

Preconditioning Subspace Iteration for Large Eigenvalue Problems with Automated Multi-Level Sub-structuring

Heinrich Voss ^{1,*} and Jiacong Yin ² and Pu Chen ²

¹ *Institute of Mathematics, Hamburg University of Technology, voss@tuhh.de*

² *State Key Laboratory for Turbulence and Complex Systems & Department of Mechanics and Aerospace Engineering, College of Engineering, Peking University, {jcyin, chenpu}@pku.edu.cn*

SUMMARY

The subspace iteration method (SIM) is a numerical procedure for normal mode analysis which has shown to be robust and reliable for solving very large general eigenvalue problems. Although its classical form as introduced by Bathe in the seventies of the last century is less efficient than the Lanczos iteration method in terms of CPU time, it is beneficial in terms of storage use if a very large number (say hundreds) of eigenmodes are needed and good approximations to the wanted eigenvectors are at hand. In this paper we take advantage of the automated multi-level sub-structuring (AMLS) to construct an accurate initial subspace for SIM. Along with the AMLS reduction we derive a very efficient preconditioning method for SIM which solves the linear systems for a transformed system with block diagonal system matrix whereas the multiplication with the mass matrix is executed in the original variables.

Copyright © 2010 John Wiley & Sons, Ltd.

KEY WORDS: Large eigenvalue problem, subspace iteration, preconditioning, initial approximation, automated multi-level sub-structuring

Dedicated to Ivo Marek on the occasion of his 80th birthday

1. Introduction

In this paper we consider the problem to determine all eigenvalues $0 < \lambda_1 \leq \lambda_2 \leq \dots$ not exceeding a given value λ_{\max} and the corresponding eigenvectors of a generalized eigenvalue problem

$$Kx = \lambda Mx \quad (1.1)$$

where $K \in \mathbb{C}^{n \times n}$ and $M \in \mathbb{C}^{n \times n}$ are Hermitian and positive definite matrices and are very large and sparse, e.g. stiffness and mass matrices of a finite element model.

*Correspondence to: Heinrich Voss, Hamburg University of Technology, Institute of Mathematics, D-21073 Hamburg, Germany

One of robust and efficient methods for solving huge generalized eigenvalue problems in structural dynamics is the subspace iteration method. Generalizing the simultaneous iteration method of Bauer [6] and Rutishauser [22] it was proposed and so named in the 1970s by Bathe [5] for computing eigenfrequencies and mode shapes of structures such as buildings and bridges using finite element discretizations. At that time, typically only few frequencies and mode shapes were needed, such as the lowest 10 to 20 eigenpairs in models containing 1000 to 10,000 degrees of freedom. Since its development, however, the subspace iteration method has been widely used for considerably larger systems reaching millions of degrees of freedom and for computing hundreds of eigenpairs.

The most expensive parts of the k th iteration step of subspace iteration for computing the p smallest eigenvalues are the solution of a linear system of equations

$$KZ_{k+1} = MX_k \quad (1.2)$$

and an orthogonalization of the columns of Z_{k+1} to receive the next iterate X_{k+1} . Here $X_0 \in \mathbb{C}^{n \times q}$, $q \geq p$ is an initial matrix, such that the span of its columns contains good approximations to the wanted eigenvectors. So, key steps in applying the subspace iteration method to huge eigenvalue problems is to establish an efficient initial matrix X_0 (including its number of columns) and to solve the linear systems (1.2) efficiently.

For both tasks we propose to take advantage of the automated multi-level sub-structuring (AMLS) introduced by Bennighof [7, 8, 9], which is capable to determine a large number of eigenpairs of a huge problem with moderate accuracy in structural analysis. In AMLS the large finite element model is recursively divided into very many sub-structures on several levels based on the sparsity structure of the system matrices. Assuming that the interior degrees of freedom of sub-structures depend quasistatically on the interface degrees of freedom, and modeling the deviation from quasistatic dependence in terms of a small number of selected sub-structure eigenmodes the size of the finite element model is reduced substantially yet yielding satisfactory accuracy over a wide frequency range of interest. Hence, we obtain a reasonable initial matrix X_0 .

Concurrently, we are able extract in the AMLS reduction an eigenvalue problem

$$\hat{K}\hat{x} = \lambda\hat{M}\hat{x}, \quad \hat{K} = U^H K U, \quad \hat{M} = U^H M U \quad (1.3)$$

which is equivalent to the original eigenproblem (1.1), where the transformed stiffness matrix \hat{K} becomes block-diagonal with small block sizes such that linear systems with system matrix \hat{K} can be solved very efficiently.

