A Basic Parallel Sparse Eigensolver for Structural Dynamics

David Day

February 3, 1998

Abstract

In this work the basic Finite Element Tearing and Interconnecting (FETI) linear system solver and the PARPACK eigensolver are combined to compute the smallest modes of symmetric generalized eigenvalue problems that arise from structures modeled 'primarily' by solid finite elements. Problems with over one million unknowns are solved. A comprehensive and relatively self-contained description of the FETI method is presented.

1 Introduction and Summary

We seek to compute the left-most modes of a symmetric generalized eigenvalue problem $Ax = Bx\lambda$ with more than one million unknowns arising from structural dynamics using a distributed memory platform with a limited amount of memory per processor. This generalized eigenvalue problem is solved using PARPACK [9, 10] in shift-invert mode. The Finite Element Tearing and Interconnecting (FETI hereafter) iterative linear system solver is used to invert [5]. Other researchers have combined FETI with an eigensolver [1, 2]. Our approach is more comprehensive and we solve much larger problems. The bulk of this work describes the basic FETI multilevel method for solving linear systems whose coefficient matrix is a positive definite stiffness matrix (which would be singular in the free boundary case) for solid structures (whose underlying partial differential equation is second order elliptic). Section two describes the reformulation of the linear system as a distributed interface linear system. Section three develops the FETI iterative solution algorithm. In section four the combined FETI/PARPACK method is applied to several model eigenvalue problems of order up to one million.

2 Formulation of the Interface Linear System

FETI is a *dual* Schur-complement domain decomposition method. The finite elements are partitioned instead of the nodes in the finite element mesh. In other words, the dual of the graph is partitioned (hence the name). This approach is natural for finite element modeling because assembly of the subdomain stiffness matrix requires no interprocessor communication. On subdomains on which no boundary conditions are imposed (the boundary conditions which make A nonsingular) the subdomain stiffness matrix is singular.

Substructuring (i.e. domain decomposition) represents an unassembled structure stiffness matrix in the form $L^T KL$ where K is a block diagonal matrix of substructure stiffness matrices and L is called a Boolean assembly matrix [11]. The linear system

$$(L^T KL)u = f. \tag{1}$$

is traditionally solved by (multi-frontal) Gaussian elimination applied to the product $L^T KL$. This note explains how FETI solvers exploits the parallelism inherent in the representation (1).

A more precise definition of L is required. L maps the unknowns to substructure unknowns. Each row of L corresponds to a substructure unknown and each column of L corresponds to a structure unknown. In each row of L there is precisely one nonzero entry. $L_{i,j} = 1$ if substructure

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract DE-AC04-94AL85000.

DISCLAIMER

node i corresponds to node j. For example $\Delta := L^T L$ is a diagonal matrix whose (j,j) entry is the number of substructures sharing the jth global unknown.

The FETI approach is to decompose (1) into the three linear systems

$$\begin{aligned} L^T p &= f && p \text{ not unique} \\ K u_s &= p && u_s \text{ not unique} \\ & && u_s \text{ exists} \iff p \in \mathcal{R}(K) \\ L u &= u_s && u \text{ exists} \iff u_s \in \mathcal{R}(L) \end{aligned}$$

Here $\mathcal{R}(L)$ denotes the range of L. The constraints that u_s and u exist determine p and u_s.

Because K is singular, it is necessary to compute the action of the pseudo-inverse, K^+, on a vector. We digress to discuss this task in detail. A stable algorithm for computing K^+ involves using the eigendecomposition of K which is too costly. An efficient method is to use the factorization $K = LDL^T$ to determine the null space $\mathcal{N}(K)$. In general pivoting during the factorization for stability reasons. Using the properties of solid elements, we are able to permute the subdomain unknowns so that on connected subdomains pivoting is unnecessary until near the end of the factorization. At that point an eigendecomposition is used. The major problem with this technique is that so far we have been unable to ensure that the subdomains are connected.

