
TOWARDS AN ULTIMATE FINITE ELEMENT ORIENTED SOLVER ON UNSTRUCTURED MESHES

Vladimir Belsky and Jacob Fish

Department of Civil Engineering

Rensselaer Polytechnic Institute

Troy, New-York

ABSTRACT

The paper investigates computational efficiency of various finite element solvers, including the state-of-the-art iterative methods based on multigrid-like and Modified Incomplete Cholesky preconditioners, as well as sparse direct solver recently developed at NASA Langley. These methods are compared to the newly developed Finite Element Oriented Solver (FEOS), which combines the advantages of the iterative and direct solution techniques. Numerical tests are conducted for both well-conditioned three dimensional problems as well as poor-conditioned problems, such as thin shells. The proposed FEOS solver has been found to possess a remarkable robustness and computational efficiency, by far superior to its comprising ingredients.

INTRODUCTION

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 dependent parameters. 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 close to singular systems). Direct solvers are ideal for solving small up to medium size problems (since the constant C for direct methods is significantly smaller than for iterative solvers), but becomes prohibitively expensive for large scale problems since the value of exponent for direct solvers is higher than for iterative methods. For large well conditioned three dimensional problems the storage and CPU time requirements for direct methods are so large that serious consideration of competing methods is a virtual reality.

Recent years saw a re-emergence of iterative solvers in finite element structural analysis due to increasing demand to analyze very large finite element systems. Nevertheless, the major obstacle that needs to be overcome before iterative solvers can be routinely used in commercial packages is circumventing their pathological sensitivity to problem conditioning, which is manifested by the

increase of constant C with degradation in problem conditioning. Moreover, for linear static analysis any type iterative method requires (except for the stiffness and preconditioner formation and factorization) for each load case a new iterative process (whereas in a direct solution, factorization is performed only once, and each load case requires only forward reduction and back substitution).

This paper presents a Finite Element Oriented Solver (FEOS), which exploits specific properties of the problem, including a finite element discretization and estimated problem conditioning in constructing a nearly optimal solution strategy. The FEOS is a hybrid solver with built in strategist that combines multigrid-like principles with efficient Incomplete Cholesky based smoothing techniques and state-of-the-art sparse direct methods for solving auxiliary preconditioned systems. Its major characteristics are summarized below:

- Computational efficiency and robustness:* FEOS is faster and requires less storage than the sparse direct solver for wide range of practical problems including those with distorted geometries, unstructured meshes, strong anisotropies - all giving rise to poor conditioned problems.

- Fully automated:* FEOS a priori selects an optimal solution strategy based on the estimated conditioning, problem size, number of load cases, etc.

Subsequent sections describe the building blocks of FEOS. Numerical experiments comparing the performance of FEOS with its basic constituents alone are given in the last section.

SPARSE DIRECT SOLVERS

Consider a sparse symmetric positive definite linear system

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

where A is the $n \times n$ symmetric and positive definite matrix; u and f are vectors of order n . Traditionally, the linear equation solvers employed have been envelope, band or frontal type. The common idea behind all these methods is that zeros outside the envelope of A are preserved in its Cholesky factor L . In order to reduce the storage requirements for the factor, the linear

system is reordered to reduce the envelope, band or front size. Among the effective envelope-reducing algorithms include the reverse Cuthill-McKee (George, 1981), Gibbs-Poole-Stockmeyer (Crane, 1976) and Gibbs-King (Lewis, 1982) ordering algorithms. Generally, zero entries within the envelope structure are not exploited as they are in true sparse solvers, and thus for large problems envelope-type methods can be much more demanding than true sparse solvers. Yet envelope-type methods have been very popular primarily due to their simplicity and ease of reordering algorithms.

A true sparse solver attempts to reduce an overall storage and arithmetic requirements by storing and computing only the logical nonzeros of the factor matrix. By this technique the linear system of equations is reordered to reduce the number of nonzeros in the factor matrix irrespective of any envelope structure. Indeed, effective ordering algorithms for true sparse solvers, such as Minimum Degree algorithm (George, 1987) generally scatter many of the nonzeros away from the diagonal and thus are entirely inappropriate for an envelope-type methods. Nevertheless, true sparse matrix methods have not gained wide acceptance among the software developers for large-scale applications primarily because general sparse orderings were difficult and time-consuming.

