

## Parallel Modal Synthesis Methods in Structural Dynamics

Jean-Michel Cros

### 1. Introduction

Within the finite element framework, we deal with large-scale eigenvalue problems induced by free vibration analysis of complex structures. The classical approach for the solution of such problems consists first of reducing the number of unknowns, allowing to reduce the computational cost of eigensolver, because only the lowest eigenfrequencies are classically researched in modal analysis. Component mode synthesis (CMS) or dynamic substructuring methods are appropriate tools for this reduction. In this paper we will discuss about the parallel implementation of CMS methods. We consider, in particular, among several CMS methods [4], the fixed-interface method which we briefly recall. Assuming that the studied domain is partitioned in  $N_s$  non-overlapping substructures, the global solution of the eigenvalue problem to be solved in the domain can be written as the sum of the local solutions (*fixed interface modes*) to elasticity eigenproblem to be solved in each subdomain clamped on the interface, and extensions (*constraint modes or coupling modes*), in each subdomain, of functions which represent the motion of the interface. The dynamic behavior of the global structure can be approximated in the low frequencies range by truncating the series which represent the different spaces. It remains to define what kind of Dirichlet's conditions has to be prescribed for the coupling of the substructures. The most classical choice [3] consists of prescribing at the discrete level the shape functions spanning the interfacial interpolation space. In this case, all the motions of the interface are represented, but, the number of *constraint modes* is equal to the number of degrees of freedom (d.o.f.) of the interface. From the parallel point of view the constraint and the fixed interface modes can be computed independently in each subdomain. An other choice, proposed by Bourquin [1], consists of filtering information in order to represent the interface's motion only in the low frequency range, thanks to a spectral decomposition of the interface operator coupling subdomains. However, the computation of the *coupling modes* involves all the subdomains. We propose a parallel implementation of these methods [5] thanks to the use of techniques arising in domain decomposition.

---

1991 *Mathematics Subject Classification*. Primary 73K12; Secondary 49R10, 65N25.

*Key words and phrases*. Eigenvalues, Modal Synthesis Methods, Domain Decomposition, Iterative Solver, Parallel Computing.

*Acknowledgements* - We would like to acknowledge O.N.E.R.A. for computing facilities.

The paper is organized as follows : Section 2 presents different parallel implementations of the Craig and Bampton (CB) method, one of them enabling to avoid the costly computation of the constraint modes. Section 3 describes some fixed interface methods using coupling modes. In Section 4, some numerical results obtained on the Intel PARAGON machine are presented. We also compare, from the cpu time point of view, the results obtained thanks to a parallel sparse eigensolver.

## 2. Parallel implementation of the Craig and Bampton method

The model reduction of each substructure ( $s$ ) is given by the projection of local mass ( $M^{(s)}$ ) and rigidity ( $K^{(s)}$ ) matrices onto the Ritz basis :

$$\begin{Bmatrix} u_i^{(s)} \\ u_b^{(s)} \end{Bmatrix} = [\Phi^{(s)} \quad \Psi^{(s)}] \begin{Bmatrix} \eta^{(s)} \\ u_b^{(s)} \end{Bmatrix} \text{ with } \Phi^{(s)} = \begin{bmatrix} \phi^{(s)} \\ 0 \end{bmatrix}, \Psi^{(s)} = \begin{bmatrix} -K_{ii}^{(s)-1} K_{ib}^{(s)} \\ I \end{bmatrix}$$

where the subscripts  $i$  and  $b$  respectively refer to the internal and boundary d.o.f.,  $\eta^{(s)}$  is the vector of normal modes intensities,  $\Psi^{(s)}$  and  $\Phi^{(s)}$  are respectively the constraint and fixed interface modes. By assembling the model reduction of each substructure we get the following reduced eigenproblem :

$$(1) \quad \tilde{K}z = \lambda \tilde{M}z$$

with the reduced rigidity ( $\tilde{K}$ ) and mass matrices ( $\tilde{M}$ ) given by :

$$\tilde{K} = \begin{bmatrix} \Omega_\alpha^{(s)^2} & \\ 0 & S \end{bmatrix}, \tilde{M} = \begin{bmatrix} I & \Phi^{(s)T} M^{(s)} \Psi^{(s)} \\ \Psi^{(s)T} M^{(s)} \Phi^{(s)} & \sum_{s=1}^{N_s} \Psi^{(s)T} M^{(s)} \Psi^{(s)} \end{bmatrix}$$

