37

Parallel Iterative Methods for Large-scale Eigenvalue Problems in Structural Dynamics

Jean-Michel Cros, Françoise Léné

1 Introduction

Consider the following generalized eigenvalue problem:

$$Kq = \lambda Mq \tag{1.1}$$

where K and M are respectively the symmetric stiffness matrix and mass matrix, the eigenvalues λ are the squares of the natural frequencies ω, and q are the eigenvectors, although only the smallest eigenpairs are wanted. We are interested in problems where K and M are very large (more than 10^4 unknowns), sparse symmetric positive definite (or semi-definite) matrices. The main difficulty is to deal with large matrices which exceed primary memory capacity of sequential computer. Distributed memory architectures have enough memory, and to take advantage of these computers algorithms must involve large tasks that can be executed in parallel. Iterative methods represent a way of developing such algorithms. Iteration techniques to solve problem (1.1), require that the system is reduced to a standard eigenproblem:

$$K^{-1}Mq = \frac{1}{\lambda}q \tag{1.2}$$

and thus a linear system has to be solved (either K^{-1} or an approximation of K^{-1} [SVdV96], [BKP97]). However in structural analysis many problems occur, such as mutiplicity of eigenvalues, semi-definiteness, etc. [GLS86], and some of these have been adressed by the developments in the Lanczos algorithm. In addition the Lanczos method seems to be the most well suited algorithm for very large problems, because it requires few iterations per converged eigenvalue, and this number remains independent of the problem size. But for large problems, a robust and parallel linear solver is required without the use of secondary menory during computation. In static analysis, for geometrical or materials reasons, the stiffness matrix is often ill-conditioned. Iterative substructuring or domain decomposition methods, such as Schur complement

Ninth International Conference on Domain Decomposition Methods
Editor Petter E. Bjørstad, Magne S. Espedal and David E. Keyes

methods, have a nice mechanical interpretation. They have proven their numerical and parallel scalability and are better than a direct method [FC95] for this kind of problem. From the CPU time point of view, the Schur dual complement method is more attractive than the primal approach [Cro97], due to the use of an economical preconditioner, and the ease to solve in a parallel way the coarse grid induced by rigid body. Therefore, the large sparse linear system, at each iteration of the Lanczos algorithm, is solved by the dual Schur complement method.

The paper is organized as follows: section 2 recalls briefly the Lanczos method and its parallel implementation. Section 3 presents the way of computing the global rigid-body modes. Section 4, describes a restarting technique to take into account the successive right-hand sides in order to reduce the number of iterations and section 5 is devoted to an extension of this technique. In Section 6, some numerical results obtained on the Intel PARAGON computer, using the finite element package MODULEF (INRIA) in a message-passing environment, are presented.

2 Lanczos Algorithm

The Lanczos algorithm for extracting the smallest eigenpairs of a system is an inverse power-based method. In its basic form, it is an algorithm for computing an orthogonal basis of a Krylov subspace, i.e., a subspace of the form:

$$\mathcal{K}_r = span\{y_0, K^{-1}My_0, ..., (K^{-1}M)^{r-1}y_0\}. \tag{2.3}$$

The main iteration of the algorithm can be briefly described by the following recurrence

$$\beta_r y_{r+1} = K^{-1}My_r - \alpha_r y_r - \beta_{r-1}y_{r-1},$$

where α_r and β_{r-1} are selected in such a way that the vector y_{r+1} is M-orthogonal to y_r and y_{r-1}:

$$\alpha_r = y_r^T MK^{-1}My_r \quad \text{and} \quad \beta_{r-1} = y_r^T MK^{-1}My_{r-1}. \tag{2.4}$$

Then, (2.4) can be expressed in matrix form as follows:

$$K^{-1}MY_r = Y_rT_r + S \ \text{ with } Y_r = [\ y_0 \ ... \ y_r\ ] \text{ and } S = [\ 0 \ ... \ \beta_{r+1}y_{r+1}\ ], \tag{2.5}$$

where T_r is a tridiagonal matrix. The application of the Rayleigh-Ritz procedure to the standard form of the initial eigenvalue problem (1.1), by a projection into the subspace generated by the Lanczos vectors $q = Y_rz$, leads to the reduced eigenvalue problem:

$$T_rz = \frac{1}{\omega^2}z. \tag{2.6}$$

We refer to [CG82] for practical considerations of the Lanczos algorithm, such as: choice of starting vector, restart procedure to take into account possible multiple eigenvalues, convergence strategy, eigenmodes and error analysis.

