\lambda}_{\max} / \hat{\lambda}_{\min}$.

We now consider a model problem in an attempt to estimate the effectivity of the condition number estimator. Numerical experiments for 3D and shell problems will be conducted in Section 5 to assess the accuracy of the estimator in multidimensions and the computational effort involved.

The model problem consists of 1D N linear elements of equal size. The two ends are constrained. The resulting stiffness is a tridiagonal $(N - 1) \times (N - 1)$ matrix of the form $\mathbf{K} = \text{tridiag}\{-1, 2, -1\}$. Its smallest eigenvalue is

$$\lambda_{\min}^{\text{exac}} = 4 \left(\sin \frac{\pi}{2N} \right)^2$$

In the aggregated model each aggregate consists of two neighboring elements and a single interface element between the aggregates. For each aggregate a single rigid body mode is utilized to construct the following $(N - 1)/3 \times N - 1$ restriction matrix:

$$\hat{\mathbf{Q}}^T = \begin{bmatrix} 1 & 1 & 1 & & & \\ & & 1 & 1 & 1 & \\ & & & & 1 & 1 & 1 & \dots \end{bmatrix} \tag{39}$$

The initial value of the minimal eigenvalue λ_{\min}^0 is computed symbolically from Eq. (36) using Maple [18].

$$\lambda_{\min}^0 = \frac{4}{3} \left(\sin \frac{3\pi}{2(N+2)} \right)^2 \tag{40}$$

The minimal eigenvalue corresponding to the eigenvector smoothed by a single Jacobi iteration (37) is given by

$$\begin{aligned} \hat{\lambda}_{\min}^{\text{Jac}} &= \frac{\sum_{k=1}^{(N-1)/3} \left(\left(\sin \frac{3k\pi}{N+2} \right)^2 - \sin \frac{3k\pi}{N+2} \sin \frac{3(k+1)\pi}{N+2} \right)}{\left(\sum_{k=1}^{(N-1)/3} \left(2 \left(\sin \frac{3k\pi}{N+2} \right)^2 + \sin \frac{3k\pi}{N+2} \sin \frac{3(k+1)\pi}{N+2} \right) \right) - \frac{1}{4} \left(\left(\sin \frac{3\pi}{N+2} \right)^2 + \left(\sin \frac{(N-1)\pi}{N+2} \right)^2 \right)} \end{aligned} \tag{41}$$

For the purpose of comparison we expand the expressions of minimal eigenvalue in Taylor’s series:

$$\begin{aligned} \lambda_{\min}^{\text{exac}} &= \pi^2 \left(\frac{1}{N} \right)^2 - \frac{1}{12} \pi^4 \left(\frac{1}{N} \right)^4 + O \left(\left(\frac{1}{N} \right)^6 \right) \\ \lambda_{\min}^0 &= 3\pi^2 \left(\frac{1}{N} \right)^2 - 12\pi^2 \left(\frac{1}{N} \right)^3 + O \left(\left(\frac{1}{N} \right)^4 \right) \\ \hat{\lambda}_{\min}^{\text{Jac}} &= \frac{3}{2} \pi^2 \left(\frac{1}{N} \right)^2 - 6\pi^2 \left(\frac{1}{N} \right)^3 + O \left(\left(\frac{1}{N} \right)^4 \right) \end{aligned} \tag{42}$$

For example, for $N = 31$, we obtain $\lambda_{\min}^{\text{exac}} = 0.01026$, $\lambda_{\min}^0 = 0.027$ and $\hat{\lambda}_{\min}^{\text{Jac}} = 0.01372$. Moreover, with a stronger smoother, such as a single Symmetric Gauss–Seidel iteration, further improvement can be obtained, yielding $\hat{\lambda}_{\min}^{\text{Gauss}} = 0.0108$.

REMARK 1. The prolongation operator $\hat{\mathbf{Q}}$ representing the rigid body modes of each aggregate as well as the restricted stiffness matrix $\hat{\mathbf{Q}}^T \mathbf{K} \hat{\mathbf{Q}}$ (36) coincides with the corresponding lower order blocks in the two level scheme and thus can be utilized in the coarse model construction by orthogonalizing Lanczos vectors to the eigenvectors found earlier.

4.1.2. Sparsity estimates

We define the sparsity of the system, as $\omega = nz(\mathbf{L})/n$, where $nz(\mathbf{L})$ is the number of nonzero terms in the factor \mathbf{L} , and n is the number of unknowns. The sparsity of the factor depends on the reordering scheme. Here, we consider only large sparse systems arising from the finite element discretization and heuristic reordering schemes which attempt to reduce the number of nonzeros in the factor irrespective of any band or envelope structure. In all our numerical examples, a Multiple Minimum Degree (MMD) algorithm [1] implemented in NASA Langley VSS code [17] is considered.

For regular grids theoretical estimates of ω exist [17]. For 2D regular grids the best possible reordering scheme provides the lower bound, $\omega = O(\log n)$, whereas the so-called imperfect strategy yields $\omega = O(n^{\log_3 4 - 1})$.

For general unstructured grids no such estimates exist, and thus reordering algorithm has to be carried out in the preprocessing stage to estimate the system sparsity and memory requirements. Fortunately, for large scale problems the ordering time used by the MMD algorithm is less than 5% of the total CPU time required for numerical factorization ([1], see also Table 1 in Section 5).

4.2. Aggregation algorithm

Prior to describing the technical details of the aggregation algorithm, we introduce the concept of 'stiff' and 'soft' elements which is utilized in the process of aggregate formation.

The element is considered 'stiff' if the spectral radius of its stiffness matrix is relatively large compared to other elements and vice versa. Following the Proposition in Section 3, we will attempt wherever possible to place 'soft' elements at the interface between the aggregates, and 'stiff' elements within the aggregates. This approach is a counterpart of the idea of 'weak' and 'strong' nodal connectivity employed in [7] in the context of node-by-node aggregation.

