Global Basis Two-Level Method For Indefinite Systems. Part 2: Computational Issues

Yong Qu and Jacob Fish
Department of Civil, Mechanical and Aerospace Engineering
Rensselaer Polytechnic Institute, Troy, NY 12180

ABSTRACT

Algorithmic aspects and computational efficiency of the global basis two-level method are investigated in the context of symmetric indefinite system of equations. The algorithm includes efficient construction of the global basis prolongator using Lanczos vectors, predictor-corrector smoothing procedures, and a heuristic two-level feedback loop aimed at ensuring convergence. Numerical experiments consisting of 3D Helmholtz equations and shear banding problems with strain softening demonstrate the excellent performance of the method.

Keywords: multilevel, multigrid, indefinite, convergence, global basis

1.0 Introduction

In Part 1 [1] we studied the effect of deviation (or error) from the optimal prolongation operator spanned by the spectrum of highest eigenmodes of the smoothing iteration matrix R referred to as algebraically "smooth" modes. We have shown that for highly indefinite systems any deviation from the optimality has a detrimental effect on the rate of convergence. This suggests that a successful two-level method should possess a coarse model able to accurately reproduce the space of algebraically "smooth" modes. The smoother in turn should be engineered to minimize the spectral radius of the smoothing iteration matrix R. In [1] we have introduced the concept of optimal prolongation operator (or equivalently the optimal coarse model), and shown that for highly indefinite systems such prolongator is (i) nonsmooth in a geometric sense, (ii) highly sensitive to any deviation from optimality, and (iii) nonlocal.

Based on the studies conducted in [1], Part 2 of the manuscript focuses on the algorithmic aspects including: efficient construction of the global basis prolongation operator using Lanczos vectors, predictor-corrector smoothing procedures, and a heuristic two-level feedback loop aimed at ensuring convergence of the global-basis two level method.

Part 2 of the manuscripts is organized as follows: In Section 2.1, we describe an algorithm designated at constructing a nearly optimal prolongation operator and a corresponding coarse model matrix. Section 2.2 presents a new incomplete factorization with threshold which serves as an efficient single-level smoother for indefinite systems. Section 2.3 discusses the possibility of using multilevel preconditioners, such as the aggregation based method [2], as the smoothers within the global basis two-level method. The overall heuristic feedback loop is described in Section 2.4. Numerical experiments on a sequence of examples involving Helmholtz equation on bounded domain and shear banding problem with strain softening conclude the manuscript.

2.0 Elements of global basis two-level method

2.1 Global basis prolongation operator

In [1] we have shown that for highly indefinite systems the prolongation operator has a pathological sensitivity to any deviation from the optimal value defined by a linear combination of higher spectrum of eigenvalues of the smoothing iteration matrix

$$R = I_N - M^{-1}K \in \Re^{N \times N} \tag{1}$$

where K and M are the stiffness and smoothing preconditioner, respectively, both $N \times N$ symmetric sparse indefinite matrices. To construct an optimal subspace for prolongation operator we employ the Lanczos method for nonsymmetric systems. The Lanczos recursion has the property of first converging to the eigenvalues with largest magnitude in the spectrum [8]. The original form of the Lanczos method for unsymmetric matrices is susceptible to breakdowns caused by division by zero or near-zero. This problem has been circumvented by the so-called look-ahead technique [3]. Possible breakdowns are prevented by relaxing the standard biorthogonality condition in the classical Lanczos algorithm to block the biorthogonality condition whenever the exact or near breakdown would happen. The Lanczos algorithm for unsymmetric smoothing iteration matrix, R, has been further simplified by exploiting the symmetry of K and M matrices [3]. The principles of the look-ahead Lanczos method as well as its utilization for coarse model construction are briefly summarized below.

2.1.1 Lanczos algorithm with look-ahead

Given nonzero starting vector, v_1, the simplified look-ahead Lanczos method generates a sequence of vectors $v_1, v_2, \ldots, v_n$, $i = 1,2,\ldots,n$. The three-term recursions of the Lanczos method can be written as

$$RV_n = V_n H_n + [0 \ldots 0 \ldots v_{n+1}\rho_{n+1}] \tag{2}$$

where $\rho_{n+1} = \|v_{n+1}\|$, $V_n = [v_1, v_2, \ldots, v_n]$ and H_n is a $n \times n$ block tridiagonal matrix.

For the simplified Lanczos method [3], the biothogonality condition for general case reduces to

$$v_n^T K v_i = 0 \qquad \forall i = 1, 2, \ldots, n_{n(l)} - 1 \tag{3}$$

where $n(l)$ monitors the number of look-ahead steps; $n = n_j$ and $n = n_{j+1} - 1$ mark the start and the end of the jth look-ahead step.

The eigenvalues of the block tridiagonal matrix H_n serve as an approximation of the eigenvalues of R. For $v_{n+1}\rho_{n+1} = 0$ all eigenvalues of H_n are also the eigenvalues of R. QR method [3] can be used to compute the eigenpairs of H_n from

$$H_n \Psi = \Psi \tilde{\Lambda} \tag{4}$$

where $\Psi = [\Psi_1 \Psi_2 \ldots \Psi_n] \in C^{n \times n}$ and $\tilde{\Lambda} = diag(\tilde{\lambda}_1 \tilde{\lambda}_2 \ldots \tilde{\lambda}_n) \in C^{n \times n}$.

The approximation, $V_n \Psi$, of the eigenvectors of *R*, i.e., $R V_n \Psi \approx V_n \Psi \tilde{\Lambda}$, is utilized for coarse model construction in Section 2.1.2.

2.1.2 Approximation of the global-basis prolongation

Since the unsymmetric Lanczos algorithm with look-ahead is used to construct the algebraically "smooth" modes, Lanczos vectors themselves are viable candidates for the prolongation operator since they approximate the same subspace as the eigenvectors, i.e.

$$V_n \Psi = \Phi \tag{5}$$

Suppose that the Lanczos recursions has found exactly the highest *n* eigenpairs of *R.* Then if the prolongation operator is defined as

$$Q = V_n \tag{6}$$

the corresponding two level iteration matrix, *L*, coincides with the two-level iteration matrix, $\tilde{L}$, with optimal prolongation operator defined by $\tilde{Q} = V_n \Psi_n$. The proof follows from

$$\begin{aligned} \tilde{L} &= R^{\nu}(I_n - \tilde{Q}\tilde{K}_0^{-1}\tilde{Q}^T K)R^{\nu} \\ &= R^{\nu}(I_n - Q\Psi_n \Psi_n^{-1} K_0^{-1} \Psi_n^{-T} \Psi_n^T Q^T K)R^{\nu} \\ &= R^{\nu}(I_n - QK_0^{-1}Q^T K)R^{\nu} = L \end{aligned} \tag{7}$$