Particular solutions to the first two equations are $\tilde{p} := L\Delta^{-1} f$ and $\tilde{u}_s := K^+ p$. The general solutions can be written in the form $p = \tilde{p} - Nx$, $u_s = \tilde{u}_s - N_r x_r$ and $u = \Delta^{-1} L^T u_s$ where the columns of N are a basis for the null space of L^T (that is $L^T N = 0$), and the columns of N_r are an orthonormal basis for the null space of K. Let $\mathcal{N}(K)$ denote the null space of K. The symbol λ is typicly used in place of x to remind the reader that this is a vector of Lagrange multipliers.

It suffices for the columns of N_r to be ortho-normal and span $\mathcal{N}(K)$. In other words the FETI formulation applies if extra columns are added to N_r. For example one could include a few extraordinarily small nonzero eigenmodes of a subdomain stiffness matrix.

There exist unique vectors x and x_r such that u_s and u exist. Because $\mathcal{N}(K^T) = \mathcal{N}(K)$ is the orthogonal complement of $\mathcal{R}(K)$,

$$0 = N_r^T p = N_r^T \tilde{p} - N_r^T N x$$

implies that $p \in \mathcal{R}(K)$. Do the same thing for $u_s \in \mathcal{R}(L)$ and there appears

$$\begin{bmatrix} N^T K^+ N & N^T N_r \\ N_r^T N & 0 \end{bmatrix} \begin{bmatrix} x \\ x_r \end{bmatrix} = \begin{bmatrix} N^T K^+ \tilde{p} \\ N_r^T \tilde{p} \end{bmatrix}$$

Section three describes how to solve this equation. Given the solutions x and x_r, one can then compute first p, next u_s and finally the (global) solution u.

3 The FETI Method

In this section the basic FETI method for solving the reduced interface linear system is developed. The indefinite interface linear system is first reduced to a symmetric positive semi-definite linear system. Dirichlet preconditioning is used. The resulting iterative method is an instance of a preconditioned conjugate projected gradient method [5].

We introduce a constraint matrix B which replaces the matrix N^T discussed in section 2. The rows of B are of the form $[\ldots 1 \ldots -1 \ldots]$ where the location of the two nonzero entries corresponds to two unknowns that correspond to the same global unknown. If an unknown on a given processor corresponds to several unknowns on other processors, then one row is added to B for each constraint. Constraining two unknowns to be equal is achieved using $C_2 = [1 \; -1]$. Constraining three unknowns to be equal is achieved using

$$C_3 = \begin{bmatrix} 1 & -1 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & -1 \end{bmatrix}$$

Constraining $n+1$ unknowns to be equal is achieved using

$$C_{n+1} = \begin{bmatrix} e & -I \\ 0 & C_n \end{bmatrix}$$

where e is a vector of ones of appropriate length. A remarkable property of this choice of redundant constraints is that $C_k C_k^T$ has only one nonzero eigenvalue, k, with multiplicity $k-1$. The null space of $C_k C_k^T$ has dimension $(k-1)(k-2)/2$.

Solution of the symmetric indefinite linear system

$$\begin{bmatrix} F & G \\ G^T & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} d \\ e \end{bmatrix} \tag{2}$$

can be problematic for iterative solvers such as SYMMLQ. The preconditioned conjugate projected gradient method (PCPG) solves an equivalent constrained positive definite system.

The eponymous projector in PCPG annihilates G,

$$\Pi = I - G(G^T G)^{-1} G^T.$$

Multiply row 1 of equation (2) by Π to eliminate y from

$$\Pi F x + \Pi G y = \Pi F x = \Pi d.$$

Note that once x has been computed, y is determined from

$$(G^T G) y = G^T (d - Fx).$$

To compute x approximate the correction c, $x = c + x_0$, to $x_0 = G(G^T G)^{-1} e$. Now $G^T x_0 = e = G^T x$ implies that $G^T c = 0$. Thus $c = \Pi c$ solves

$$(\Pi F \Pi) c = \Pi (d - F x_0)$$

It is possible to view FETI as a multilevel method with coarse space $\mathcal{N}(K)$ and coarse space preconditioner $(G^T G)^{-1}$. $G = BR$ where the rows of B, $[\ldots 1 \ldots -1 \ldots]$, act like a gradient operator. Thus $G^T G$ contains a product of gradients, like a *Laplacian*. And $(G^T G)^{-1}$ acts like an inverse Laplacian preconditioner, suitable for second order elliptic partial differential equations.