Our numerical experiments comparing envelope-type solvers with recently developed sparse direct solver at NASA Langley show that major shortcomings have been overcome, and that a good implementation of the sparse direct solver outperforms envelope-type solvers by orders of magnitude in both execution time and storage requirements.

MULTIGRID PRINCIPLES

Since the pioneering work of Fedorenko (1962), multigrid literature has grown in astonishing rate. This is not surprising since the multigrid-like methods possess the highest rate of convergence among the iterative techniques for solving symmetric positive definite linear systems. The principal idea of multigrid 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.

To clarify the basic principles we will denote the auxiliary grid functions 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.:

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

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

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

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

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

$$\frac{1}{2} (A(u^i + Qu_0^i), u^i + Qu_0^i) - (f, u^i + Qu_0^i) \Rightarrow \min u_0^i \in R^m \quad (5)$$

where (\dots) denotes the bilinear form defined by

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

A direct solution of the equation (5) yields a classical two-grid procedure. Alternatively, one may introduce an additional auxiliary grid for u_0 and so forth, leading to a natural multi-grid sequence. To fix ideas we will consider a two-grid process resulting from the direct minimization of (5) which yields

$$A_0 u_0^i = Q^T (f - Au^i) \quad (7)$$

where $A_0 = Q^T A Q$ is the restriction of the matrix A . The resulting classical two-grid algorithm can be viewed as a two-step procedure:

a) *Coarse grid correction*

$$\begin{aligned} r^i &= f - Au^i \\ u_0^i &= A_0^{-1} Q^T r^i \\ \tilde{u}^i &= u^i + Qu_0^i \end{aligned} \quad (8)$$

where \tilde{u}^i is a partial solution obtained after the coarse grid correction. Even though the auxiliary system of equations is much smaller than that of the source problem, its solution for large scale systems can be most efficiently obtained by means of sparse direct solver.

b) *Smoothing*

$$u^{i+1} = \tilde{u}^i + P^{-1} (f - A\tilde{u}^i) \quad (9)$$

where P is a smoothing preconditioner. For example, if the Jacobi method is employed for smoothing, then

$$P = \omega (\text{diag}(A)) \quad (10)$$

where ω is a weighting factor. A more efficient preconditioner of the form of Incomplete Cholesky factor is described in the next section.

To assess the rate of convergence we can associate the error vectors e^i, \tilde{e}^i defined by

$$e^i = u - u^i \quad \tilde{e}^i = u - \tilde{u}^i \quad (11)$$

where u is the exact solution of the source problem. Then the error resulting from the coarse grid correction (8) can be cast into the following form

$$\tilde{e}^i = (I - QA_0^{-1}Q^T A) e^i \quad (12)$$

where I is the identity $n \times n$ matrix. Combining equations (9), (11), the influence of smoothing on error reduction is given by:

$$e^{i+1} = (I - P^{-1}A) \tilde{e}^i \quad (13)$$

and from the equations (12), (13) the error vector of the two-grid

process with one post-smoothing iteration can be expressed as:

$$e^{i+1} = (I - P^{-1}A)(I - QA_0^{-1}Q^TA)e^i \quad (14)$$

Denoting

$$\begin{aligned} G &= I - P^{-1}A \\ T &= I - QA_0^{-1}Q^TA \end{aligned} \quad (15)$$

equation (14) can be rewritten in the following concise form

$$e^{i+1} = GTe^i \quad (16)$$

It is essential to note that T and $S = I - T$ are A -orthogonal projectors, namely:

$$(ATw, Sv) = 0 \quad \forall w, v \in R^n \quad (17)$$

$$\|v\|_A^2 = \|Tv\|_A^2 + \|Sv\|_A^2 \quad (18)$$

which yields that

$$\|T\|_A \leq 1 \quad (19)$$

Note that the projector T eliminates the effect of the prolongation operator, i.e.:

$$TQ = 0 \quad (20)$$

The rate of convergence of the two-grid method in heterogeneous media for one-dimensional problems has been assessed by Fish and Belsky (1994), and has been shown to be governed by a factor $q^2/(4 - q^2)$, where