From the numerical point of view the most CPU time consuming operations are:

- solution of the large-scale linear system with K^{-1}
- M-orthogonalization of y_r
- computation of matrix-vector products with M

These tasks are naturally parallelized by substructuring. The physical domain is divided in N_s nonoverlapping subdomains, and the problem on a global domain is replacing by solving iteratively a condensed problem on the interface of the subdomains. This involves at each iteration solving of locally independant problems. Then, each subdomain is allocated to a processor which knows only the data corresponding to its subdomain and information about neighboring subdomains through interface decomposition. As a sequel, a processor is responsible for computing a fixed subset of each vector (Lanczos vector, search directions,etc.). The internal problem in each subdomain is solved by a direct method while the interface problem, which incorporates a coarse grid induced by rigid body modes of subdomains without external Dirichlet conditions, is handled by a parallel Preconditioned Conjugate Projected Gradient (PCPG) method [FR94]. Finally, a full reorthogonalization of the Lanczos vectors is performed. The reduced eigenvalue problem (2.6) is solved in a sequential way thanks to suitable methods from optimized LAPACK library.

3 Global Rigid-body Modes within Substructuring Framework

Structures having rigid-body modes arise frequently, especially for aeronautic applications. In these cases the inverse iteration process which consists in:

$$\begin{aligned} 1)\quad g_r &= My_r && (3.7)\\ 2)\quad y_{r+1} &= K^{-1}g_r && (3.8) \end{aligned}$$

is modified to filter rigid-body modes [GR94] and becomes:

$$\begin{aligned} 1)\quad g_r &= My_r && (3.9)\\ 2)\quad \tilde{g}_r &= g_r-(MR,g_r)R && (3.10)\\ 3)\quad \tilde{y}_{r+1} &= K^{-1}\tilde{g}_r && (3.11)\\ 4)\quad y_{r+1} &= \tilde{y}_{r+1}-(MR,\tilde{y}_{r+1})R && (3.12) \end{aligned}$$

where the matrix R stores the rigid-body modes of the structure. Step 2 and step 4 respectively express the self-equilibrium of the inertia load, and the fact that the new Lanczos vector is M-orthogonalized with respect to the rigid-body modes. The difficulty is then to compute the global rigid-body modes of the structure. We recall the algebraic system induced by the dual Schur complement method [FR94], and we note with an (s) superscript, a matrix or a vector quantity associated to the s^{th} given substructure:

$$K^{(s)}u^{(s)} = f^{(s)} - B^{(s)^T}\mu \quad \text{in } \Omega^{(s)},\ \text{for } s=1,...,N_s, \qquad (3.13)$$

$$\sum_{s=1}^{N_s} B^{(s)}u^{(s)} = 0 \quad \text{on } \Gamma, \qquad (3.14)$$

where the vector of Lagrange mutipliers μ represents the interaction forces between the substucture $\Omega^{(s)}$ with $s = 1, ..., N_s$ along their interface Γ, u is the displacement vector, f the loading vector, and $B^{(s)}$ is a signed boolean matrix which localizes a substructure quantity to the substructure interface $\Gamma^{(s)}$.

If the global structure has no Dirichlet boundary conditions, it will be considered as floating. Hence the stiffness matrix is singular and the restriction of its rigid-body mode u_r in each subdomain verifies the following relations:

$$K^{(s)} u_r^{(s)} = 0 \quad \text{in } \Omega^{(s)}, \text{ for } s = 1, ..., N_s, \tag{3.15}$$

$$\sum_{s=1}^{N_s} B^{(s)} u_r^{(s)} = 0 \quad \text{on } \Gamma. \tag{3.16}$$

Let us then introduce the convention of Farhat and Roux [FR94], the following new quantities:

$$G^{(s)} = B^{(s)} R^{(s)} \quad \text{and } G = [G^{(1)} \dots G^{(N_f)}], \tag{3.17}$$

where the matrix $R^{(s)}$ stores the rigid-body modes of the substructure $\Omega^{(s)}$. In such a situation, the number of floating substructures N_f, is equal to the number of substructures N_s and G does not have full column rank, thus a set of nonzero coefficient Ψ exists such that:

$$\sum_{s=1}^{N_s} B^{(s)} R^{(s)} \Psi^{(s)} = 0 \quad \text{on } \Gamma. \tag{3.18}$$