The approximation for the maximum eigenvalue can be easily estimated using the Gerschgorin theorem in the context of the element stiffness matrices:

$$\max_a \lambda_a^e \leq \beta^e \quad \beta^e = \max_i \left(\sum_j |k_{ij}^e| \right) \quad (43)$$

In the remainder of this subsection we focus on the aggregation algorithm.

Consider the finite element mesh containing N_E elements and N_N nodes. Let $C(i)$ be the set of nodes belonging to the element E^i :

$$C(i) = \{N^j : N^j \in E^i\} \quad (44)$$

The goal of the aggregation algorithm described below is to construct a set of N_A aggregates denoted as

$$A = \{A^i, i \in [1, N_A]\} \quad (45)$$

satisfying the following conditions:

(i) *Element-by-element aggregation*

$$A^i = \{E^j, j \in M_E^i\} \quad (46)$$

$M_E^i \subset [1, N_E]$ - the set of element numbers corresponding to the aggregate i

(ii) *Disjoint covering*. Elements belonging to different aggregates cannot be neighbors. Two elements E^i and E^j are considered to be neighbors if

$$C(i) \cap C(j) \neq \emptyset \quad (47)$$

(iii) *Full nodal covering*. Each node belongs to some aggregate:

$$\bigcup_{\{i: E^i \in A\}} C(i) = \{N^j, j \in [1, N_N]\} \quad (48)$$

(iv) *Marking the 'slave' nodes and nodes with essential boundary conditions as separate aggregates*. Each

node containing either essential boundary condition and/or ‘slave’ degree(s)-of-freedom, which depends on so-called ‘master’ degree(s)-of-freedom, is considered as an aggregate. Denote the set of such nodes as N_{NB}

Step 1: Setup

(1.1) For each node N^j , $j = [1, N_N]$ select the elements containing this node:

$$B(j) = \{E^i : N^j \in E^i\} \tag{49}$$

(1.2) For each element E^i , $i = [1, N_E]$ select the set of neighboring elements $NE(i)$, that are the elements containing common nodes:

$$NE(i) = \{E^k : E^k \in B(j), j \in C(i)\} \setminus E^i \tag{50}$$

Step 2: Start-up aggregation

(2.1) Define the set of elements NA available for aggregation. These are all the elements which do NOT contain nodes with essential boundary conditions or the ‘slave’ nodes:

$$NA = [1, N_E] \setminus \{B(j), N^j \in N_{NB}\} \tag{51}$$

(2.2) Find the ‘peripheral’ element E^s , that is, the element with minimal number of neighbors:

$$s = \arg \min_{i \in NA} |NE(i)| \tag{52}$$

where $|X|$ is a number of elements in the set X . Element E^s is a starting element for the aggregation algorithm.

(2.3) Setup:

- the current aggregate counter $i = 1$;
- the set of interface elements $NI = [1, N_E] \setminus NA$, i.e. the elements between different aggregates.

Step 3: Formation of the current aggregate

Basic aggregation version:

Aggregate A^i contains the element E^s and all its available neighbors:

$$A^i = E^s \cup (NE(s) \cap NA) \tag{53}$$

Adaptive aggregation version:

Aggregate A^i contains the element E^s and those of its available neighbors which satisfy the relative stiffness condition:

$$A^i = E^s \cup \{E^j \in NE(s) \cap NA, \beta^j \geq \mu \beta^s\} \tag{54}$$

where β^j is a Gerschgorin upper bound on the stiffness matrix maximal eigenvalue of the element E^j , and μ is a coarsening parameter.

Step 4: Update the sets of the interface and available elements

(4.1) Update the set of the interface elements:

$$NI = NI \cup \{(E^k \in NE(j), E^j \in A^i) \cap (E^k \notin A^i)\} \tag{55}$$

(4.2) Update the set of the available elements:

$$NA = NA \setminus \{(E^k \in NE(j), E^j \in A^i) \cup A^i\} \tag{56}$$

Step 5: Find the new starting element

(5.1) Form the set of ‘frontal’ elements FR , that are available elements neighboring to the interface elements:

$$FR = \{(E^k \in NE(j), E^j \in NI) \cap NA\} \tag{57}$$

(5.2) Basic version: select arbitrary new starting element belonging to FR :

$$E^s \in FR \quad (58)$$

Adaptive version: select the stiffest new starting element from FR :

$$s = \arg \max_{j: E^j \in FR} (\beta^j) \quad (59)$$

Step 6: Stopping criteria

If $FR = \emptyset$ then stop; else $i = i + 1$ and repeat Steps 3–6.

REMARK 2. For simplicity we presented only the aggregation algorithm for lower order elements. In the case of higher order elements the ‘full nodal covering’ requirement may not be satisfied at the completion of the algorithm described above. There will be a significant number of nodes belonging to the elements in the interface region giving rise to a very large auxiliary coarse model. To further reduce the size of the auxiliary model, the same aggregation algorithm is recursively applied for the interface elements only until all the nodes would be covered by some aggregate. This procedure also provides a ‘cleaning’ phase to ensure that all nodes in the source grid are included within one of the aggregates.

REMARK 3. A similar scheme can be applied for the p -type discretization with the only exception being that the aggregates may contain only a single element in order to reduce the aggregate size. Higher-order modes in the interface region are treated as indicated in Remark 2.