2.1.3 Coarse model matrix computation

If Lanczos vectors in (6) were computed exactly, the coarse model matrix, $K_0 = Q^T K Q$, would have been block diagonal due to block biothogonality condition (3). The size of the block would have been the same as that of the corresponding look-ahead step. However, due to loss of orthogonality of the Lanczos vectors in finite precision computations [8] the relationship between eigenpairs of *R* and those of the block tri-diagonal matrix H_n is not trivial. Loss of orthogonality can be also viewed as a deviation from the optimal prolongation, which may deteriorate the performance of the two-level method. Moreover, increasing the number of Lanczos vectors in the prolongation operator without reorthogonalization may cause near singularity of the coarse model matrix due to nearly linear dependence of Lanczos vectors. If we were to allow for some loss of orthogonality, we would then need to compute the off diagonal terms of the coarse model matrix and to factorize it. This additional computational cost is comparable to that of complete reorthogonalization, and thus complete reorthogonalization seems to be a better choice. From the computational efficiency standpoint we bound the num-

ber of Lanczos vectors selected by $C\sqrt[4]{N}$ where N is the size of the source matrix and C is a constant (typically in the range from one to ten). This choice is selected to ensure that the Lanczos process with full reorthogonalization does not dominate the entire computational cost. If the desired size of the coarse model is too small to resolve all the algebraically "smooth" modes, it is advantageous to improve the quality of the smoother rather than to increase the number of Lanczos vectors.

2.2 Single-level smoother

In this section, we develop an Incomplete LDL^T predictor-corrector based smoother for indefinite systems which is a modification of the row-based Crout-Doolittle factorization for symmetric matrices with a dropping rule based on the numerical value of the fill-ins introduced [4][5].

For indefinite systems, the main difficulties arise due to inaccuracies caused by dropping, unstable triangular solve and breakdown caused by zero pivots [6]. In attempt to construct a robust smoother we study several remedies including scaling, reordering, diagonal pertubation, pivoting, use of mixture of 1×1 and 2×2 pivot blocks and selection of appropriate dropping strategies.

2.2.1 Scaling

The objective of scaling is to improve conditioning of the original matrix. For incomplete factorization by threshold, scaling also allows to use the same drop tolerance for all rows. For positive definite systems positive diagonal entries of the source matrix are typically used for scaling. For indefinite systems, on the other hand, we propose an iterative approach (see *Algorithm 1*) designated to compute the diagonal scaling matrix P which makes the ∞-norm of each row of the scaled system $P^{-1}KP$ to be bounded by $[1-\mu, 1+\mu]$ where μ is a small positive number. The iterative process is required to preserve the symmetry of the preconditioner.

Algorithm 1: Scaling of Indefinite Systems

1. *Set* $P = I_n$, *flag = 0.*

2. *Do while flag = 0*

3. *flag = 1; T =* I_n

4. *Loop over all nonzeros in matrix K*

5. *For each nonzero* K_{ij}

6. *if* $\left|K_{ij}/P_{ii}/P_{jj} - 1\right| > \mu$, *then flag = 0.*

7. *if* $\sqrt{|K_{ij}/P_{ii}/P_{jj}|} > T_{ii}$ *, then* $T_{ii} = \sqrt{|K_{ij}/P_{ii}/P_{jj}|}$

8. *if* $\sqrt{|K_{ij}/P_{ii}/P_{jj}|} > T_{jj}$ *, then* $T_{jj} = \sqrt{|K_{ij}/P_{ii}/P_{jj}|}$

9. *End of loop*

10. *P=PT*

11. *Enddo*

2.2.2 Reordering

The effects of reordering on the performance of the smoother have been studied by many researchers [10]. For positive definite systems it has been observed that the Reverse Cuthill-McKee and Minimum Degree, have little effect on convergence. Based on our numerical experiments, we have observed that indefinite systems are highly sensitive to reordering. Reverse Cuthill-McKee reordering scheme seems to be superior for weakly indefinite systems since it produces less nonzeros than other reorderings. Minimum Degree reordering scheme has been found to be a better choice for highly indefinite systems, which require high amount of fill-ins. Similar observations were reported in [10] for nonsymmetric systems.

2.2.3 Diagonal pertubation and block diagonal pivoting

The simplest method to avoid small pivots is to replace them by larger values to ensure they do not create very large off-diagonal entries. For indefinite systems, this stabilization procedure, or so-called diagonal pertubation, can be carried out by using a different shift for each row prior to factorization or by computing the shifts during the factorization. The sign of the shift should be the same as that of the pivot. The disadvantages of the diagonal pertubation are inaccuracy caused by the shift and a lack of rigorous approach in determining the value of the shift.

An alternative is to use pivoting. For the symmetric matrix, we must use symmetric pivoting to maintain symmetry. Thus like the Bunch-Kaufman factorization and its variants [7][9] the diagonal matrix is replaced by the block diagonal matrix, composed of 1×1 and 2×2 blocks. To incorporate pivoting in symmetric incomplete factorization, both the row and column data structures are required. Figure 1 illustrates the data structures for symmetric pivoting. In the implementation of symmetric pivoting, no actual row or column exchanges are made. The new indices are determined by the permutation vectors which are updated with each symmetric pivoting. We store diagonal matrix, *D*, and upper triangular matrix, L^T, in the original configuration and use the row headers, column headers and the permutation arrays to keep track of the current configuration.

For sparse matrices, the pivoting scheme must improve stability while minimizing the impact of pivoting on sparsity [6][9]. *Algorithm 2* is designated to find sucha balance.

.