This equation implies that the rigid displacement field defined by $R^{(s)}\Psi^{(s)}$ in each substructure is continuous across the substructure interface and satisfies definition (3.15), or in other words, the *global* rigid-body modes can be expressed as a linear arrangement of the *local* rigid-body modes. It can be shown [LT90] that $Ker(G) = Ker(G^T G)$, and then Ψ is also solution of the following problem:

$$G^T G\, \Psi = 0 \quad \text{where } \Psi^T = \{\Psi^{(1)^T}, ..., \Psi^{(N_s)^T}\}, \tag{3.19}$$

which provides an easy way to compute the matrix Ψ. To solve problem (3.19), matrix $G^T G$ must be assembled to compute singularities. Let us note that the coarse problem $G^T G$ is the same as previously where the decomposition induces some floating $(N_f < N_s)$ substructures, but the original problem is well posed. Finally, the relations (3.10) and (3.12) are respectively replaced by:

$$\tilde{g}_r^{(s)} = g_r^{(s)} - \gamma\, R^{(s)} \Psi^{(s)} \quad \text{with } \gamma = \sum_{s=1}^{N_s} (M^{(s)} R^{(s)} \Psi^{(s)}, g_r^{(s)}) \tag{3.20}$$

$$\tilde{y}_{r+1}^{(s)} = \tilde{y}_{r+1}^{(s)} - \gamma\, R^{(s)} \Psi^{(s)} \quad \text{with } \gamma = \sum_{s=1}^{N_s} (M^{(s)} R^{(s)} \Psi^{(s)}, \tilde{y}_{r+1}^{(s)}) \tag{3.21}$$

4 Successive Right-hand Sides

The presented technique has been analyzed by Saad [Saa87], and applied to improve substructure based iterative solver for different applications [FC95] [RTD95]. The domain decomposition method leads to an interface problem which is solved thanks to a conjugate gradient method.

$$C\mu = b, \tag{4.22}$$

where C is the interface operator. The conjugate gradient algorithm generates an orthogonal basis for the Krylov subspace $\mathcal{K}_k = span(g_0, Cg_0, ..., C^{k-1}g_0)$, where $g_0 = b - C\mu_0$ is the initial residual, and μ_0 an initial guess. Let us assume known the k first search directions, thus the approximate solution at the $(k+1)^{th}$ iteration can be expressed by:

$$\mu_{k+1} = \mu_0 + \sum_{i=1}^{k} \frac{(b - C\mu_0, w_i)}{(Cw_i, w_i)} w_i. \tag{4.23}$$

At each iteration of the Lanczos algorithm, the same interface problem has to be solved with a new right-hand side:

$$C\,\mu^2 = b^2. \tag{4.24}$$

The information (p^1 search directions collected) from the solution of the first system (4.22) is used to provide an optimal guess $\mu^2_{0,opt}$ for the solution of the second system thanks to expression (4.23).

$$\mu^2_{0,opt} = \mu^2_0 + \sum_{i=1}^{p^1} \frac{(b^2 - C\mu^2_0, w_i)}{(Cw_i, w_i)} w_i. \tag{4.25}$$

The generalization of this restarting procedure for many right-hand sides is given by:

$$\mu^m_{0,opt} = \mu^m_0 + \sum_{i=1}^{p^1+...+p^{m-1}} \frac{(b^m - C\mu^m_0, w_i)}{(Cw_i, w_i)} w_i. \tag{4.26}$$

In practice, the initial guess solution μ^m_0 is chosen equal to zero, which significantly simplifies the computation of the expression (4.26). Let us note that a full reorthogonalization of the search directions is necessary [Rou94] to ensure stability of the algorithm and to avoid computing the same search directions again:

$$w_{k+1} = g_{k+1} - \sum_{i=1}^{p^1+...+p^{m-1}+k} \gamma_i w_i \quad \text{with } \gamma_i = \sum_{i=1}^{p^1+...+p^{m-1}+k} \frac{(g_{k+1}, Cw_i)}{(Cw_i, w_i)} \tag{4.27}$$

5 Parametric Studies

The restarting techniques can be interpretated as a Krylov preconditioner, when the matrix C changes. This situation occurs when solving nonlinear problem [Rog93]