REMARK 4. The aggregation algorithm described above allows us to deal with multi-point constraints in a conventional way since the elements containing the ‘slave’ nodes form a separate aggregate. Each multi-point constraint can be represented as follows:

$$x_s = Tx_m \quad (60)$$

where x_s , x_m are the ‘slave’ and ‘master’ degrees-of-freedom, respectively; T is a transformation matrix representing the multi-point constraint (MPC) data. In accordance with (60) the vector $x = (x_s, x_m)^T$ can be expressed as

$$x = \begin{bmatrix} T \\ I \end{bmatrix} x_m \quad (61)$$

Given the decomposition of the element stiffness matrix K_e for elements containing the ‘slave’ degrees-of-freedom

$$K_e = \begin{bmatrix} K_e^{ss} & K_e^{sm} \\ K_e^{ms} & K_e^{mm} \end{bmatrix} \quad (62)$$

the modified element stiffness matrix \tilde{K}_e corresponding to master degrees-of-freedom only is given by:

$$\tilde{K}_e = T^T K_e^{ss} T + K_e^{ms} T + T^T K_e^{sm} + K_e^{mm} \quad (63)$$

4.3. Adaptive construction of prolongation operator

One of the key issues in the proposed aggregation procedure is a selection of the coarse model parameter γ . All the eigenvectors of the eigenvalue problem on each aggregate corresponding to the eigenvalues $\lambda^a \leq \gamma$ are included within the diagonal block of the global prolongation operator. In order to make this parameter dimensionless, the eigenvalue problem on each aggregate is formulated in the following manner:

$$K^a \phi^a = \lambda^a D^a \phi^a \quad (64)$$

where D^a is a diagonal of K^a . Typically 6–50 modes are needed to satisfy $\lambda^a \leq \gamma$ requirement. The Lanczos algorithm with partial orthogonalization [12] was adopted.

The value of the parameter γ determines the effectiveness of coarse grid correction. In the limit as

$\gamma \rightarrow \max_{\alpha} \lambda^{\alpha}$, the auxiliary problem captures the response of the source system for all frequencies and therefore the two-level procedure converges in a single iteration even without smoothing. On the negative side, for large values of γ , eigenvalue analysis on each aggregate becomes prohibitively expensive and the auxiliary matrix becomes both large and dense. At the other extreme in the limit as $\gamma \rightarrow 0$, the prolongation operator contains the rigid body modes of all the aggregates only, and thus the auxiliary problem becomes inefficient for ill-posed problems.

4.4. Adaptive smoothing

Selection of smoothing procedure is another important issue as the cost of smoothing is a major expense in multilevel procedures. Comprehensive studies conducted in [11] revealed that one of the most efficient smoothing schemes is based on the Modified Incomplete Cholesky factorization (MIC). We employ two versions of MIC, with and without additional fill-ins using ‘by value’ as the fill-in strategy. By this technique one compares the values of the terms in the incomplete factor and chooses the largest ones to be included [13]. One of the most important parameters in both versions is the diagonal-scaling parameter α which insures positive definiteness of the incomplete factor. Its optimal value depends on the number of fill-ins. For a larger number of fill-ins the optimal value of the diagonal-scaling decreases. The optimal number of fill-ins is determined experimentally, whereas the value of the diagonal-scaling parameter is determined adaptively by incrementally increasing it until all positive pivots are obtained.

For optimal performance in terms of memory requirement, we also employ the SSOR based smoother, which does not require additional storage as opposed to MIC based methods.

5. Numerical examples

An obstacle test consisting of 20 industry and model problems (see Figs. 1 and 2) was designed to (i) determine the optimal values of computational parameters and to (ii) compare the two-level scheme with the existing state-of-the-art solvers including the Multifrontal solver with MMD reordering scheme, and the PCG solver with nearly optimal Modified Incomplete Cholesky factorization preconditioner. The following 20 problems comprise the obstacle test:

- (1) Automobile Body: MIN3 [14] shell and 2 node beam elements; 265 128 d.o.f.s.
- (2) High Speed Civil Transport (HSCT): MIN3 [14] shell elements; 88 422 d.o.f.s.
- (3) Wheel structure, 3 Node DKT + DMT shell [15], 320 358 d.o.f.s.
- (4) Wheel structure, 3 Node DKT + DMT shell [15], 138 462 d.o.f.s.
- (5) Casting Setup for Casting in Airfoil: 10 node tetrahedral elements; 158 166 d.o.f.s.
- (6) Turbine Blade with Platform: 10 node tetrahedral elements; 207 840 d.o.f.s.
- (7) Ring-Strut-Ring Structure: 4 node tetrahedral elements; 102 642 d.o.f.s.
- (8) Nozzle for Turbines: 10 node tetrahedral elements; 131 565 d.o.f.s.
- (9) Diffuser Casing with Gates for Casting: 10 node tetrahedral elements; 131 529 d.o.f.s.
- (10) Concrete canoe: 8 node ANS [16] shell elements; 132 486 d.o.f.s.
- (11) 3D weave unit cell: 4 node tetrahedral elements; 85 586 d.o.f.s.
- (12) Submarine frame: 3 Node DKT + DMT shell [15], 184 914 d.o.f.s.
- (13) SDRC [21] benchmark problem 1: 10 node tetrahedrals; 178 536 d.o.f.s.
- (14) SDRC [21] benchmark problem 2: 10 node tetrahedral; 145 911 d.o.f.s.
- (15) EMRC [20] benchmark problem 0: 3 node DKT + DMT shell [15]; 61 206 d.o.f.s.
- (16) EMRC [20] benchmark problem 2: 8 node hexahedrals; 27 783 d.o.f.s.
- (17) EMRC [20] benchmark problem 3: 8 node hexahedrals; 52 728 d.o.f.s.
- (18) EMRC [20] benchmark problem 5: 8 node hexahedrals; 52 728 d.o.f.s.
- (19) EMRC [20] benchmark problem 7: 20 node hexahedrals; 38 475 d.o.f.s.
- (20) EMRC [20] benchmark problem 8: 8 node ANS [16] shells; 66 246 d.o.f.s.

For construction of decision graphs in Section 5.1.2, additional problems have been generated by changing the values of the thickness of the shell.