The reduced stiffness matrix includes Schur complement matrix ( $S$ ) and  $\Omega_\alpha^{(s)^2}$  which is a diagonal matrix storing the squares of the subdomains eigenvalues. The reduced mass matrix includes static condensed mass matrices and the inertia coupling between the constraint and fixed interface modes. The size of (1) is equal to the sum of the number of fixed interface modes, chosen in each subdomain, and the interface's number of d.o.f. (which in the case of complex structures is large). This "reduced" eigenproblem is still too large to take benefit from eigensolver such QR's method and claims the use of subspace algorithm (Lanczos, ...). The generalized eigenproblem is then reduced to a standard form and involves matrix-vector products with  $\tilde{K}^{-1}\tilde{M}$ . Different implementations have been studied in order to reduce the computational cost thanks to the use of parallel machine with distributed memory. The first one (*solver I*), is the most natural, because it uses the fact that fixed interface modes and constraint modes can be computed independently in each subdomain. Then, a processor is on charge of a subdomain and the local reduced matrices are built in parallel. However the reduced eigenproblem (1) is assembled and solved in sequential way. The second one (*solver II*), looks like Hybrid Craig and Bampton method, proposed by Farhat and G eradin [7], but consists in a primal formulation. The Schur complement matrix is not assembled and the action of  $S^{-1}$  on a vector is done by using iterative Schur complement method [8] with different acceleration techniques [5] (generalized Neumann-Neumann preconditioner, technique for taking into account multiple right hand sides, ...). The main task, consists in the computation of the constraint modes in order to build  $\tilde{M}$ . This operation may be avoid. For this purpose a third method (*solver III*) allows for the computation of matrix-vector products with reduced mass matrix in a implicit way. The details of this operation are as

follows. The product  $\tilde{M}^{(s)}z^{(s)}$  (with any vector  $z$ ) can be written in matrix form as :

$$(2) \quad \begin{aligned} & \begin{bmatrix} I & \Phi^{(s)T} M^{(s)} \Psi^{(s)} \\ \Psi^{(s)T} M^{(s)} \Phi^{(s)} & \Psi^{(s)T} M^{(s)} \Psi^{(s)} \end{bmatrix} \begin{Bmatrix} z_1^{(s)} \\ z_2^{(s)} \end{Bmatrix} \\ &= \begin{Bmatrix} z_1^{(s)} + \Phi^{(s)T} M^{(s)} \Psi^{(s)} z_2^{(s)} \\ \Psi^{(s)T} M^{(s)} \Phi^{(s)} z_1^{(s)} + \Psi^{(s)T} M^{(s)} \Psi^{(s)} z_2^{(s)} \end{Bmatrix} \end{aligned}$$

The description of the algebraic operations requires the knowledge of the different components of this vector, without any explicit assembling of the matrices  $[\Phi^{(s)T} M^{(s)} \Psi^{(s)}]$ ,  $[\Psi^{(s)T} M^{(s)} \Psi^{(s)}]$ ,  $[\Psi^{(s)T} M^{(s)} \Phi^{(s)}]$ . First of all, for each subdomain, the following matrix-vector product is computed :

$$(3) \quad \begin{bmatrix} M_{ii}^{(s)} & M_{ib}^{(s)} \\ M_{bi}^{(s)} & M_{bb}^{(s)} \end{bmatrix} \begin{bmatrix} \phi^{(s)} \\ 0 \end{bmatrix} = \begin{bmatrix} M_{ii}^{(s)} \phi^{(s)} \\ M_{bi}^{(s)} \phi^{(s)} \end{bmatrix}$$