[Rou90]. We propose to use the Krylov preconditioner in another context. Suppose we have computed the eigenpairs of a structure, and that later a new system has to be solved coming from a small modification of this structure:

$$C^{mod}\mu = b. \tag{5.28}$$

The existing preconditioner P (Dirichlet or lumped preconditioner in the case of the dual Schur complement method) is then improved by the Krylov space which comes from the previous computation (p_1 search directions w and matrix-vector products Cw) with matrix C. It is given as follows:

$$P^{new}g_{k+1} = Pg_{k+1} + \sum_{i=1}^{p_1} \frac{(g_{k+1}, w_i) - (Pg_{k+1}, Cwi)}{(Cw_i, w_i)} w_i. \tag{5.29}$$

The technique gives good results if the spectrum of C and C^{mod} are close. Consequently the modification must be done far from the interface between substructures.

6 Numerical Results

We study a steel three-dimensional cantilever beam ($20m\times4m\times4m$). The finite element discretization is done with 6,400 hexahedral Q1-Lagrange elements (23,595 dof). The beam is cut in 8 slices ($1\times1\times8$), each substructure has 3,267 dof and the interface has 2,541 dof. Ten eigenpairs are required. The computation is carried out on 8 nodes of the Intel Paragon. Figure (1) shows the iteration history with different acceleration techniques. The restarting technique reduces dramatically the iteration number. We note that due to the particular decomposition (no cross points), the improvement is less important in case of the Dirichlet preconditioner. The three times appearing in the legend to the figure (1), correspond to the total CPU time, the time of the preparation step (assembly and factorization of the local stiffness matrix), and the time spent in the Lanczos and the dual Schur complement methods. In practice, the dual Schur complement method represents almost 80% of this last CPU time. The CPU times point out the best result for the lumped preconditioner. The Krylov preconditioner, figure (2) is tested on a structure 3% longer than the previous one (only one substructure has been modified). The reuse of the Krylov preconditioner reduces the CPU time (-20%). This is a useful numerical tool for parametric studies under conditions pointed out in section 5.

The beam is now box partitioned into 8 ($2\times2\times2$), 16 ($2\times2\times4$) and 32 ($2\times2\times8$) subdomains. Figure (3) and (4) show the numerical and parallel scalability of the method proposed. The CPU time is less for boxes decomposition than it was for slices because of the smaller bandwidth of the local problems.

The Schur dual complement method with coarse grid is insensitive to the number of subdomains. The accuracy of eigenpairs is governed by that of the linear system solution, which must be increased when many eigenpairs are sought.