A Dirichlet preconditioner $\tilde{F}$ is an approximate inverse of $F = BK^+B^T$. To define $\tilde{F}$, more notation will be needed. Let p denote the number of substructures and let K_i and B_i denote the substructure stiffness and constraint matrices:

$$K = \mathrm{diag}(K_1, \ldots, K_p) \qquad B = [B_1, \ldots, B_p]$$

Then $F = \Sigma_i B_i K_i^+ B_i^T$. Our goal is to define a kind of inverse of $B_i K_i^+ B_i^T$. Each row of B represents a constraint and is of the form $[\ldots 1 \ldots -1 \ldots]$. But the columns of B that correspond to interior nodes vanish. We define $\hat{B}_i$ to be the matrix obtained by deleting the zero columns from B_i. If a global unknown corresponds to an unknown on only one subdomain, then the corresponding subdomain unknown is called an interior unknown. But if a global unknown corresponds to unknowns on more than one subdomain, then the corresponding subdomain unknowns are called boundary unknowns. Thus $B_i K_i^+ B_i^T = \hat{B}_i X_i \hat{B}_i^T$ where X_i consists of the elements of K_i^+ that correspond to boundary nodes.

Next we show that X_i is a Schur complement of K_i. Since we will be discussing one subdomain only in this paragraph, the subscript i is dropped. K can be permuted so that interior unknowns precede all boundary unknowns. In this case,

$$K = \begin{bmatrix} K_{ii} & K_{ib} \\ K_{bi} & K_{bb} \end{bmatrix} = \begin{bmatrix} I & 0 \\ K_{bi} K_{ii}^{-1} & I \end{bmatrix} \begin{bmatrix} K_{ii} & K_{ib} \\ 0 & S \end{bmatrix}$$

where $S = K_{bb} - K_{bi} * K_{ii}^{-1} K_{ib}$. If S were nonsingular then

$$K^{-1} = \begin{bmatrix} K_{ii}^{-1} & -K_{ii}^{-1} K_{ib} S^{-1} \\ 0 & S^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -K_{bi} K_{ii}^{-1} & I \end{bmatrix} = \begin{bmatrix} * & * \\ * & S^{-1} \end{bmatrix}$$

and $X = S^{-1}$.

Dirichlet preconditioning approximates the inverse of the sums, $F^+ = (\Sigma_i \hat{B}_i S_i^+ \hat{B}_i^T)^+$ by the sum of the inverses $\tilde{F} = \Sigma_i \hat{B}_i S_i \hat{B}_i^T$.

The effectiveness of the Dirichlet preconditioner is enhanced by including a diagonal scaling matrix W^{-1} and using $W^{-1}\tilde{F}W^{-1}$ as an approximate inverse of F [6]. The scaling can be derived in the special case in which each $K_i = I$ [16]. First observe that $\tilde{F} = F = BB^T$. The matrix BB^T can be permuted into a direct sum of matrices of the form $C_k C_k^T$ and the zero matrix. Recall that $C_k k^{-1} C_k^T$ is a projection. We define $W = \text{diag}(W_1, ..., W_p)$ where the (k,k) entry of W_i is the number of unknowns in any subdomain that correspond to the same (global) unknown as the k-th boundary unknown on the i-th subdomain. Then (in the special case $K_i = I$) $W^{-1/2}FW^{-1/2}$ is a projection and $F^+ = W^{-1}\tilde{F}W^{-1}$.

In summary, the basic FETI method is implemented as follows.
Initialize: $x_0 = G(G^TG)^{-1}e$ and $r_0 = d - Fx$
For $k = 1, 2, ...$

1. $w_{k-1} = \Pi r_{k-1}$
2. $z_{k-1} = W^{-1}\tilde{F}W^{-1}w_{k-1}$
3. $y_{k-1} = \Pi z_{k-1}$
4. $\zeta_{k-1} = y_{k-1}^T z_{k-1} / y_{k-2}^T z_{k-2} \quad (\zeta_1 = 0)$
5. $p_k = y_{k-1} + p_{k-1}\zeta_{k-1} \quad (p_1 = y_0)$
6. $p_k := (I - P\Omega^{-1}Q^T)p_k \quad (k > 0)$
7. $P(:,k) = p_k$
8. $Q(:,k) = Fp_k$
9. $\Omega(k,k) = P(:,k)^T Q(:,k)$
10. $\nu_k = y_{k-1}^T z_{k-1} / \omega_k$
11. $x_k = x_{k-1} + p_k \nu_k$
12. $r_k = r_{k-1} - Q(:,k)\nu_k$

Remark. In steps 6, 7 and 8 the search directions are stored and the current the search direction is explicitly re-orthogonalized against all the previous search directions. This extra work is observed to be worthwhile [12].