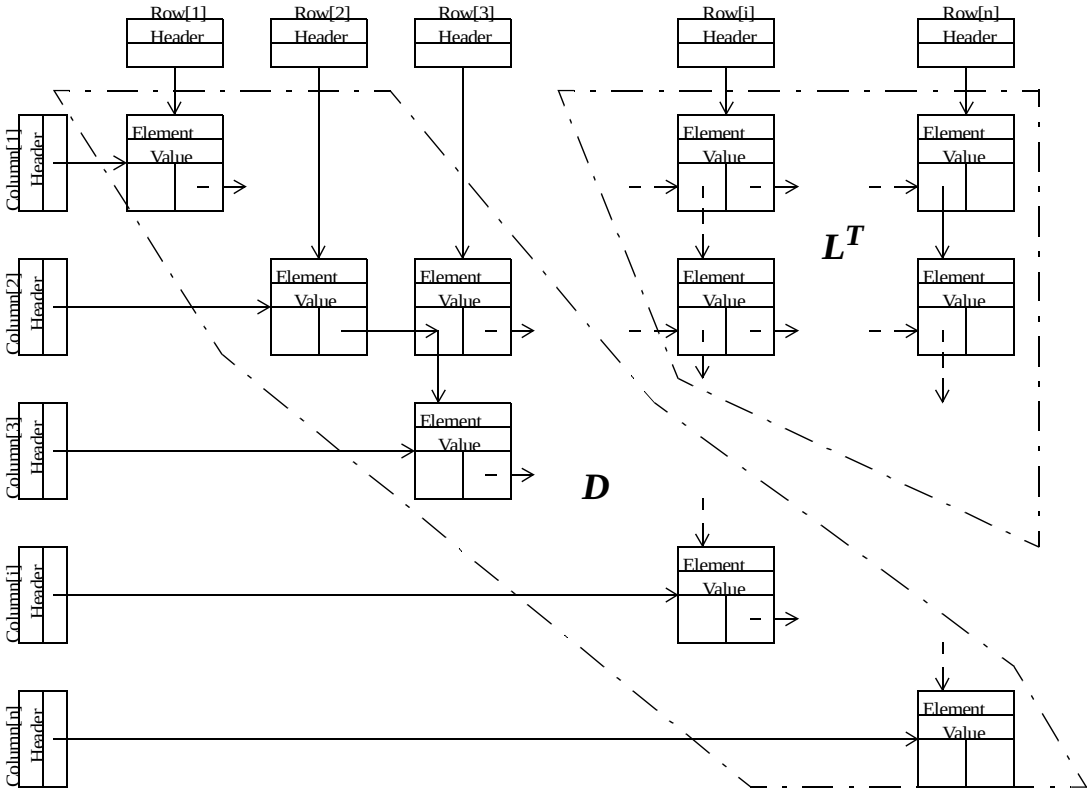

FIGURE 1. : Data structures for Modified Incomplete LDL^T Factorization by Threshold

Algorithm 2: Pivot Strategy

After the computation of dense row w for ith row,

1. *Find maximum nonzero* w_p *of dense row w;*

2. *If* $|D_{ii}| \geq w_p \times \hat{\alpha}$,

3. *use* D_{ii} *as* 1×1 *pivot*

else

4. *loop over all nonzeros of w*

5. *Find such* 2×2 *block* $E = \begin{bmatrix} D_{ii} & w_q \\ w_q & D_{qq} \end{bmatrix}$ *which has the maximum 2-norm and also satisfies*

$$Height(q) \leq Height(i+1) + m$$

*where Height(*) is the maximum column index of nonzeros in row *;*

m is a small positive integer.

6. *end of loop*

7. *If such E has been found and* $\|E\|_2 \geq \hat{\alpha} \times |D_{ii}|$

8. *use E as* 2×2 *pivot*

else

9. *use* $sign(D_{ii}) \times \hat{a} \times w_p$ *as* 1×1 *pivot*

10. *Endif*

11. *Endif*

In the above algorithm, $\hat{\alpha}$ and *m* are user-defined parameters controlling the number of 2×2 pivot blocks and the sparsity pattern of the incomplete factorization. In the incomplete factorization, the growth of elements of diagonal matrix, *D*, and upper triangular matrix, L^T is not as critical as in the complete factorization because each element is updated fewer times than in the complete factorization. Therefore, we can use significantly smaller values of $\hat{\alpha}$ to reduce the number of 2×2 pivot blocks and to preserve sparsity. The value of the parameter *m* is also used to control sparsity. Although we may expect to use large values of *m* for problems with structurally zero pivot, the value *m* should be generally small.

2.2.4 Selecting appropriate dropping schemes

The proposed algorithm is based on two dropping strategies:

(i) The largest *lfil* nonzeros in the dense vector are retained;

(ii) The nonzeros larger than the threshold tolerance *droptol* are retained in the dense vector.

The former is used to construct the *tentative* (or initial) *incomplete factorization* while the latter is employed to construct the *enhanced incomplete factorization* (details are described in Section 2.3). For the numerical dropping schemes, it is vital to apply the same dropping criterion for all matrix rows. Using the data structure described in Figure 1 it is possible to allocate the memory for incomplete factors dynamically during factorization. Moreover, we have found that changing the sequence of updating the remaining diagonal elements and dropping the small element of dense vectors in the incomplete factorization will significantly improve the performance of the smoother. Our experience indicates that updating the diagonal entries with dropped elements [5][6] makes the incomplete factorization less accurate, especially for highly indefinite systems.

Algorithm 3 described below incorporates the aforementioned features into the incomplete factorization originally developed in [6]. The data structures shown in Figure 1 are used to store D and L^T.

Algorithm 3: Modified Incomplete LDL^T Factorization by Threshold

1.Select reordering scheme and reorder matrix K

If droptol > 0.0001

use Reverse Cuthill-McKcee reordering

else

use Minimal Degree reordering

Endif

2.Use Algorithm 1 to scale the matrix K.

3.Load diagonal of scaled matrix into D

4.Set i = 1

5.*While* $i \le n$ *, Do*

6. *Load row i of scaled matrix K into vector w*

7. *For j = 1,...,i-1 Do:*

Case 1: 1×1 *block* $[D_{jj}]$

If $L^T_{ji} \neq 0$

Update vector w

if $p > i$ *and* $L^T_{jp} \neq 0$ *then* $w_p = w_p - L^T_{jp} \times L^T_{ji} \times D_{jj}$

Endif

Case 2: 2×2 *block* $\begin{bmatrix} D_{jj} & D_{jj+1} \\ D_{jj+1} & D_{j+1j+1} \end{bmatrix}$

If $L^T_{ji} \neq 0$ *or* $L^T_{j+1i} \neq 0$

Update vector w

$$if\ \ p > i\ \ and\ \ L^T_{jp} \neq 0\ \ or\ \ L^T_{j+1p} \neq 0$$

$$w_p = w_p - D_{jj} \times L^T_{jp} \times L^T_{ji} - D_{jj+1} \times L^T_{jp+1} \times L^T_{ji} - D_{jj+1} \times L^T_{jp} \times L^T_{j+1i} - D_{j+1j+1} \times L^T_{j+1p} \times L^T_{j+1i}$$

Endif

Endif

EndDo

8. *Apply the dropping rule to w*

Scheme (i): The largest lfil nonzeros in w are kept.