Then, *at each iteration* of a given eigensolver, the operations described below are done :

Computation of  $\Psi^{(s)T} M^{(s)} \Phi^{(s)} z_1^{(s)}$

1. Compute  $M_{ii}^{(s)} \phi^{(s)} z_1^{(s)}$  with (3).
2. Solve Dirichlet's problem in  $\Omega^{(s)}$  with zero on  $\Gamma^{(s)}$  (interface with other subdomains) and external force :

$$(4) \quad \begin{bmatrix} K_{ii}^{(s)} & K_{ib}^{(s)} \\ K_{bi}^{(s)} & K_{bb}^{(s)} \end{bmatrix} \begin{Bmatrix} u_i^{(s)} \\ 0 \end{Bmatrix} = \begin{Bmatrix} M_{ii}^{(s)} \phi^{(s)} z_1^{(s)} \\ 0 \end{Bmatrix}$$

3. Compute the forces induced by the opposite displacement, solution of (4) :

$$(5) \quad \begin{bmatrix} K_{bi}^{(s)} & K_{bb}^{(s)} \end{bmatrix} \begin{Bmatrix} -K_{ii}^{(s)-1} M_{ii}^{(s)} \phi^{(s)} z_1^{(s)} \\ 0 \end{Bmatrix} = \begin{Bmatrix} -K_{bi}^{(s)} K_{ii}^{(s)-1} M_{ii}^{(s)} \phi^{(s)} z_1^{(s)} \\ \end{Bmatrix}$$

4. Compute  $M_{bi}^{(s)} \phi^{(s)} z_1^{(s)}$  with (3).
5. Assemble interface's contributions of vectors obtained at the two previous steps :

$$(6) \quad \left\{ M_{bi}^{(s)} \phi^{(s)} z_1^{(s)} - K_{bi}^{(s)} K_{ii}^{(s)-1} M_{ii}^{(s)} \phi^{(s)} z_1^{(s)} \right\}$$

Computation of  $\Psi^{(s)T} M^{(s)} \Psi^{(s)} z_2^{(s)}$

1. Solve Dirichlet's problem in  $\Omega^{(s)}$  with  $z_2^{(s)}$  on  $\Gamma^{(s)}$  :

$$(7) \quad \begin{bmatrix} K_{ii}^{(s)} & K_{ib}^{(s)} \\ K_{bi}^{(s)} & K_{bb}^{(s)} \end{bmatrix} \begin{Bmatrix} u_i^{(s)} \\ z_2^{(s)} \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}$$

2. Compute the matrix-vector product, the local mass matrix  $M^{(s)}$ , solution of problem (7) :

$$(8) \quad \begin{bmatrix} M_{ii}^{(s)} & M_{ib}^{(s)} \\ M_{bi}^{(s)} & M_{bb}^{(s)} \end{bmatrix} \begin{Bmatrix} -K_{ii}^{(s)-1} K_{ib}^{(s)} z_2^{(s)} \\ z_2^{(s)} \end{Bmatrix} = \begin{Bmatrix} (-M_{ii}^{(s)} K_{ii}^{(s)-1} K_{ib}^{(s)} + M_{ib}^{(s)}) z_2^{(s)} \\ (-M_{bi}^{(s)} K_{ii}^{(s)-1} K_{ib}^{(s)} + M_{bb}^{(s)}) z_2^{(s)} \end{Bmatrix}$$

3. Solve Dirichlet's problem in  $\Omega^{(s)}$  with zero on  $\Gamma^{(s)}$  and external forces :

$$(9) \quad \begin{bmatrix} K_{ii}^{(s)} & K_{ib}^{(s)} \\ K_{bi}^{(s)} & K_{bb}^{(s)} \end{bmatrix} \begin{Bmatrix} u_i^{(s)} \\ 0 \end{Bmatrix} = \begin{Bmatrix} (-M_{ii}^{(s)} K_{ii}^{(s)-1} K_{ib}^{(s)} + M_{ib}^{(s)}) z_2^{(s)} \\ 0 \end{Bmatrix}$$