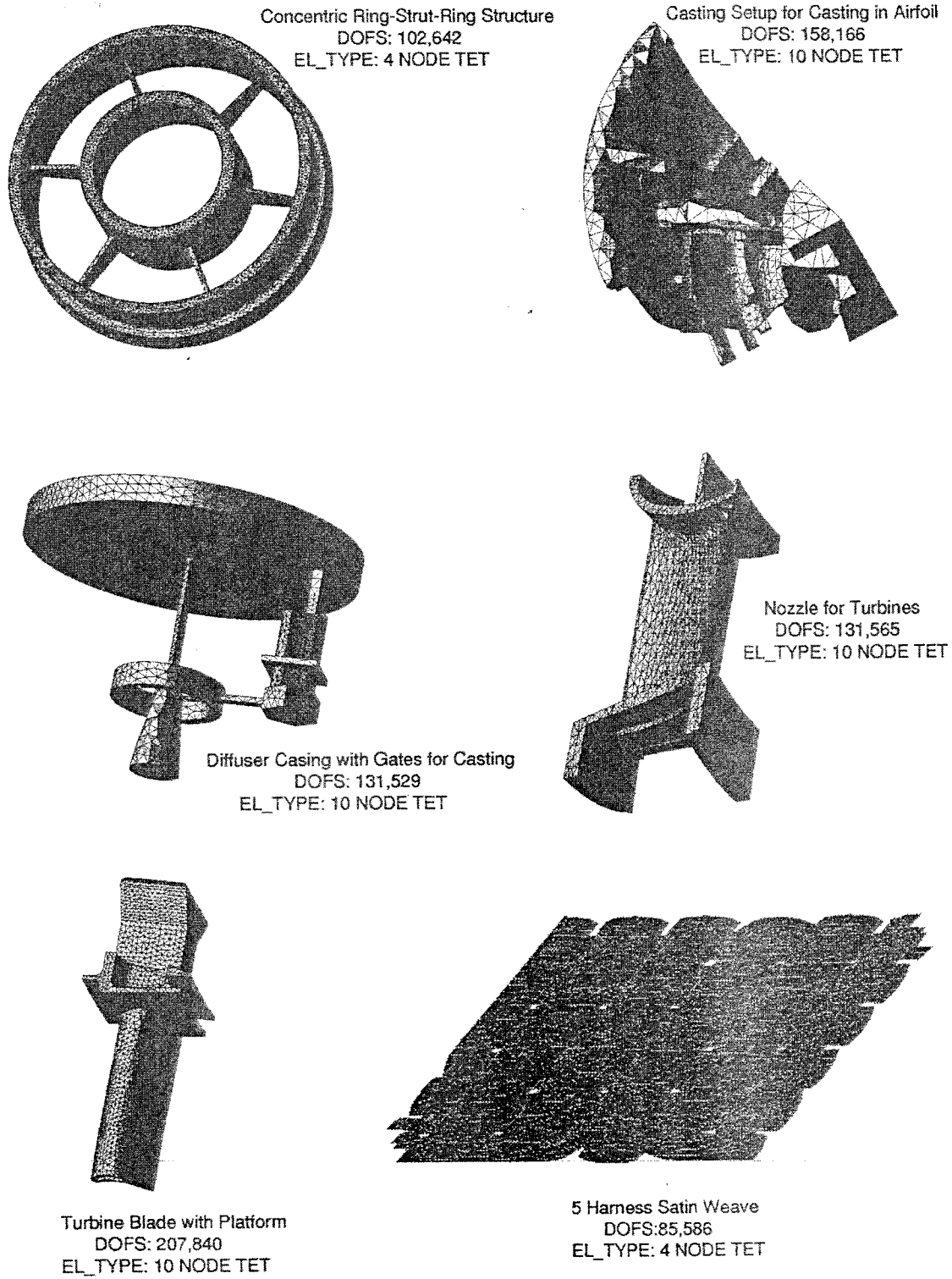
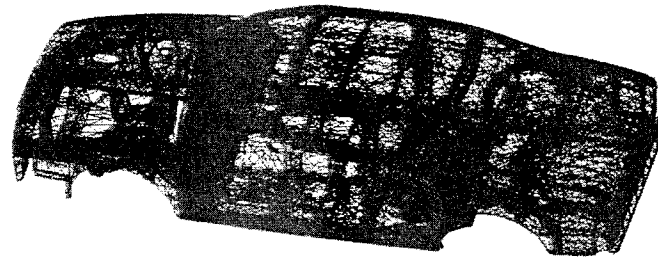
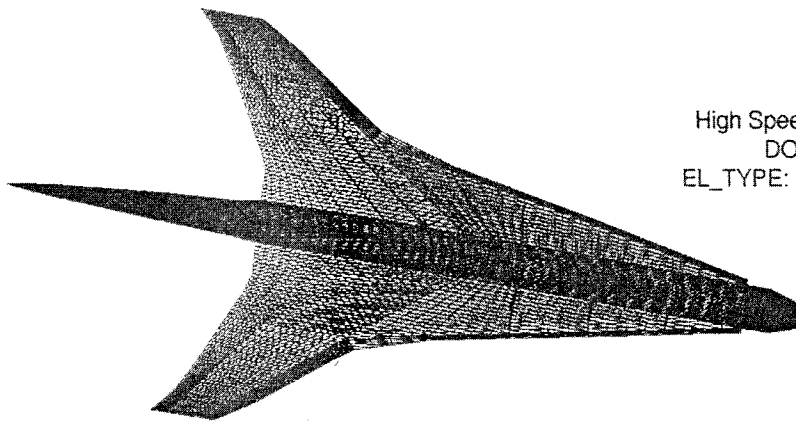


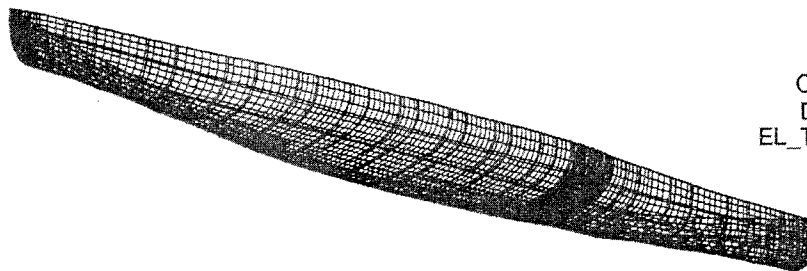
Fig. 1.