Scheme (ii): $If(|w_p| \leq droptol), w_p = 0$

9. *Use Algorithm 2 to select the pivot scheme*

Case 1: 1×1 *block* $[D_{ii}]$

Update remaining diagonal elements

$$D_{pp} = D_{pp} - w_p \times w_p / D_{ii} \quad \forall (w_p \neq 0)$$

$$w_p = w_p / D_{ii} \qquad \forall (w_p \neq 0)$$

Copy w to L^T

$i = i + 1$

Case 2: 2×2 *block* $\begin{bmatrix} D_{ii} & D_{iq} \\ D_{iq} & D_{qq} \end{bmatrix}$

Permute i+1th row(column) with qth row(column) of K

$$D_{ii+1} = D_{iq}, D_{i+1i+1} = D_{qq}$$

Load row i+1 of matrix K into vector v

Update v using all previous rows except row i.(Same as step 6)

Apply same dropping rule to v

Compute the inverse of 2×2 *block*

$$E = \begin{bmatrix} E_{11} & E_{12} \\ E_{12} & E_{22} \end{bmatrix} = \begin{bmatrix} D_{ii} & D_{ii+1} \\ D_{ii+1} & D_{i+1i+1} \end{bmatrix}^{-1}$$

Update remaining diagonal elements

$$if\ p > i \ \ and \ \ w_p \neq 0 \ \ or \ \ v_p \neq 0$$

$$D_{pp} = D_{pp} - E_{11} \times w_p \times w_p - 2E_{12} \times w_p \times v_p - E_{22} \times v_p \times v_p$$

Compute

$$\begin{bmatrix} w \\ v \end{bmatrix} = E \times \begin{bmatrix} w \\ v \end{bmatrix}$$

Copy w to L^T

$i = i + 1$

Copy v to L^T

$i = i + 1$

10. *EndDo*

11.*Back Scale D and* L^T

2.3 Multilevel methods as smoothers

Convergence studies conducted in [1] are valid for any smoothing preconditioner, *M*. The effectivity of the smoother is measured by the spectral radius of the smoothing iteration matrix *R*. In other words, any preconditioner, including single-level, multilevel, domain decomposition and element-by-element method, could be in principle used as the smoother within the framework of the global basis two-level method. The coarse model in turn serves as a buffer zone against divergence aimed at capturing the modes which have not been smoothed out by the smoother of choice.

In Section 3 we focus on extending the application of the global basis two-level method to multilevel smoothers. In particular, we consider the Generalized Aggregation based preconditioners (GAM) developed in [2] as smoothers within the framework of the global basis two level method.

2.4 Some thoughts on adaptive global-basis algorithm

For indefinite systems, it is not feasible to *a prior* estimate the optimal algorithmic parameters, such as the optimal value of dropping parameters for the incomplete factorization, the optimal number of aggregates, the optimal size of the aggregates and the optimal number of local modes selected on each aggregate [2]. For a certain choice of parameters, the quality of smoother can be estimated on the basis of the magnitude of the maximum eigenvalue of R. Here we provide some initial thoughts on constructing a heuristic algorithm for adaptive coarse model construction.

The algorithm starts by constructing a *tentative smoother.* In the case of a single-level preconditioner, the *tentative smoother* employs *Algorithm 3* with dropping scheme (i) in attempt to estimate the maximum eigenvalue of the smoothing iteration matrix. The maximum nonzero *lfil* for row *i* is set equal to the number of nonzeros of L^T part of row *i* of matrix *K*. The original sparse pattern is ignored and only the *lfil* nonzeros with the largest magnitude are kept. This *tentative* incomplete factorization is computationally inexpensive and needs the same memory as the original matrix. In the case the GAM preconditioner [2] is employed as a smoother, the *tentative GAM preconditioner* would employ the same zero-fill-in incomplete factorization as a relaxation scheme within GAM preconditioner. The *tentative* local coarse model of the *GAM* preconditioner is then formed by selecting very few local eigenmodes from one-neighbor aggregates with zero Neumann boundary conditions [2].

If the smoother is found to be satisfactory (see Section 3 for the criterion), the coarse model is formed to include the algebraically "smooth" modes. If the number of required "algebraically" smooth modes exceeds the user prescribed limit ($C\sqrt[4]{N}$), the quality of the smoother is improved.

On the other hand, if the *tentative incomplete factorization* for the single-level smoother is found to be not satisfactory, i.e., a very large coarse model is required, the new tolerance setting is heuristically (see Section 3) determined on the basis of the magnitude of the maximum eigenvalue. The *enhanced incomplete factorization* is carried out using *Algorithm 3* with the new tolerance setting for dropping scheme (ii). The quality of the enhanced smoother is estimated using Lanczos method. If it remains unsatisfactory, the value of dropping tolerance is further reduced (see Section 3 for the details). This process continues until the satisfactory smoother is found. The Lanczos vectors from the previous incomplete factorizations are utilized for the prolongation operator construction.

If the *tentative GAM preconditioner* for the GAM smoother is found to be not satisfactory, we may either enhance the relaxation scheme by using *Algorithm 3* with the new tolerance setting for dropping scheme (ii), or construct an *enhanced GAM coarse model* by increasing the value number of modes on each aggregate or taking larger aggregates. Schematics of the adaptive global basis algorithm are illustrated in Figure 2.