Figure 1 Restarting technique

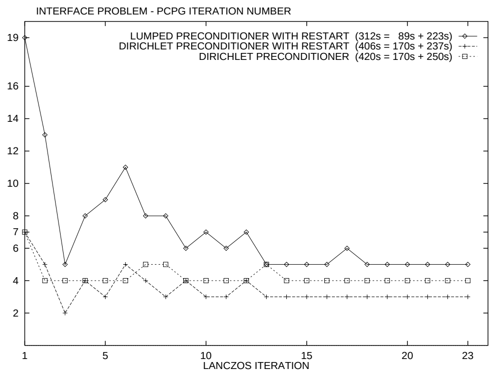


Figure 2 Krylov preconditioner

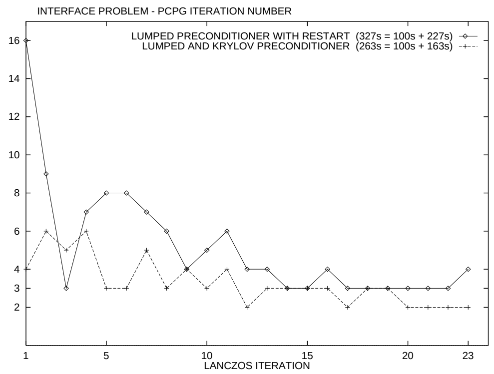


Figure 3 Numerical scalability

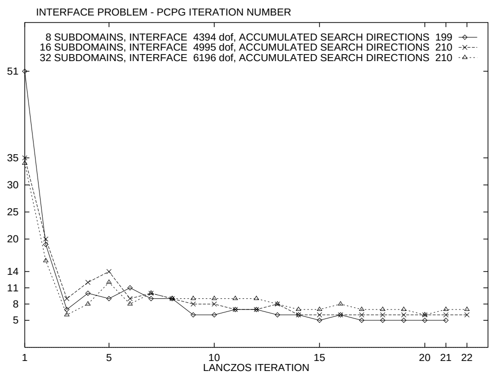
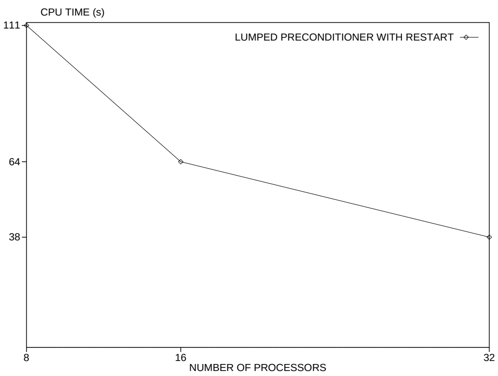


Figure 4 Parallel scalability



7 Conclusion

The method proposed has been tested with success on different examples [Cro97], especially an ill-conditioned problem (steel-elastomer structure) and presents good features for the parallel solution of large scale eigenvalue problems. It can be improved by including new developments in domain decomposition solvers. For the classical shift-and-invert approach $M(K - \sigma M)^{-1} Mq = \frac{1}{(\lambda - \sigma)} Mq$, in which σ is chosen close to the desired eigenvalue λ, a new coarse grid must be introduced, because there are no more floating subdomains. Finally, extension to nonsymmetric eigenproblem provides no difficulties.

REFERENCES

[BKP97] Bramble J., Knyazev A., and Pasciak J. (1997) A subspace preconditioning algorithm for eigenvectors/eigenvalue computation. *Advances in Computational Mathematics* to appear.

[CG82] Carnoy E. and Géradin M. (1982) On the practical use of the Lanczos algorithm in finite element applications to vibration and bifurcation problems. In *Matrix Pencils*, number 973 in Lectures Notes in Mathematics, pages 156–176. Springer Verlag. Proceedings, Pite Havsbad.

[Cro97] Cros J.-M. (to appear 1997) *Résolution de problèmes aux valeurs propres en* *calcul des structures par utilisation du calcul parallèle.* Doctoral dissertation, Ecole Normale Supérieure de Cachan.

[FC95] Farhat C. and Chen P.-S. (1995) Tailoring domain decomposition methods for efficient parallel coarse grid solution and for systems with many right hand sides. In Keyes and Xu [KX95], pages 401–406.

[FR94] Farhat C. and Roux F. (Juin 1994) *Implicit parallel processing in structural mechanics*, volume 2 of *Computational Mechanics Advances.* North-Holland.

[GLS86] Grimes R., Lewis G., and Simon D. (1986) Eigenvalue problems and algorithms in structural engineering. In Cullum J. and Willoughby R. (eds) *Large Scale Eigenvalue Problems.*

[GR94] Géradin M. and Rixen D. (1994) *Mechanical Vibrations.* Wiley.

[KX95] Keyes D. E. and Xu J. (eds) (1995) *Proc. Seventh Int. Conf. on Domain Decomposition Meths.* Number 180 in Contemporary Mathematics. AMS, Providence.

[LT90] Lascaux P. and Theodor R. (1990) *Analyse numérique matricielle appliquée à l'art de l'ingénieur*, volume 1. Masson.

[Rog93] Rogé C. (1993) *Méthodes de calcul de structures composites sur calculateur MIMD.* Doctoral dissertation, Université Paris 6.

[Rou90] Roux F.-X. (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.*, pages 314–319. SIAM, Philadelphia.

[Rou94] Roux F.-X. (1994) Parallel implementation of domain decomposition method for non-linear elasticity problems. In Keyes D. E., Saad Y., and Trulhar D. G. (eds) *Domain-based Parallelism and Problem Decomposition Methods in Computational Science and Engineering*, pages 161–175. SIAM, Minneapolis.

[RTD95] Roux F.-X. and Tromeur-Dervout D. (1995) Parallelization of a multigrid solver via a domain decomposition method. In Keyes and Xu [KX95], pages 439–444.

[Saa87] Saad Y. (1987) On the Lanczos method for solving symmetric linear systems with several right-hand sides. *Math. Comp.* 48(178): 651–662.

[SVdV96] Sleijpen G. and Van der Vorst H. (1996) A Jacobi-Davidson iteration method for linear eigenvalue. *SIAM J. Matrix Anal. Appl.* 17: 401–425.