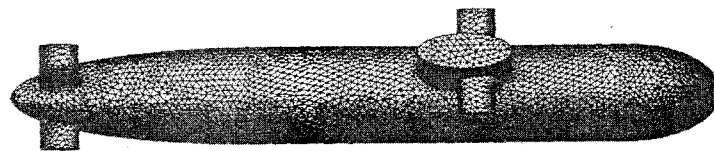
Automobile Body
DOFS: 265,128
EL_TYPES: MIN3S, BEAM



High Speed Civil Transport
DOFS: 88,422
EL_TYPE: MIN3S, DKT-DMT



Concrete Canoe
DOFS: 186,245
EL_TYPE: 8-node ANS



Submarine
DOFS: 174,221
EL_TYPE: DKT-DMT

Fig. 2.

5.1. Parametric study

In this section we present the results of numerical investigation of the following computation parameters: number of levels; limiting eigenvalue parameter γ for selection of the modes to be included in the prolongation

operator; number of fill-ins and diagonal scaling parameter α for Modified Incomplete Cholesky factorization; and coarsening parameter μ .

5.1.1. Selection of number of levels

To add the selection of optimal number of levels, we construct decision graphs in the space of condition number and sparsity for all the problems comprising the obstacle test. Decision graphs are scatter plots with superimposed discriminant functions, which classify the set of sample problems, each of which represents an optimal solver in terms of CPU time. Attention is restricted to one- (multifrontal solver with MMD reordering scheme) and two-level methods. For the two-level method considered, we utilize the optimal values of prolongation, smoothing and aggregation parameters calibrated in Sections 5.1.2–5.1.4.

Each problem for which the single level solver was faster has been marked by a circle in the space of sparsity and conditioning as shown in Fig. 3, whereas problems for which the two-level scheme has been found to be optimal in terms of CPU time were marked by a square. It can be seen that the space of the decision graph is subdivided into two regions with some minor overlap along the boundaries. The discriminant function is a straight line, except for some small problems considered ($<10\,000$ d.o.f.s.) where an overhead involved in constructing the multilevel preconditioner is relatively significant.

Another interesting observation is that for 80% of shell problems considered (shell problems marked by a dark circle or a square in Fig. 3), the one-level scheme has been found to be optimal, whereas for 97% of 3D problems (marked by an empty circle or a square in Fig. 3), the two-level scheme turned out to be faster.

Table 1 summarizes the cost and efficiency of predicting the condition number. It can be seen that the effectivity index, defined by the ratio of predicted and exact condition number, ranges between one-half to two. The cost of predicting the condition number ranges between 5 to 25% of total solution time. The cost of sparsity prediction, also shown in Table 1, was less than 5% of total solution time for all problems considered.

When a new problem is encountered, the optimal number of levels in terms of solution speed can be determined from the decision graph. Memory considerations play an important role in the solution strategy selection process. Sparsity and conditioning assessment as well as memory requirements are all estimated in the preprocessing stage (see Section 5.2 for memory requirements of various solution techniques). For all problems considered the direct in-core solver consumes two to eight times more memory than a two-level scheme with MIC smoother, and up to fifteen times more memory than the two-level scheme with SSOR smoother. Thus, if for a given problem a single level scheme is estimated to be the fastest, but may consume more memory than available (without swap), it is eliminated from the list of candidate strategies. Subsequently, the two-level scheme with MIC smoothing is tested for memory, and if not suitable, the two-level scheme with a weaker (SSOR) smoother is considered. If all three schemes fail to satisfy memory requirements, other possibilities,

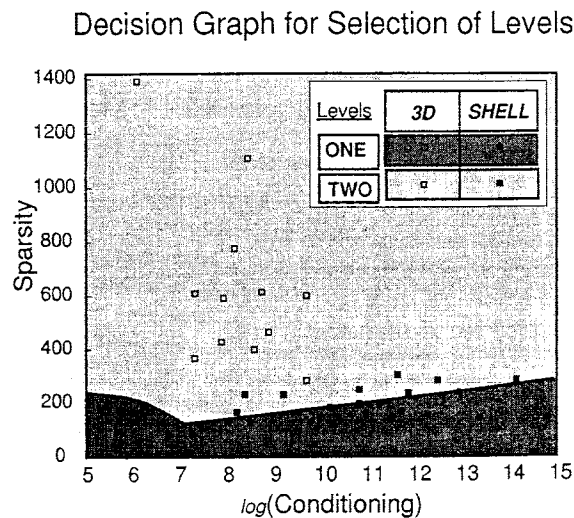


Fig. 3.

Table 1
Effectivity and efficiency of estimators

Problem	Elements	Equations	Effectivity index κ^{est}/κ^{ex}	Efficiency percentage of total CPU	
				Sparsity ω	Condition κ
CAR	MIN3	265128	0.613	3	9
HSCT-88k	MIN3	88422	1.227	3	10
Wheel 320k	DKT-DMT	320358	0.901	2	12
Wheel 138k	DKT-DMT	138462	0.923	2	9
Casting	Tetra-10	158166	0.926	2	9
Turbine Bld	Tetra-10	207840	NA	3	25
Ring-strut	Tetra-4	102462	0.975	4	15
Nozzle Tur	Tetra-10	131565	0.924	3	18
Diffuser	Tetra-4	75717	1.814	4	20
Canoe	ANS8	132486	0.908	2	5
Weave	Tetra-4	85586	1.218	4	25
Submarine	DKT-DMT	184914	0.828	2	8
SDRC1	Tetra-10	178536	0.854	4	22
SDRC2	Tetra-10	145911	2.000	5	25
EMRC0	DKT-DMT	61206	0.518	2	11
EMRC2	Hexa-8	27783	0.764	3	12
EMRC3	Hexa-8	52728	0.754	2	19
EMRC5	Hexa-8	52728	1.417	3	25
EMRC7	Hexa-20	38475	1.324	1	11
EMRC8	ANS8	66246	1.191	1	7

such as three or more level schemes, out-of-core direct and out-of-core iterative [19] solvers, should be considered. These alternatives have not been investigated in the present manuscript.