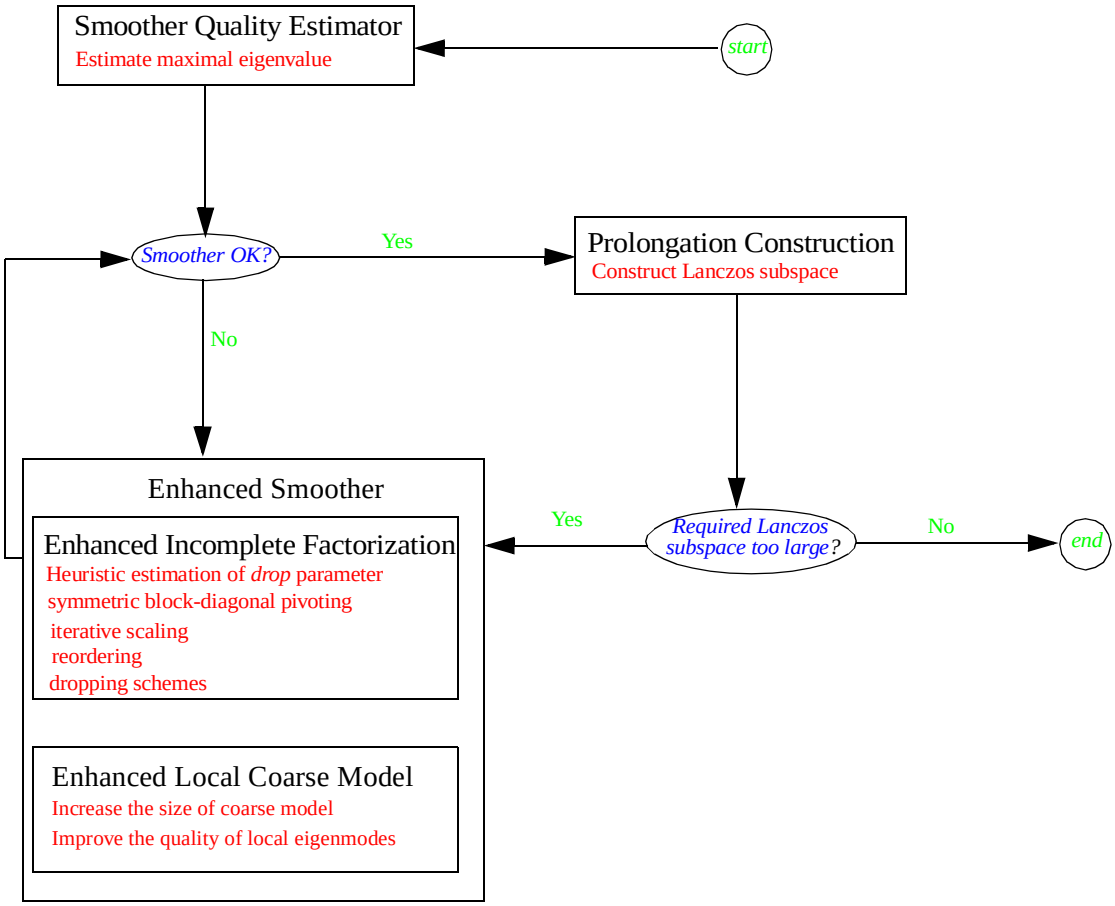

FIGURE 2. : Schematics of the heuristic adaptive two-level algorithm

3.0 Numerical Examples and Discussion

3.1 Helmholtz equation on bounded domains

Consider Helmholtz's equation in the region enclosed between two concentric cubes of length $2(\Gamma_0)$ and $6(\Gamma_1)$. The strong form of the governing equations is given as

$$\nabla^2 u(x) + k^2 u(x) = 0 \qquad x \in \Omega \tag{8}$$

$$\frac{\partial u}{\partial n} = \frac{\partial}{\partial r}(e^{ikr}/r)\Big|_{\Gamma_0} \qquad x \in \Gamma_0 \subset \partial\Omega \tag{9}$$

$$\frac{\partial u}{\partial n} = \frac{\partial}{\partial r}(e^{ikr}/r)\Big|_{\Gamma_1} \qquad x \in \Gamma_1 \subset \partial\Omega \tag{10}$$

where n is a coordinate in a direction normal to Γ_0 and Γ_1; $\Gamma_0 \cup \Gamma_1 = \partial\Omega$ and $\Gamma_0 \cap \Gamma_1 = 0$; r is distance from the center of the cube. Equations (8)-(10) describe the acoustic pressure u of a wave in a transmitting medium.

Because of symmetry, only one-eighth of the domain is discretized. Three meshes consisting of 3,072, 23,925 and 156,009 4-node linear tetrahedral elements have been considered. The coarsest discretization is shown in Figure 3.

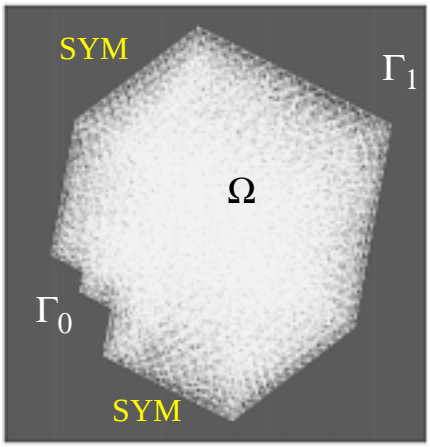

FIGURE 3. : Typical finite element mesh and boundary conditions

The resulting discrete linear system of equation, $Kx = f$, is symmetric, complex and indefinite. It is convenient to transform the complex symmetric linear system into a real symmetric system by replacing each term in the stiffness, K_{AB}, force vector, f_A, and the solution vector, x_A, by $\begin{bmatrix} K_{AB}^R & K_{AB}^I \\ K_{AB}^I & -K_{AB}^R \end{bmatrix}$, $\begin{bmatrix} f_A^R \\ f_A^I \end{bmatrix}$ and $\begin{bmatrix} x_A^R \\ -x_A^I \end{bmatrix}$ respectively. The superscripts R and I denote the real and imaginary parts which can be interpreted as two degrees-of-freedom per node.

We consider four approaches for the solution of discrete linear system of equations:

(i) Global-basis two-level method applied to source equations (*Source-GB*),

(ii) Local-basis two-level method applied to source equations [2] (*Source-LB*),

(iii) Local-basis two-level method applied to normal equations [2] (*Normal-LB*),

(iv) Direct sparse solver with minimum degree reordering [2] (*Direct*).

Schemes (ii) and (iii) use eigenfunctions defined on overlapping aggregates with Dirichlet boundary conditions for construction of prolongation operator [2]. SSOR is employed as smoother for Scheme (ii), while for Scheme (iii) we adopt the Incomplete Cholesky precondi-

tioner for normal equation [11] as a smoother. Scheme (i) and (ii) use QMR as an accelerator whereas a Conjugate Gradient acceleration for normal system is employed in Scheme (iii).

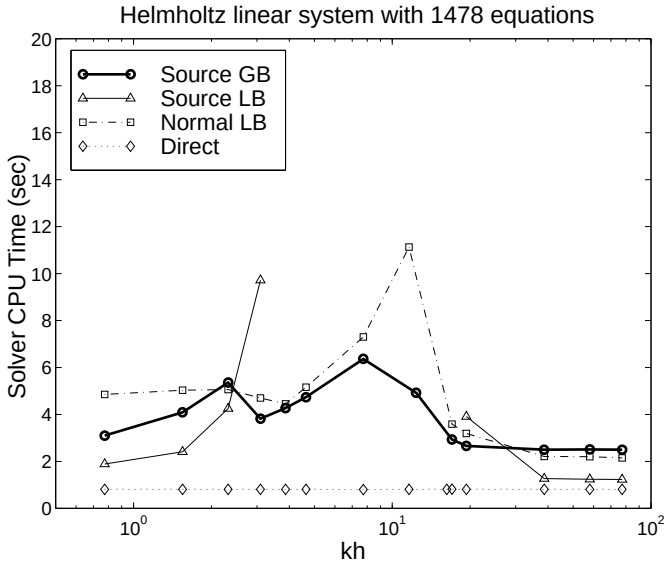

FIGURE 4. : CPU Time versus *kh* for discrete Helmholtz linear systems with 1,478 equations