Unfortunately, the transformed mass matrix \hat{M} contains many dense submatrices requiring a huge amount of storage such that it is very expensive to evaluate the right hand side of the transformed system

$$\hat{K}\hat{Z}_{k+1} = \hat{M}\hat{X}_k \quad (1.4)$$

correspondent to (1.2). The way out is to combine the benefits of both approaches, i.e. to apply subspace iteration to the transformed system (1.4), but to evaluate its right hand side taking advantage of the transformation matrix U and the sparse structure of the original mass matrix M . Thus, since the matrix \hat{K} is block diagonal with small block sizes the effort of solving the linear system (1.2) in every iteration step is reduced considerably, AMLS can be considered as a preconditioner and the whole approach is a preconditioned subspace iteration.

Our paper is organized as follows. Sections 2 and 3 review the essential properties of the subspace iteration method and the automated multi-level sub-structuring method. Section

4 proposes the combination of both methods, and Section 5 contains a numerical example demonstrating the efficiency of our approach. The paper closes with conclusions in Section 6.

2. Subspace iteration method

The basic subspace iteration method for determining p eigenvalues corresponding to the p smallest eigenvalues $\lambda_1 \leq \dots \leq \lambda_p$ of problem (1.1) reads as follows:

Algorithm 1.

Input: initial matrix $X_0 \in \mathbb{C}^{n \times q}$, $q \geq p$

- 1: **for** $k = 1, 2, \dots$ **do**
- 2: inverse iteration step: solve $KZ_k = MX_{k-1}$ for Z_k
- 3: compute projected stiffness and mass matrix $K_k := Z_k^H K Z_k$ and $M_k := Z_k^H M Z_k$
- 4: solve projected eigenproblem $K_k Q_k = M_k Q_k \Lambda_k$ for Q_k and Λ_k
- 5: compute next iteration matrix $X_k = Z_k Q_k$
- 6: check for convergence
- 7: **end for**
- 8: perform Sturm sequence check

SIM was shown to converge linearly by Bathe [1] where the convergence rate for the i smallest eigenvalue λ_i is

$$\lambda_i / \lambda_{q+1}, \quad i = 1, \dots, q. \quad (2.1)$$

A Sturm sequence check can be used to show that all eigenvalues in an interval have been found, and the convergence can be checked using the following error estimation: Assume that the diagonal elements $\lambda_i^{(k)}$ of Λ_k are arranged in increasing order and let $q_i^{(k)}$ be the i th column of Q_k , then it holds

$$\min_j \left| \frac{\lambda_j - \lambda_j^{(k)}}{\lambda_j} \right| \leq \left\{ 1 - \frac{(\lambda_i^{(k)})^2}{(q_i^{(k)})^H q_i^{(k)}} \right\}^{1/2}.$$

Many acceleration techniques are reported in the literature such as shifts [4], and aggressive shifts [28], using refined Ritz vectors [16], combining SIM with Chebyshev polynomials [26], selective repeated inverse iteration [19], SIM in conjunction with sub-structure techniques [21] and multi-level sub-structuring [20, 25], to name just a few.

Using the subspace iteration method, it is crucial to establish an effective starting matrix X_0 , considering the quality of the approximation properties of its columns, and the number q of vectors, and to solve the linear systems (1.2) efficiently. In re-analysis or optimization problems of structures the eigenvectors computed with the last parameter setting are usually good initial vectors, in general they can be constructed by Guyan's reduction or reduced quadratic eigenproblems [10].

In the original development of the subspace iteration procedure based on the experience on seeking a small number of eigenpairs Bathe [2] suggested to use $p := \min\{p + 8, 2p\}$ initial vectors. But with large computer memory available at present, a much larger number of iteration vectors can now be used efficiently. Based on a model for the distribution of eigenvalues, i.e. assuming that $\lambda_j - \lambda_1$ is a known polynomial of j of degree less than or equal to 4, Bathe [3] developed a formula for an optimal q , and if the explicit form of the polynomial

is not known he suggested to use the very simple formula $q := \max\{p + 8, 2p\}$, where $p + 8$ is added to have enough iteration vectors when p is small (for a linear distribution of eigenvalues a formula is derived in [23]).