Remark. Iteration stops once $\|w_k\| < TOL\|w_0\|$. The reduction in $\|w\|$ is observed to be proportional to the reduction in $\|Au - f\|$.

4 Eigenvalue Problems

In this section the FETI method is applied to a few model generalized eigenvalue problems. We first discuss the extension of the algorithm to eigenvalue problems and we conclude with a summary of proposed future work.

A given structure is first decomposed or partitioned using the DOMDEC decomposition package [4, 13, 7]. Using DOMDEC makes it possible to determine subdomains with low geometric aspect ratio, which is critical for FETI [5].

To apply the FETI method, it is necessary to solve many linear systems which coefficient matrix G^TG. We use a full sweep of the conjugate gradient method with stable DKGS [3] reorthogonalization to compute the decomposition $(G^TG)^{-1} = PD^{-1}P^T$ where D is diagonal and save P and D. This algorithm is scalable, simple to implement and easy to adapt to more complex applications (e.g. singular G^TG). More efficient algorithms could be used here, such as a parallel sparse direct solver. But for eigenvalue problems the cost of inverting G^TG is amortized over many solves, and the difference in cost of these approaches is insignificant.

The PARPACK package is used in shift-invert mode to compute the smallest eigenvalues of (A, M). The tolerance used is $\tau = 10n\epsilon$ where ϵ is the machine precision and n is the number of global unknowns. The FETI method is used to compute u such that $\|Au - f\| < \tau\|f\|$.

In our first set of model problems are steel cubes clamped at one face and discretized using brick elements. The substructures are irregular. First we applied to FETI solver to various problems, holding n/p the number of unknowns per processor fixed.

Grid	Procs	Iters
6^3	8	20
12^3	64	37
18^3	216	41
24^3	512	43

We observed that the number of iterations required to solve the linear system to a fixed accuracy grew modestly with the number of processors.

Next we applied the eigensolver to a $32 \times 32 \times 32$ cube, in which case there are roughly $100K$ unknowns, and computed the smallest mode using different numbers of processors.

Procs	Time (Min)
64	24
125	11
256	6

We have also computed the smallest mode for a $64 \times 64 \times 64$ cube with $800K$ unknowns. This computation too 36 minutes using 512 processors. The time to compute additional modes increases roughly linearly in the number of requested modes.

We have also applied the FETI solver to a family of model structures with more complex geometry. These structures are challenging to model because one of the materials is a rubber whose stiffness is a factor of one million less than the stiffnesses of the other metal materials. For a model with over one million unknowns, we have attempted to reduce the residual error by a factor of approximately 10^{-9}. For these applications a generalized scaling of the Dirichlet preconditioner is used. If a global unknown corresponds to unknowns i_k on subdomains p_k, then the entry of W that corresponds to i_j is

$$(\Sigma_k K^{(p_k)}_{i_k,i_k})/K^{(p_j)}_{i_j,i_j})$$

instead of $\Sigma_k 1$ [8]. But after 600 iterations the residual had only been reduced by a factor of 10^{-5}. If the rubber is removed from the structure, then FETI converges in 275 iterations. In this case we are able to compute the 10 smallest modes of the system in approximately six hours using 256 processors. And if all of the materials are replaced by a single material, then 200 iterations suffice to solve a typical linear system.