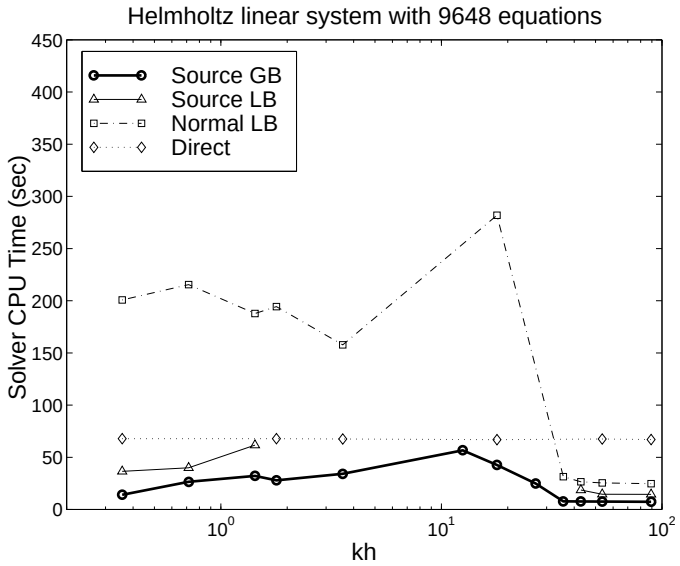

FIGURE 5. : CPU Time versus *kh* for discrete Helmholtz linear systems with 9,648 equations

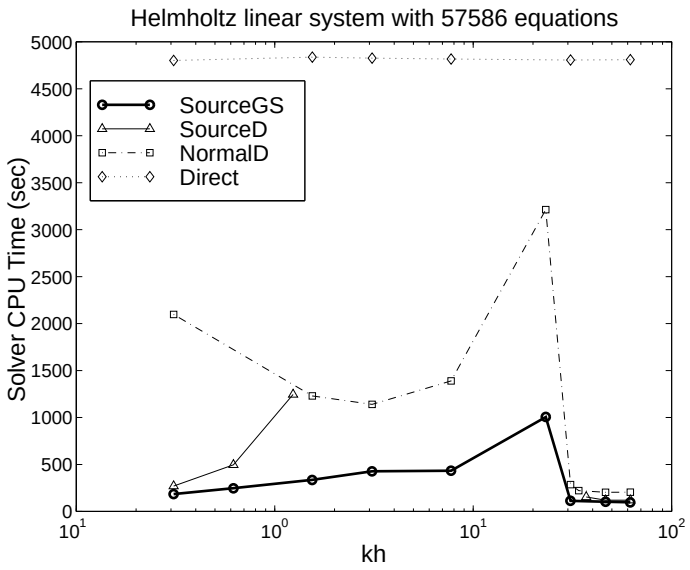

FIGURE 6. : CPU versus *kh* for Helmholtz linear system of 57,586 equations

Figures 4-6 show the CPU time on SUN ULTRA2/2200 versus the product of the average element size and the wave number, *kh*, for the three meshes considered. The product *kh* has been selected since it represents a measure of solution accuracy [12]. Even though practitioners dealing with wave propagation problems are primarily interested in the range, *kh<1*, required for solution accuracy[13], we conduct numerical experiments outside the range of the usual interest. Our interest in a much wider spectrum of *kh* values stems from the fact that not only the analyst may frequently encounter highly nonuniform meshes, where the precise definition of *h* is questionable, but primarily, because our ultimate goal is to develop a generic black-box equation solver for positive definite, weakly and highly indefinite systems.

It can be seen from Figures 4-6 that for the medium and large meshes the global-basis method (with a single-level smoother) applied to the source system (*Source-GB)* is the best solver over the whole spectrum of *kh* while the *Direct* solver is better suited for the small mesh. The break even point between the one- and two-level methods considered is approximately 5,000 equations. In the case of 50,000 unknowns, the global-basis two-level method is faster than the *Direct* solver by a factor of 5-30. The local-basis method [2] (*Source-LB)* only converges at both ends of the spectrum when the system is positive definite or slightly indefinite (*kh<2*) or negative definite (*kh > (20-40)*). Although the local-basis method applied to the normal system (*Normal-LB)* does converge for all situations, it is significantly slower than the global-basis method, especially when the system is highly indefinite (*2< kh < 20-40*). Convergence was measured in terms of the normalized L_2-norm of the residual with the tolerance of 10^{-6}.

For the global-basis method with a single level smoother, we used $\mu = 0.1$ for adaptive scaling and $\hat{\alpha} = 0.001$ and *m = 4* for pivoting. In the adaptive algorithm, the *tentative incomplete factorization* smoother is considered satisfactory if $|\lambda_{max}| < 10$. The initial drop tolerance for the *enhanced incomplete factorization* is determined as $droptol = min(1/(\sqrt{|\lambda_{max}|}), 0.01)$ and is reduced by factor of 5 for subsequent incomplete factorizations.

Figure 7 illustrates the effect of deviation from the optimal prolongation operator on the performance of the local-basis two-level method [2] in terms of iteration count. We use the small mesh with wave number $k = 10$ as an illustration. The Modified Incomplete Factorization with a threshold as described in Section 2 is employed as a smoother. In this example we consider the values of *droptol* = 0.15 and 0.05. The maximum eigenvalue of R is 20.8 and 0.8 for *droptol* equal 0.15 and 0.05, respectively. It can be seen that in both cases the coarse model increases the number of iterations. This result is consistent with our findings in Part 1 [1]. It can be seen that simply increasing the size of coarse model may not be helpful, and in this particular case it deteriorates the performance of the local-basis two-level method.

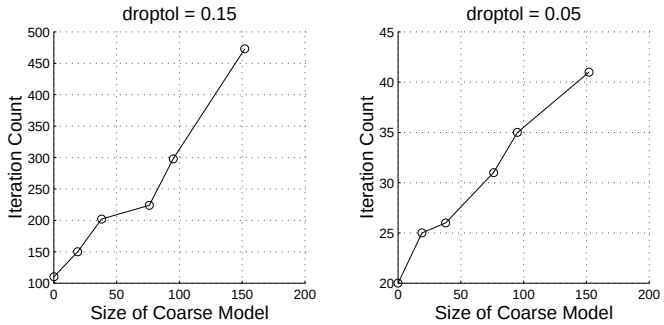

FIGURE 7. : Convergence of local-basis methods

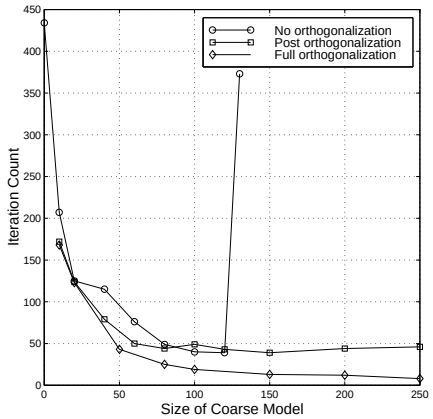

FIGURE 8. : Convergence of global-basis methods

Another form of deviation from the optimal prolongation is due to loss of orthogonality of the Lanczos vectors. The example we used is that of the small mesh with $k = 3$. Figure 8 compares various global-basis two-level method which utilize Lanczos vectors for prolongation operator. The Lanczos vectors have been generated by the Lanczos recursions with and without reorthogonalization. To distinguish the effect of prolongation error from the effect caused by near singularity of the coarse model due to linear dependence of Lanczos vectors, we orthogo-