Obviously, $\text{span}(Z_k) = \text{span}(X_k)$ for every k , and therefore in exact arithmetic SIM is equivalent to the simultaneous iteration

$$KX_k = MX_{k-1}, \quad k = 1, 2, \dots, m \quad (2.2)$$

and subsequent projection of (1.1) to $\text{span}(X_m)$ to extract the wanted eigeninformation.

But in real arithmetic all columns of X_k converge to an eigenvector corresponding to the smallest eigenvalue, and the columns of X_k become a poorer and poorer basis of $\text{span}(X_k)$. The Ritz analysis in steps 3. and 4. and the basis transformation in step 5. of Algorithm 1 remedy this deficiency.

However, if the number q of initial vectors is much larger than the number p of wanted eigenpairs then the rate of convergence λ_j/λ_{q+1} , $j = 1, \dots, p$ usually will be quite small, and a small number of iteration steps will yield a sufficient accuracy, in particular if the initial basis X_0 contains good approximations to the wanted eigenvectors. Then the Ritz analysis can be postponed to the end of the inverse iteration, and the following simplified SIM results:

Algorithm 2.

Input: initial matrix $X_0 \in \mathbb{C}^{n \times q}$, $q > p$

- 1: **for** $k = 1, 2, \dots, m$ **do**
- 2: solve $KX_k = MX_{k-1}$ for X_k
- 3: **end for**
- 4: compute projected stiffness and mass matrix $K_m := X_m^H K X_m$ and $M_m := X_m^H M X_m$
- 5: solve projected eigenproblem $K_m Q_m = M_m Q_m \Lambda_m$ for Q_m and Λ_m
- 6: check for convergence
- 7: perform Sturm sequence check

3. Automated Multi-Level Sub-structuring

In this section we summarize the AMLS method for computing eigenvalues and corresponding eigenvectors of the linear eigenvalue problem (1.1) in a frequency range of interest. Similarly as in the component mode synthesis method (CMS) [11, 15] the structure is partitioned into a small number of sub-structures based on the sparsity pattern of the system matrices, but more generally than in CMS these sub-structures in turn are sub-structured on a number of levels yielding a tree topology for the sub-structures. Figure 1 shows the graph of an example where each parent sub-structure has at most two children sub-structures. Other sub-structure trees are possible and are in use.

AMLS consists of two ingredients. First, based on the sub-structuring the stiffness matrix K is transformed to block diagonal form by Gaussian elimination, and secondly, the dimension is reduced substantially by modal condensation of the sub-structures.

We describe a typical reduction step of AMLS. After reordering the degrees of freedom (this is assumed only for explanation; in an implementation the reordering is incorporated

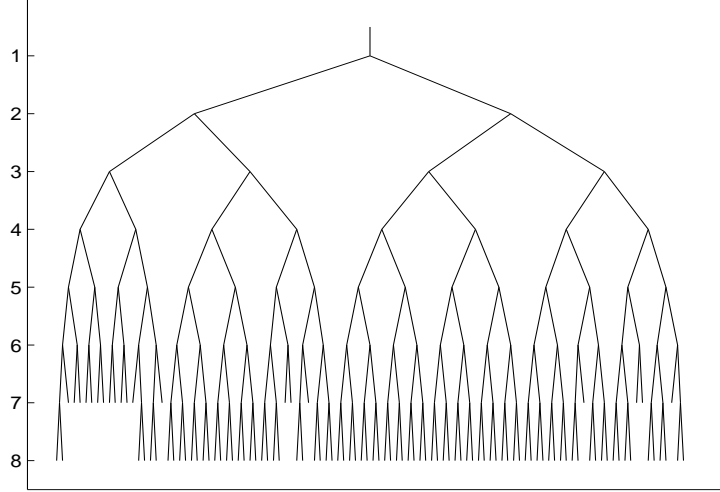


Figure 1. Sub-structure tree

implicitly) the current reduced problem obtains the following form

$$\begin{bmatrix} K_p & O & O \\ O & K_c & K_{cr} \\ O & K_{rc} & K_r \end{bmatrix} \begin{bmatrix} x_p \\ x_c \\ x_r \end{bmatrix} = \lambda \begin{bmatrix} M_p & M_{pc} & M_{pr} \\ M_{cp} & M_c & M_{cr} \\ M_{rp} & M_{rc} & M_r \end{bmatrix} \begin{bmatrix} x_p \\ x_c \\ x_r \end{bmatrix} \quad (3.1)$$

where x_p denotes the degrees of freedom which were already obtained in previous reduction steps, x_c are the degrees of freedom which are treated in the current step of AMLS, and x_r collects the ones to be handled in remaining steps of the algorithm. Notice, that x_p is empty in the first step, and x_c corresponds to unknowns of one sub-structure (in the first step of a leaf of the sub-structure tree). So, the dimensions of K_c and M_c are usually quite small.