4. Compute the forces induced by opposite displacement, solution of (9) :

$$(10) \quad \begin{aligned} & \left[ \begin{array}{cc} K_{bi}^{(s)} & K_{bb}^{(s)} \end{array} \right] \left\{ \begin{array}{c} (K_{ii}^{(s)-1} M_{ii}^{(s)} K_{ii}^{(s)-1} K_{ib}^{(s)} - K_{ii}^{(s)-1} M_{ib}^{(s)}) z_2^{(s)} \\ 0 \end{array} \right\} \\ & = \left\{ (K_{bi}^{(s)} K_{ii}^{(s)-1} M_{ii}^{(s)} K_{ii}^{(s)-1} K_{ib}^{(s)} - K_{bi}^{(s)} K_{ii}^{(s)-1} M_{ib}^{(s)}) z_2^{(s)} \right\} \end{aligned}$$

5. Assemble interface contributions of vectors (10) and (8). We get then :

$$\left\{ (M_{bb}^{(s)} - M_{bi}^{(s)} K_{ii}^{(s)-1} K_{ib}^{(s)} + K_{bi}^{(s)} K_{ii}^{(s)-1} M_{ii}^{(s)} K_{ii}^{(s)-1} K_{ib}^{(s)} - K_{bi}^{(s)} K_{ii}^{(s)-1} M_{ib}^{(s)}) z_2^{(s)} \right\}$$

**Computation of  $\Phi^{(s)T} M^{(s)} \Psi^{(s)} z_2^{(s)}$**

1. Computation of the forces arising from the solution of (7) :

$$(11) \quad \begin{aligned} & \left[ \begin{array}{cc} \phi^{(s)T} M_{ii}^{(s)} & \phi^{(s)T} M_{ib}^{(s)} \end{array} \right] \left\{ \begin{array}{c} -K_{ii}^{(s)-1} K_{ib}^{(s)} z_2^{(s)} \\ z_2^{(s)} \end{array} \right\} \\ & = \left\{ -\phi^{(s)T} M_{ii}^{(s)} K_{ii}^{(s)-1} K_{ib}^{(s)} z_2^{(s)} + \phi^{(s)T} M_{ib}^{(s)} z_2^{(s)} \right\} \end{aligned}$$

The reduced matrix-vector product is achieved through the assembling of the contributions of each subdomain. Let us note that all the matrices (except the mass matrix) are already used by the iterative Schur complement method.

**3. Fixed interface methods with coupling modes**

With regard to the previous method, a small set of modes is required for the coupling of the substructures. The new Ritz basis [1] is indeed defined by :

$$\left\{ \begin{array}{c} u_i^{(s)} \\ u_b^{(s)} \end{array} \right\} = [\Phi^{(s)} \quad \Psi^{(s)}] \left\{ \begin{array}{c} \eta^{(s)} \\ \xi \end{array} \right\} \text{ with } \psi^{(s)} = R^{(s)} \psi \text{ and } \Psi^{(s)} = \left[ \begin{array}{c} -K_{ii}^{(s)-1} K_{ib}^{(s)} \psi^{(s)} \\ \psi^{(s)} \end{array} \right]$$

where  $\psi$  are the first normal modes of the Schur complement matrix (interface modes),  $R^{(s)}$  is the restriction matrix of the global interface ( $\Gamma$ ) to the local interface ( $\Gamma^{(s)}$ ),  $\xi$  is the vector of the coupling modes intensities and  $\Psi^{(s)}$  defines the coupling modes. Thanks to this definition of coupling modes,  $\tilde{K}$  is diagonal, and the size of the reduced eigenproblem is equal to the sum of the number of fixed interface modes chosen in each subdomain and the number ( $N_\Gamma$ ) of coupling modes. Then, the solution of the reduced eigenproblem is easy. The difficulty is now to find the first normal modes of the interface's operator, in the present case the Schur complement matrix (*method 1*) with or without mass condensed matrix (denoted by  $B$ ).