nalize the Lanczos vectors after the completion of the Lanczos recursions without reorthogonalization (*post-orthogonalization*). The results are shown in Figure 8. It can be seen that the first 20 modes, generated by the Lanczos recursions without reorthogonalization, considerably reduce the iteration count while the following 90 modes only reduces the iteration count by half. It can also be seen that using more than 130 Lanczos vectors without orthogonalization makes the two-level method diverge. Even with post-orthogonalization, the number of the iterations increases when the size of coarse model is larger than 150 due to the deviation from the optimal prolongation. When the Lanczos vectors were generated with reorthogonalization (*full orthogonalization*), the number of iterations always decreased with the increase in the size of coarse model. This suggests that if a larger coarse model is needed, reorthogonalization must be applied in the process of Lanczos recursions to reduce the effects of prolongation error and near singularity of the coarse model. Also, as we have mentioned in Section 2.1.3, it is advantageous to apply the reorthogonalization in the Lanczos process to maximize computational efficiency.

Solution Methods			**Number of cycles and CPU Split-up Times**				
Smoother		**Global Basis Coarse Model**	**Smoother Construction**	**Global Basis Construction**	**# of cycles**	**Iterative Process**	**Solver Total**
Type	**Size**	**Size**	**CPU (sec)**	**CPU(sec)**		**CPU(sec)**	**CPU(sec)**
Single-level	0	0	1	0	>1000	NA	NA
	0	20/40/60	1	4/10/16	>1000	NA	NA
	0	120	1	26	461	629	656
GAM (Neumann)	319	0	6	0	>1000	NA	NA
	1459	0	10	0	830	237	247
	319	20	6	5	427	164	175
	1459	20	10	6	237	104	120
	319	40	6	11	274	125	142
	1459	40	10	12	108	56	78
	319	60	6	17	103	79	102
	1459	60	10	18	58	48	76
GAM (Dirichlet)	276	0	10	0	933	224	234
	317	0	11	0	555	139	150
	276	20	10	5	249	97	112
	317	20	11	5	153	61	77
	276	40	10	11	102	49	70
	317	40	11	11	73	38	60
	276	60	10	18	46	33	61
	317	60	11	18	37	30	59

TABLE 1. Comparision of Solvers

Results of the preliminary investigation on the application of multilevel preconditioners as smoothers within the framework of the global-basis two-level method are illustrated in Table 1. We considered the problem of the medium mesh with *k = 4*. The following solution schemes have been studied: The single-level smoother with or without the global-basis coarse model, the GAM smoother (Neumann BC) with or without global-basis coarse model, and GAM smoother (Dirichlet BC) with or without the global-basis coarse model. The *tentative incomplete factorization*, described in Section 2.4, was employed as a single-level smoother and as a relaxation scheme within the GAM preconditioner. It can be seen that the global-basis two-level method with a single-level smoother does not converge within 1000 steps unless over 120 global modes are selected, in which case the computational costs of the global basis coarse model construction dominates the entire computational cost. If, on the other hand, we employ the GAM preconditioner as a smoother, the computational cost can be significantly reduced.

3.2 Shear banding problem

We considered a linearized shear banding problem, illustrated in Figure 9. The cube is discretized with $16 \times 16 \times 16$, $24 \times 24 \times 24$ and $32 \times 32 \times 32$ 8-node hexahedral elements totaling to 14793, 46875 and 107811 degrees-of-freedom. We assume that a shear band (softening zone) develops on the diagonal plane of two layers of elements [14]. We considered the spectrum of ratios between the stiffness inside and outside the shear band, E_{band}/E, in the range of 0.3 and -0.7.

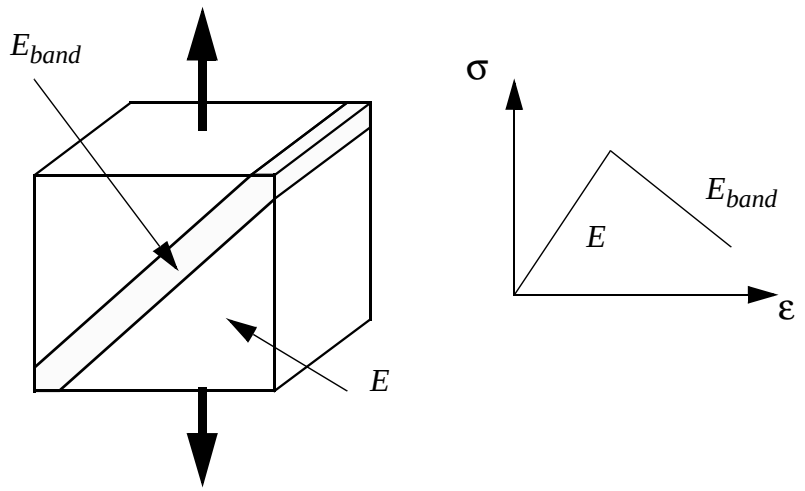

FIGURE 9. : Model for shear banding problem

The four approaches described in 3.1 have been tested. Figures 10-12 show the CPU time versus the measure of indefiniteness (E_{band}/E) for the three meshes considered. It can been seen that for all three meshes considered, the global-basis method has the best performance among the four approaches. The behavior of the local-basis methods is similar to that of the Helmholtz equations. It converges when the system is positive definite or weakly indefinite,

$E_{band}/E > 0.1$, but diverges for highly indefinite system, $-0.7 \le E_{band}/E \le -0.1$. The normal equation approach converges over the whole spectrum and is competitive with the direct solver for problems above 100,000 unknowns.

The influence of deviation from the optimal prolongation operator is similar to that illustrated in Figures 7 and 8. The performance of two-level method is very sensitive to the quality of the prolongation operator. The deviation from the optimal prolongation operator may deteriorate the performance of two-level method significantly.