Future work on this approach starts with modifications to FETI necessary to solve highly heterogeneous problems. We are experimenting with modifying the decomposition, the scaled Dirichlet preconditioner and the coarse space. Next extensions to solve linear systems in which the stiffness matrix is singular are needed. In this case it is possible to determine $\mathcal{N}(A)$ from $\mathcal{N}(G^T G)$ [1, 2]. Experiments with an eigensolver based on inexact solves, such as a restarted Jacobi-Davidson method are needed [14, 15]. Techniques to (usually) detect missed modes by restarting will need to be included. A check for degeneracy, $\mathcal{N}(A) \cap \mathcal{N}(M) \neq 0$, given $\mathcal{N}(A)$ is needed. And enhancements of the graph partitioning software are needed to ensure that each subdomain has one connected component, to ensure that all brick or wedge elements are connected at at least three vertices to other elements in the same subdomain and to align subdomains with certain material interfaces.

References

[1] J.-M. Cros and F. Léné (1996). Parallel iterative methods to solve large-scale eigenvalue problems in structural dynamics, In Bjørstad P., Espedal M. and Keyes D. (eds) *Proc. Ninth Int. Conf. on Domain Decomposition Meths.*, John Wiley and Sons Ltd.

[2] J.-M. Cros (1997). Résolution de problèmes aux valeurs propres en calcul des structures par utilisation du calcul parallèle. Doctoral dissertation, Ecole Normale Supérieure de Cachan.

[3] J. Daniel, W. Gragg, L. Kaufman and G. Stewart (1976). Reorthogonalization and stable algorithms for updating the Gram-Schmidt QR factorization, *Math. Comp. 30*, 772-795.

[4] C. Farhat (1988). A simple and efficient automatic FEM domain decomposer, *Comput. and Struct. 28*, 579-602.

[5] C. Farhat and F. Roux (1992). An unconventional domain decomposition method for an efficient parallel solution of large-scale finite element systems, *SIAM J. Sci. Stat. Comp. 13*, 379-396.

[6] C. Farhat and F. Roux (1993). Implicit parallel processing in structural mechanics, CU-CAS-93-26, College of Engineering, University of Colorado, Campus Box 429, Boulder, CO 80309.

[7] C. Farhat and M. Lesoinne (1993). Automatic partitioning of unstructured meshes for the parallel solution of problems in computational mechanics, *Internat. J. Numer. Meths. Engrg. 36*, 745-764.

[8] C. Farhat and D. Rixson (1996). Preconditioning the FETI and balancing domain decomposition methods for problems with intra- and inter-subdomain coefficient jumps, In Bjørstad P., Espedal M. and Keyes D. (eds) *Proc. Ninth Int. Conf. on Domain Decomposition Meths.*, John Wiley and Sons Ltd.

[9] R. Lehoucq, D. Sorensen and C. Yang (1998). *ARPACK USERS GUIDE: Solution of Large Scale Eigenvalue Problems by Implicitly Restarted Arnoldi Methods* SIAM, Philadelphia, PA.

[10] K. Maschhoff and D. Sorensen (1996). PARPACK: An Efficient Portable Large Scale Eigenvalue Package for Distributed Memory Parallel Architectures, *Lecture Notes in Computer Science 1184*, Springer-Verlag, Berlin.

[11] K. C. Park and C. Felippa (1996). A variational framework for solution method developments in structural mechanics, CU-CAS-96-21, College of Engineering, University of Colorado, Campus Box 429, Boulder, CO 80309.

[12] F.-X. Roux (1990). Acceleration of the outer conjugate gradient by reorthogonalization for a domain decomposition method for structural analysis problems.
In Chan T. F., Glowinski R., Périaux J., and Widlund O. B. (eds) *Proc. Third Int. Conf. on Domain Decomposition Meths.*, 314-319, SIAM, Philadelphia.

[13] H. Simon (1991) Partitioning of unstructured problems for parallel processing, *Comput. Sys. Engrg. 2*, 135-148.

[14] G. Sleijpen, J. Booten, D. Fokkema and H. van der Vorst (1996). Jacobi-Davidson type methods for generalized eigenproblems and polynomial eigenproblems, *BIT 36*, 595-633.

[15] G. Sleijpen and H. van der Vorst (1996). A Jacobi-Davidson iteration method for linear eigenvalue problems, *SIMAX 17*, 401-424.

[16] R. Tuminaro (1996). Private communication.

Report Number (14) SAND--98-0410C
CONF-9705260--

Publ. Date (11) 199802
Sponsor Code (18) DOE/CR , XF
JC Category (19) UC-905 , DOE/ER

DOE