$$(12) \quad Su_\Gamma = \lambda_\Gamma Bu_\Gamma \quad \rightarrow \quad S^{-1}Bu_\Gamma = \frac{1}{\lambda_\Gamma}u_\Gamma$$

From the computation view point, the method presents an interest, when the interface is large, only if the Schur complement is not assembled. Once again, the iterative Schur complement method ( $S$  is *not explicitly* inverted) is used. Various acceleration techniques have been used to reduce as best as possible the cost of this operation (Section 2). We have considered different mass matrices : identity, lumped ( $B = \sum_{s=1}^{N_s} M_{bb}^{(s)}$ ) and static condensed mass matrix. For the latter, the matrix-vector product can be done without any assembling operation (Section 2). Nevertheless, the computation of the first normal modes of the Schur complement is

a costly operation. The use of an approximated inverse [2] (*method 2*) of the Schur complement matrix (such that  $T \approx S^{-1}$ ) avoids to consider an iterative method, because only matrix-vector product with  $T$  are then required.

$$(13) \quad TBu_\Gamma = \frac{1}{\lambda_\Gamma} u_\Gamma$$

A good inverse of the Schur complement matrix is clearly the Neumann-Neumann preconditioner, written here in its basic form :

$$(14) \quad T = \sum_{s=1}^{N_s} R^{(s)T} S^{(s)-1} R^{(s)}$$

For sake of generality, it would be required to take into account decompositions with cross-points and floating subdomains. In the case of cross-points, [2] defines a new extension operator (reflection and cut-off function) which possess an interpretation at the continuous level. We propose to improve (14) by averaging the contributions of each subdomain through the introduction of weighting matrices [8]. On the other hand, the global problem can be well posed (no rigid body modes), but the decomposition may lead to local Neumann's problems not well posed (no Dirichlet's conditions). Shifting [2] the global problem enables to erase the floating subdomains, however one can also handle directly with them thanks to a filter operator which stems from balancing method [9]. Finally, the approximated inverse is given by :

$$(15) \quad T = (I - G(G^T S G)^{-1} G^T S) \sum_{s=1}^{N_s} R^{(s)T} P^{(s)} S^{(s)+} P^{(s)} R^{(s)}$$

where  $P^{(s)}$ , stands for the weighting matrix, defined such that  $\sum P^{(s)} = I_{/\Gamma}$ ,  $G$  is the rectangular matrix which stores the interface restriction of the local rigid body motions, and  $S^{(s)+}$  is the inverse of the projection of the image of  $S^{(s)}$  in  $\mathbb{R}^{N_{lfron(s)}}$ , where  $N_{lfron(s)}$  denotes the local interface size. Let us note, that the rigid body motions have to be also considered as coupling modes because, by construction, they are perpendicular to  $u_\Gamma$ .

#### 4. Numerical results

In order to present some numerical experiments, we consider a three-dimensional beam. The finite element discretization consists in 3000 hexahedral Q1-Lagrange elements (11253 d.o.f.); the mesh is cut in 8 boxes ( $2 \times 2 \times 2$ ); the size of the interface is thus large (2253 d.o.f.) and each substructure possess 1728 d.o.f. Ten eigenpairs are required. The computation is carried out on 8 nodes of the Intel Paragon.

Table 1 reports the cpu times of the main tasks of the CB's method. As we can see, with solver III, the cpu time compared with that of solver I is reduced of 65%. Concerning the method 1, we plot (Fig.1) the relative error committed on the global eigenfrequencies versus the number ( $N_\Gamma$ ) of coupling modes. We have compared the results for different mass matrices, used during the computation of the coupling modes. For static condensed mass matrix the improvement is obvious : with 10 coupling modes, the relative error is less than 1%. With lumped and identity mass the graphs are nearly the same, but more coupling modes are included in the approximation's space to get the same relative error. To understand these results,