5.1.2. Prolongation parameters

In order to determine the optimal value of γ in terms of the CPU time, we have carried out extensive computational experiments for a wide range of problems, including well-posed and ill-posed cases. Surprisingly, it has been found that the optimal value of γ is independent of the problem condition and only slightly differs for different problems. Fig. 4 shows that for the HSCT problem the optimal value of γ , which minimizes the CPU time, was equal to 0.0035 independent of quality of MIC smoother (number of fill-ins and diagonal scaling parameter). On the other hand, for the Diffuser Casing problem the CPU time was practically independent of γ . For the Nozzle of Turbine problem significant reduction in terms of CPU time was observed for relatively large values of γ ranging from 0.0075 to 0.0100, whereas in the Automobile Body problem increasing the value of γ only slightly reduces the number of iterations but increases the CPU time. Based on these results we have built in $\gamma = 0.0050$ for further numerical studies and comparisons, which provides a reasonably good performance for all problems considered.

5.1.3. Smoothing parameters

The efficiency of the MIC based smoothing procedure highly depends on the two computational parameters: the number of fill-ins and the diagonal scaling. Typically, increasing the number of fill-ins allows one to decrease the value of the diagonal-scaling parameter. It can be seen (Fig. 4) that for Diffuser Casing and HSCT problems, the optimal value of fill-ins is in the range of 4–6, with a minimal value of diagonal-scaling parameter α which ensures positive pivots. For the Nozzle for Turbines and Automobile Body problems, it was observed that the number of fill-ins has a minor effect on the effectiveness of the iterative process. We did not consider a number of fill-ins greater than 8 due to increased in-core memory requirements.

Based on the computational experiments, the following strategy has been developed for determination of nearly optimal values of α and number of fill-ins:

- MIC with number of fill-ins is equal to 4.
- Initial diagonal-scaling parameter $\alpha = 0.01$.

CPU time as a function of coarse grid parameter γ and number of fill-ins.

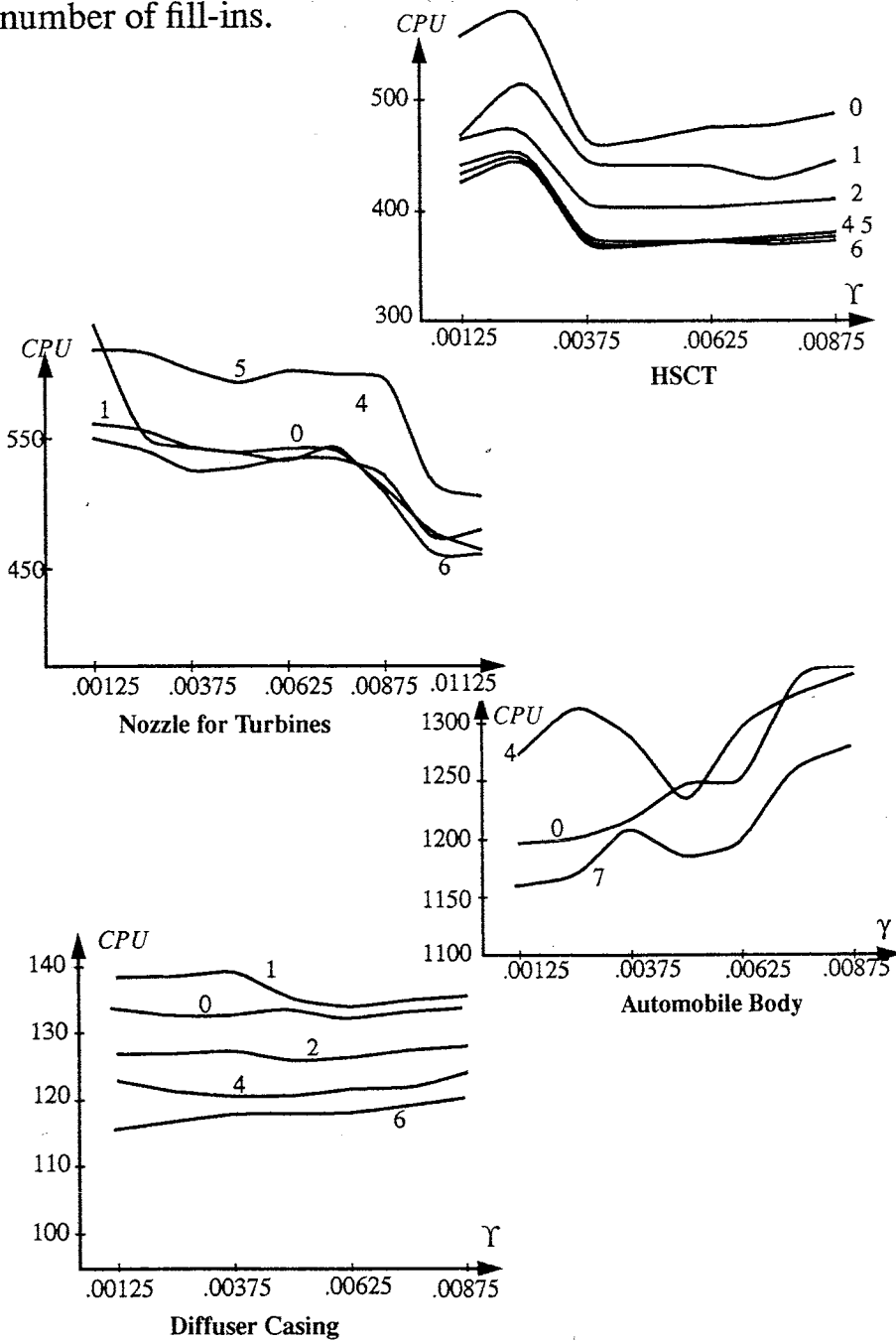


Fig. 4.

- Increasing α by the increment of $\Delta\alpha = 0.0025$ if non-positive pivot is encountered in the process of incomplete factorization, or if the two-level iteration procedure diverges.