$$q = \left(\sqrt{d_1 d_2} \right) / \frac{1}{2} (d_1 + d_2) \quad (21)$$

and d_1 represent the stiffnesses of the microconstituents. Note that if the media is homogeneous and the mesh is uniform, then $d_1 = d_2$ and one recovers the classical multigrid estimate, which states that asymptotically the error reduces by a factor of three with each new multigrid cycle. On the other hand if one phase is significantly stiffer than the other, i.e. $d_1 \gg d_2$, then the multigrid method converges in a single cycle or very few cycles at most.

For poor conditioned problems it is necessary to accelerate the rate of convergence of the multigrid method. Using (14) the reduction of error in a single cycle $s^i = e^{i+1} - e^i$ can be expressed as linear function of the residual, i.e., $s^i = M^{-1}r^i$, where M is termed as multigrid preconditioner. Various forms of the second-order acceleration schemes can be expressed as follows

$$u^{k+1} = \beta^k u^k + (1 - \beta^k) u^{k-1} + \beta^k \alpha^k M^{-1} r^k \quad (22)$$

where acceleration parameters α, β are found by either (i) minimizing the energy functional or L_2 norm of residuals, or by (ii) conjugate gradient method, which imposes orthogonality condition of the form

$$(M^{-1}AM^{-1}r^{k+1}, r^j) = 0 \quad \forall j \leq k \quad (23)$$

The major drawback of the multigrid method for general unstructured meshes is the fact that it requires construction of mesh hierarchy in the solution process. This linkage seems to be undesirable for general purpose FE codes, unless it is utilized in

the adaptive context, where the sequence of meshes generated by the process of adaptivity is exploited in the solution process. Algebraic multigrid (Ruge, 1987), on the hand, does not require formulation of continuous problem, which corresponds to the given algebraic system equations, and no grids are involved, but as a result of that the efficiency suffers. Instead, FEOS automatically constructs auxiliary mesh hierarchies from the source grid by recursively simplifying the kinematics of the source grid.

INCOMPLETE CHOLESKY PRECONDITIONERS

Perhaps one of the most efficient smoothers for multigrid-like methods is based on Incomplete Cholesky Factorization. By this technique an approximate factorization of the stiffness matrix is introduced without generating too many fill-ins. Such an approach leads to the factorization of the type $LDL^T = A - E$, where E is an error matrix which is not explicitly formed. For this class of methods the error is introduced by either prescribing the position of elements to be rejected (Manteuffel, 1980) or by discarding those elements in the factor which are smaller than specified tolerance (Axelsson, 1983). This rejection process often leads to an unstable factorization that may result in a nonpositive definite preconditioner. Several remedies have been proposed including modification of factorization by making it more diagonally dominant (Manteuffel, 1980) or by correcting diagonal elements in the factorization process (Axelsson, 1983).

Our experience with multigrid-like methods suggests that the simplest version of Incomplete Cholesky Factorization, which preserves the sparsity pattern of the source stiffness matrix and ensures its stability by means of diagonal scaling, is the most suitable smoothing procedure. Heavier MIC-based smoothers, which allow partial fill-ins suffer from duplication of computational effort in the sense that they are acting on the same lower frequency modes of error which can be adequately captured on the auxiliary coarse mesh

PERFORMANCE STUDIES

The performance comparisons between the FEOS, PCG with Modified Incomplete Cholesky (MIC) preconditioner and NASA Langley sparse direct solvers were carried out on the SPARC 10 workstation. The numerical results are summarized in Table 1. The convergence criterion for the FEOS and PCG solvers was selected as $1.0e-8$ in the relative residual norm. Note that the sparse direct solver outperformed the envelope-type solver by orders of magnitude for the problems considered, and therefore the envelope-type solver's results were not included in the Table. Our numerical experimentation agenda included three 3D solid mechanics problems (intersection of two cylinders - model 1, inclusion problem - model 2, bracket problem - model 3) and two cylindrical shell problems with thickness/span ratio of 1/100 for model 4 and 1/300 for model 5. The finite element mesh for the Model 3 is presented in Figure 1.