TABLE 1. 3d beam (2-2-2) - cpu times of the parallel implementations of the Craig and Bampton method

	constraint modes		
	with solver I	with solver II	without solver III
constraint modes	80,2s	80,2s	∅
fixed interface modes	90,4s	90,4s	90,4s
assembling reduced eigenproblem	267,7s	133,9s	∅
solving reduced eigenproblem	70,1s	71,3s	86,8s
restitution	0,6s	0,6s	1,6s
total	508,8s	376,2s	177,2s

TABLE 2. 3d beam (2-2-2) - method 1 - coupling modes computing

Mass	Identity		Lumped		Static condensed
$N_\Gamma$	20	30	20	30	20
Lanczos	117,2s	167,2s	126,9s	176,7s	138,2s
Matrix (mass)-vector	0,007s	0,01s	7,9s	10,9s	24,7s

TABLE 3. 3d beam (2-2-2) - cpu times

solver III	method 1 ( $N_\Gamma = 15$ )	method 2 ( $N_\Gamma = 25$ )	parallel sparse eigensolver
177s	190s	130s	97s

we plot (Fig. 2) the spectrum of the Schur complement matrix with different mass matrices along the interface. As we can see, the spectrum of the Schur complement, when static condensed mass is used, is very closed to this of the global structure. Then, knowing the excitation's range of the global structure it seems possible to define a by-passing frequency criterion, allowing to select enough coupling modes during their computations to get a good approximation of the global eigenvalues.

Table 2 reports the cpu times corresponding to the main operation (method 1) : the computation of the coupling modes. Taking into account the static condensed mass is not too expensive. Despite the use of acceleration techniques (Fig. 3) (such as coarse grid induced by rigid modes, preconditioner, techniques enabling to take into account multiple right hand sides), the computation of the coupling modes remains a cpu times consuming operation.

Now we compare (Table 3) the cpu times required to find 10 eigenpairs with different techniques. From the cpu times point of view, despite the different implementations proposed, in particular for the CB's method, parallel sparse eigensolver [6], based on Lanczos algorithm and nonoverlapping domain decomposition method, is the fastest method. This result was predictable because in the three methods (solver III, method 1 and parallel sparse eigensolver) the same interface problem (Fig. 3) is solved, and CMS needs additional operations (such that fixed interface modes computation, restitution of the solution in each subdomain). In addition, parallel sparse eigensolver provides the best approximation of the eigenpairs. By another way, for all methods, parallel scalability [6] is guaranteed thanks to the iterative substructuring method, if load balancing is correct.

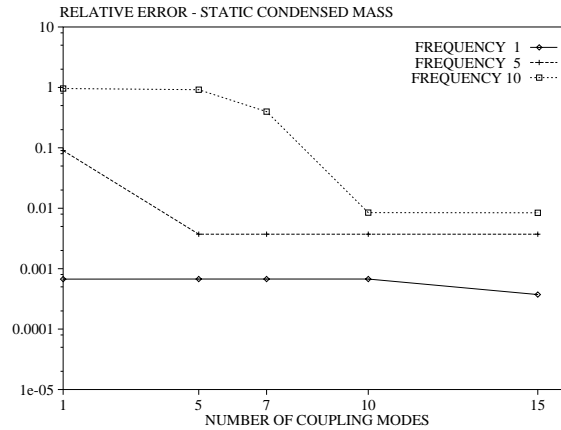


FIGURE 1. Error for beam 3d, method 1,  $N_{\Gamma}$  increasing

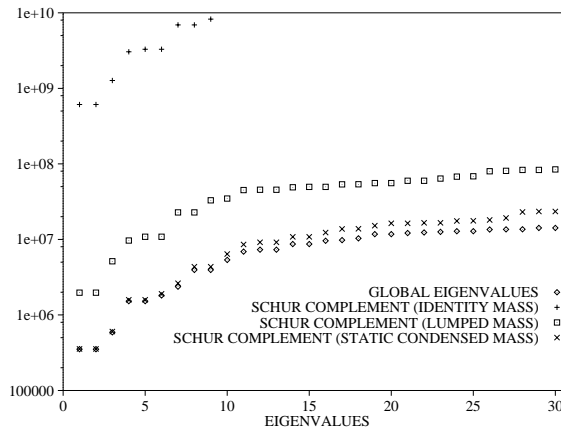


FIGURE 2. Beam 3d, eigenspectrum of the Schur complement with different mass matrices

By the way, method 2 is shown to be less accurate (with 25 coupling modes the relative error is less than 3%) than method 1 (see also [2, 10]), but quickly gives an approximation of the global eigenpairs with regard to the others CMS.