5.1.4. Aggregation parameters

Numerical experiments in the obstacle test indicated that the value of the coarsening parameter μ had very little effect on the convergence of the iteration procedures. The only problem where considerable improvement

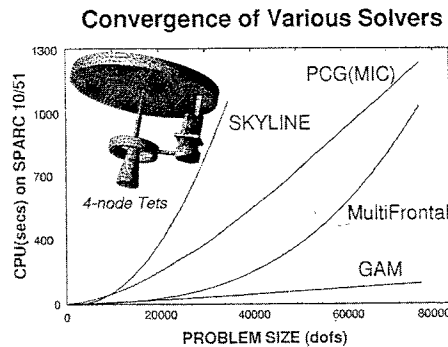


Fig. 5.

was observed was a 2-D problem for randomly distributed short fibers in matrix material, where the fiber/matrix stiffness ratio was equal to 100. The problem was modeled using quadrilateral finite elements. For this problem the converged solution was achieved in 23 iterations using the basic aggregation algorithm, while using the adaptive version of aggregation procedure with optimal value $\mu = 1.68$ the convergence was achieved in 18 iterations. In subsequent studies we employed the basic version of the aggregation procedure.

5.2. Comparison with other solvers and discussion

First, we compare the rate of convergence in terms of CPU time for the Diffuser Casing problem as obtained with the GAM solver, the Skyline Direct solver, the Multifrontal solver [17] and the PCG solver with Modified Incomplete Cholesky preconditioner. Fig. 5 shows that for the GAM solver the CPU time grows linearly with problem size as opposed to the other three solvers considered. It can be seen that even for relatively small problems with 35 000 d.o.f.s. the GAM outperforms the traditional Skyline solver by a factor of 27. For the problem with 70 000 d.o.f.s. the GAM solver outperforms the Multifrontal solver [17] by a factor of 9 and the PCG with Modified Incomplete Cholesky preconditioner by a factor of 12. Table 2 compares the performance of GAM solver in terms of the total CPU time and memory requirements with the Multifrontal solver [17] and the PCG solver with Modified Incomplete Cholesky preconditioner for all problems considered in the obstacle test.

In the second set of problems the GAM is compared with the 'smoothed aggregation' technique introduced in [7]. We have observed that for a 2-D model elasticity problem on a square domain, this approach gives an improvement in terms of number of iterations (16 instead of 23). However, for the ill-posed shell problem (HSCT) the number of iterations becomes more than twice larger (154) in comparison with the GAM solver. Furthermore, smoothing of the approximation field on each aggregate creates a denser prolongation operator, which in turn increases the CPU time of restriction and yields denser auxiliary matrix.

Table 3 contains split-up CPU times for (i) aggregation, (ii) restriction of stiffness matrix, (iii) factorization of auxiliary matrix, (iv) incomplete factorization of source matrix, and (v) iterative two-level procedure in all obstacle test problems. The aggregation time consists of eigenvalue analysis on each aggregate and construction of the prolongation operator. It can be seen that the cost of the aggregation procedure ranges from 5% (Automobile Body) to 30% (Ring-Strut), averaging 10% for all problems. The cost of iterative process ranges from 80% in poor conditioned problems (HSCT and Automobile Body) to less than 30% in well conditioned 3D problems, averaging 70% for all problems considered. It should be noted that the above split-up times can be easily modified by controlling the prolongation and smoothing parameters, but those considered above reflect a nearly optimal performance for problems (positive definite systems, single right-hand side vector) and computer architectures (single processor) considered.

6. Conclusions

An automated adaptive multilevel solver for linear (or linearized) systems of equations has been developed. The optimal number of levels is selected on the basis of estimated conditioning, sparsity of the factor and

Table 2
Comparison of solvers

Problem	Elements	Equations	Hip memory (MB)				CPU (s) [# of iterations]			
							SUN-ULTRA SPARC			
			Multifrontal	GAM1	GAM2	CG(IC)	Multifrontal	GAM1 ^a	GAM2 ^b	CG(IC)
CAR	MIN3	265128	512	255	154	235	587	1116 [60]	1848 [132]	22908 [2730]
HSCT-88k	MIN3	88422	207	99	70	73	197	450 [81]	1260 [277]	6033 [2277]
Wheel 320k	DKT-DMT	320358	862	310	206	241	2890	875 [38]	910 [115]	17752 [2083]
Wheel 138k	DKT-DMT	138462	288	119	78	101	427	276 [33]	474 [81]	3229 [914]
Casting	Tetra-10	158166	1012	192	132	177	5954	752 [49]	1775 [143]	12158 [1833]
Turbine Bld	Tetra-10	207840	>1500	280	235	269	NA	820 [25]	1567 [75]	4226 [437]
Ring-strut	Tetra-4	102642	337	79	64	73	618	126 [19]	176 [29]	1585 [681]
Nozzle Tur	Tetra-10	131565	996	167	103	149	5184	462 [26]	914 [73]	3579 [614]
Diffuser	Tetra-4	75717	327	67	55	58	753	116 [19]	120 [24]	1231 [660]
Canoe	ANS8	132486	522	167	91	153	672	497 [32]	1116 [101]	7727 [1183]
Weave	Tetra-4	85586	448	84	77	79	2619	148 [14]	153 [18]	654 [291]
Submarine	DKT-DMT	184914	370	158	101	129	746	408 [35]	497 [50]	6178 [1362]
SDRC1	Tetra-10	178536	1401	224	147	206	9446	498 [20]	946 [57]	1038 [128]
SDRC2	Tetra-10	145911	984	187	117	179	3184	342 [17]	466 [25]	4468 [679]
EMRC0	DKT-DMT	61206	122	62	47	44	78	104 [32]	327 [108]	1460 [985]
EMRC2	Hexa-8	27783	173	37	25	31	480	56 [20]	66 [41]	317 [273]
EMRC3	Hexa-8	52728	425	80	57	57	1912	161 [21]	215 [38]	779 [347]
EMRC5	Hexa-8	52728	424	70	46	56	1814	92 [11]	87 [13]	193 [82]
EMRC7	Hexa-20	38475	438	84	54	72	1863	137 [10]	169 [18]	287 [78]
EMRC8	ANS8	66246	226	88	61	82	227	172 [23]	600 [115]	3497 [1054]