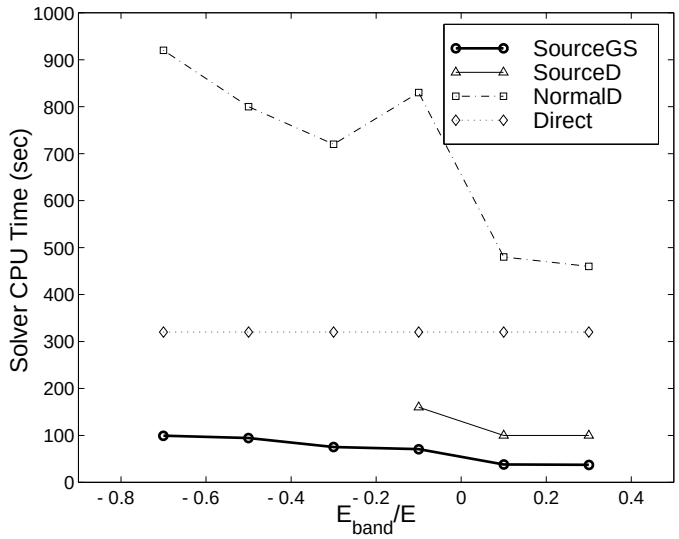

FIGURE 10. : CPU Time vs. E_{band}/E for shear banding problem with 14,739 equations

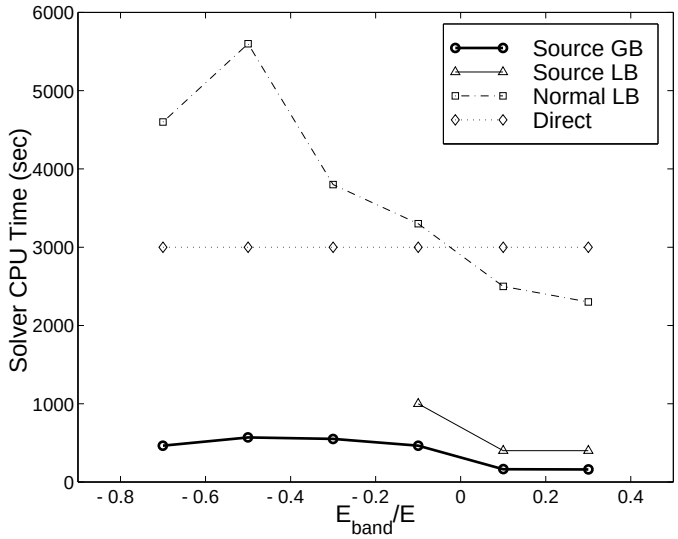

FIGURE 11. : CPU Time vs. E_{band}/E for shear banding problem with 46,875 equations

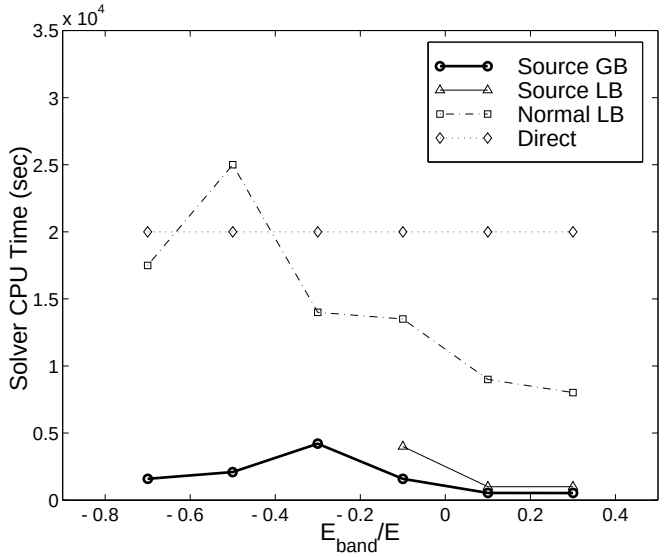

FIGURE 12. : CPU Time vs. E_{band}/E for shear banding problem with 107,811 equations

4.0 Summary

Algorithmic aspects and computational efficiency of the global basis two-level method, including nearly optimal global basis prolongation operator, single-level and multilevel smoothers, and a heuristic feedback control loop, were studied. Numerical experiments conducted on the Helmholtz equations and the shear banding problems affirm the potential of the method.

Further studies will be conducted to develop optimal smoothing preconditioners, and to extend the application of the solver to nonsymmetric indefinite systems. From the practical point of view, adaptivity and automation of the method are the key remaining issues.

ACKNOWLEDGMENT

This work was supported by the Office of Naval Research through grant number N00014-97-1-0687.

References

1 J. Fish and Y. Qu, “Global Basis Two-Level Method For Indefinite Systems. Part 1: Convergence studies,” submitted to *International Journal for Numerical Methods in Engineering*, (1999).

2 J. Fish, Y. Qu and A. Suvorov, “Towards robust two-level methods for indefinite systems,” *International Journal for Numerical Methods in Engineering*, Vol. 45, pp. 1433-

1456, (1999).

3 R. W. Freund and N. M. Nachtigal, “Software for Simplified Lanczos and QMR Algorithms,” *Applied Numerical Mathematics,* Vol. 19, pp. 319--341, (1995).

4 G.H. Golub and C.F. Van Loan, “Matrix Computations,” *The Johns Hopkins University Press,* (1983)

5 “An improved incomplete factorization,” *ACM Transaction on Mathematical Software,* Vol. 21,pp.5-17,(1995)

6 Chow, E. and Y. Saad, “Experimental Study of ILU Preconditioners for Indefinite Matrices,” *J. Comput. and Appl. Math.* Vol. 87, pp. 387-414, (1997).

7 J.R.Bunch and L. Kaufman, “Some Stable Method for Calculating Inertia and Solving Symmetric Linear Systems,” *Mathematics of Computation,* Vol. 31, pp. 163--179, (1977)

8 C.Rajakumar and C.R.Rogers, “The Lanczos Algorithm Applied to Unsymmetric Generalized Eigenvalue Problem”, *International Journal for Numerical Methods in Engineering,* Vol.32, pp. 1009-1026,(1991).

9 C.Ashcraft, R.G.Grimes and J.G.Lewis, “Accurate Symmetric Indefinite Linear Equation Solvers,” *SIAM J. MATRIX ANAL. APPL*. Vol. 20, No.2, pp. 513-561, (1998).

10 M. Benzi, D. B. Szyld, and A. van Duin,“Orderings for Incomplete Factorization Preconditioning of Nonsymmetric Problems,” *SIAM J. Scientific Computing* 20, 3, pp. 1652--1670,(1999).

11 Y.Saad, “Preconditioning techniques for nonsymmetric and indefinite linear systems,” *Journal of Computational and Applied Mathematics*, Vol.24, pp.89, (1988).

12 A.Bayliss, C.I.Goldstein and E.Turkel, “On accuracy conditions of the numerical computation of waves,” *Journal of Computational Physics*, Vol. 59,(1985)

13 N.N.Abboud and P.M. Pinsky, “Finite element dispersion analysis of the numerical computation of waves,” *Int. J. For Numerical Methods In Engineering*, Vol. 35, pp. 1183-1218, (1992)

14 T.Belytschko, J.Fish and B. Engelmann, “A finite element with embedded localization zones,” *Comp. Meth. Appl. Mech. Engng*., Vol.108, (1993).