In conclusion, the methods presented in this paper are particularly well suited to the architecture of parallel computers. The different methods can be improved in order to take into account other finite elements (plates, shells). A comparison with other methods using coupling modes [2, 10] seems necessary. Let us lastly note that particular methods can be derived for special cases (decomposition without cross-point).

## References

1. F. Bourquin and F. d'Hennezel, *Application of domain decomposition techniques to modal synthesis for eigenvalue problems*, Proc. Fifth Int. Conf. on Domain Decomposition Meths. (Philadelphia) (D. E. Keyes, T. F. Chan, G. A. Meurant, J. S. Scroggs, and R. G. Voigt, eds.), 1992.

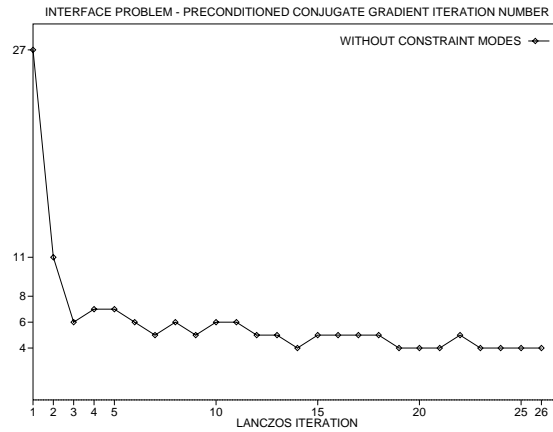


FIGURE 3. Beam 3d, solver III, solution of the reduced eigenvalue problem

2. F. Bourquin and R. Namar, *Decoupling and modal synthesis of vibrating continuous systems*, Proc. Ninth Int. Conf. on Domain Decomposition Meths. (P. Bjørstad, M. Espedal, and D. Keyes, eds.), Wiley and Sons, 1996.
3. R.R. Craig and M.C.C. Bampton, *Coupling of substructures for dynamic analysis*, AIAA Journal **6** (1968), 1313–1319.
4. R.R. Craig, Jr, *A review of time domain decomposition and frequency domain component mode synthesis methods*, Combined Experimental/Analytical Modeling of Dynamic Structural systems, vol. 67, ASME, 1985.
5. J.M. Cros, *Résolution de problèmes aux valeurs propres en calcul des structures par utilisation du calcul parallèle*, Ph.D. thesis, Ecole normale supérieure de Cachan, 1997.
6. J.M. Cros and F. Léné, *Parallel iterative methods to solve large-scale eigenvalue problems in structural dynamics*, Proc. Ninth Int. Conf. on Domain Decomposition Meths. (Bergen), 1996.
7. C. Farhat and M. Géradin, *On a component mode synthesis method and its application to incompatible substructures*, Computer and Structures **51** (1994), 459–473.
8. P. Le Tallec, *Domain decomposition methods in computational mechanics*, Computational Mechanics Advances, no. 2, North-Holland, 1994.
9. J. Mandel, *Balancing domain decomposition*, Comm. Numer. Meth. Engrg. **9** (1993), 233–241.
10. D. Rixen, *Substructuring and dual methods in structural analysis*, Ph.D. thesis, Université de Liège, 1997.

LABORATOIRE DE MODÉLISATION ET MÉCANIQUE DES STRUCTURES, UNIVERSITÉ PIERRE ET MARIE CURIE, 4 PLACE JUSSIEU, F75252 PARIS CEDEX 05

*E-mail address:* cros@ccr.jussieu.fr