^a GAM1: optimal CPU.^b GAM2: optimal memory.Table 3
GAM split-up times

Problem	Elements	Equations	CPU (s) SUN-ULTRA SPARC					
			Solver total	Aggregation	Restriction	Factorization	Incomplete	Iterative process
CAR	MIN3	265128	1116	65	48	37	91	875
HSCT-88k	MIN3	88422	450	18	20	34	18	360
Wheel 320k	DKT-DMT	320358	875	72	43	132	47	580
Wheel 138k	DKT-DMT	138462	288	24	18	23	18	204
Casting	Tetra-10	158166	752	65	122	17	49	500
Turbine Bld	Tetra-10	207840	820	93	209	76	74	368
Ring-strut	Tetra-4	102642	126	44	8	6	8	60
Nozzle Tur	Tetra-10	131565	462	54	114	17	46	231
Diffuser	Tetra-4	75717	116	36	7	10	7	56
Canoe	ANS8	132486	497	43	45	4	41	364
Weave	Tetra-4	85586	148	48	8	37	10	46
Submarine	DKT-DMT	184914	408	45	21	29	20	294
SDRC1	Tetra-10	178536	498	72	109	28	57	231
SDRC2	Tetra-10	145911	342	50	98	11	42	141
EMRC0	DKT-DMT	61206	104	7	10	11	6	70
EMRC2	Hexa-8	27783	56	5	6	8	7	31
EMRC3	Hexa-8	52728	161	14	11	55	15	66
EMRC5	Hexa-8	52728	92	9	8	29	13	32
EMRC7	Hexa-20	38475	137	17	33	16	31	40
EMRC8	ANS8	66246	172	15	36	6	21	95
		Average	376	40	49	29	31	227

available memory. The concept of aggregation introduced earlier in [5-7] has been generalized to automate the construction of nearly optimal auxiliary coarse models based on the information available in the source grid only. By this technique the approximation space over each aggregate is adaptively and automatically selected

based on the spectral characteristics of individual aggregates. Numerical experiments conducted on twenty industry and model problems affirm the solver's high efficiency and robustness characteristics.

The quest for the ultimate equation solver is still in its embryonic stage. Further research is essential to promote the proposed methodology from the status of 'interesting and having potential' to the general purpose equation solver. The three solver characteristics that need further development are:

- (i) *Generality*. The proposed framework needs to be extended to account for indefinite and nonsymmetric systems of equations, as well as problems with multiple right-hand sides.
- (ii) *Predictability*. Decision graphs and optimal solution strategies have to be developed for problems with multiple right-hand sides and multiple processor machines.
- (iii) *Memory efficiency*. So far only in-core solution methods have been considered. Clearly, an ultimate solution engine should have out-of-core capabilities, since it is not usually possible to keep the entire stiffness matrix in core.

References

- [1] A. George and J.W.H. Liu, The evolution of the minimum degree ordering algorithm, *SIAM Rev.* 31(1) (1989) 1–19.
- [2] O. Axelsson and V.A. Barker, *Finite Element Solution of Boundary Value Problems* (Academic Press, 1984).
- [3] A. Brandt, S.F. McCormick and J.W. Ruge, Algebraic multigrid (AMG) for sparse matrix equations, in: D.J. Evans, ed., *Sparsity and Its Applications* (Cambridge University Press, Cambridge, 1984).
- [4] W. Leontief, *The Structure of the American Economy 1919–1939* (Oxford U.P., NY, 1951).
- [5] V.E. Bulgakov, Multi-level iterative technique and aggregation concept with semi-analytical preconditioning for solving boundary-value problems, *Commun. Numer. Methods Engrg.* (1993) 649–657.
- [6] V.E. Bulgakov and G. Kuhn, High-performance multilevel iterative aggregation solver for large finite-element structural analysis problems, *Int. J. Numer. Methods Engrg.* 38 (1995) 3529–3544.
- [7] P. Vanek, Acceleration of convergence of a two-level algorithm by smoothing transfer operator, *Appl. Math.* 37 (1992) 265–274.
- [8] W. Hackbush and U. Trottenberg, *Multigrid Methods* (Springer-Verlag, Berlin, 1992).
- [9] I. Fried, Discretization and round-off errors in the finite element analysis of elliptic boundary value problems and eigenvalue problems, Ph.D. Thesis, MIT, 1971.
- [10] G.W. Stewart, *Introduction to Matrix Computations* (Academic Press, NY, 1973).
- [11] O. Axelsson, Analysis of incomplete matrix factorizations as multigrid smoothers for vector and parallel computers, *Appl. Math. Comput.* 19 (1986) 3–22.
- [12] H.D. Simon, The Lanczos algorithm with partial reorthogonalization, *Math. Comput.* (1984) 115–142.
- [13] M.A. Ajiz and A. Jennings, A robust incomplete Cholesky conjugate gradient algorithm, *Int. J. Numer. Methods Engrg.* 20 (1984) 949–966.
- [14] A. Tessler, A C^0 anisotropic three-node shallow shell element, *Comput. Methods Appl. Mech. Engrg.* 78 (1990) 89–103.
- [15] J. Fish and T. Belytschko, Stabilized rapidly convergent 18-degrees-of-freedom flat shell triangular element, *Int. J. Numer. Methods Engrg.* 33 (1992) 149–162.
- [16] K.C. Parks and G. Stanley, A curved C^0 shell element based on assumed natural coordinate strains, *J. Appl. Mech.* 108 (1986) 278–290.
- [17] O.O. Storaasli, VSS code, Computational Structures Branch, NASA Langley, 1996.
- [18] The Math Works, Inc., Cochituate Pl., 24 Prime Park Way, Natick, MA.
- [19] COSMOS finite element analysis code, Structural Research & Analysis Corp., 12121 Wishire Blvd., LA, CA.
- [20] EMRC, P.O. Box 696, Troy, Michigan 48099.
- [21] SDRC, 2000 Eastman Drive, Milford, Ohio 45150-2789.