Applying the variable transformation

$$\begin{bmatrix} x_p \\ x_c \\ x_r \end{bmatrix} = \begin{bmatrix} I & O & O \\ O & I & -K_c^{-1}K_{cr} \\ O & O & I \end{bmatrix} \begin{bmatrix} x_p \\ \tilde{x}_c \\ x_r \end{bmatrix} =: U_c \tilde{x} \quad (3.2)$$

and multiplying with U_c^H from the left to retain the symmetry of the eigenvalue problem, the degrees of freedom of the current sub-structure are decoupled from the remaining ones in the stiffness matrix. Thus (3.1) obtains the following form:

$$\begin{bmatrix} K_p & O & O \\ O & K_c & O \\ O & O & \tilde{K}_r \end{bmatrix} \begin{bmatrix} x_p \\ \tilde{x}_c \\ x_r \end{bmatrix} = \lambda \begin{bmatrix} M_p & M_{pc} & \tilde{M}_{pr} \\ \tilde{M}_{cp} & M_c & \tilde{M}_{cr} \\ \tilde{M}_{rp} & \tilde{M}_{rc} & \tilde{M}_r \end{bmatrix} \begin{bmatrix} x_p \\ \tilde{x}_c \\ x_r \end{bmatrix}, \quad (3.3)$$

where

$$\begin{aligned}\tilde{K}_r &= K_r - K_{rc}K_c^{-1}K_{cr}, \\ \tilde{M}_r &= M_r - M_{rc}K_c^{-1}K_{cr} - K_{rc}K_c^{-1}M_{cr} + K_{rc}K_c^{-1}M_cK_c^{-1}K_{cr} \\ \tilde{M}_{pr} &= M_{pr} - M_{pc}K_c^{-1}K_{cr} = \tilde{M}_{rp}^H \\ \tilde{M}_{cr} &= M_{cr} - M_cK_c^{-1}K_{cr} = \tilde{M}_{rc}^H.\end{aligned}$$

Recall that most of the columns of K_{cr} are null vectors. Only those columns of K_{cr} contain non-zero components the corresponding degrees of freedom of which lie on the boundary of the current sub-structure. Hence, $K_c^{-1}K_{cr}$ can be computed very efficiently since only a small linear system with a very small number of right hand sides has to be solved.

Next the dimension of the problem is reduced by modal condensation of the current level. To this end the eigenvalue problem

$$K_c\phi_j = \omega_j M_c\phi_c, \quad \phi_c^H M_c\phi_j = \delta_{ij} \tag{3.4}$$

is solved, and problem (3.3) is projected to the space spanned by the columns of $Z_c = \text{tridiag}[I_p, P_c, I_r]$, where $\dim I_p = \dim K_p$, $\dim I_r = \dim K_r$ and the columns of P_c are the eigenmodes of problem (3.4) corresponding to eigenvalues ω_j not exceeding the cut-off frequency ω_{cutoff} . Again this step is very inexpensive because of the small dimension of K_c and M_c .

After we have passed through all sub-structures we finally arrive at the reduced eigenvalue problem

$$\tilde{K}\tilde{x} = \lambda\tilde{M}\tilde{x}, \tag{3.5}$$

where \tilde{K} is a diagonal matrix.

The description of AMLS above is valid for any arrangement of the sub-structures, but handling the sub-structures in an appropriate order is most important to ensure computational efficiency.

The transformed matrix (3.3) demonstrates that the decoupling of a sub-structure modifies only the matrices of sub-structures which are connected to it. Hence, to keep the storage needed for the transformed mass matrix as small as possible one should start the elimination at the leaves of the partition tree, and as soon as all sub-structures which are connected to an interface on the superior level have been reduced the interface should be decoupled from all other sub-structures in the stiffness matrix and should be reduced as well.

More generally, the symmetric Gaussian eliminations and the condensations are performed moving through the partition tree on a postorder traversal, i.e. beginning on the finest level visiting all tree nodes where a sub-structure is decoupled and reduced as soon as all of its children have been reduced. In this interleaving way one avoids the storage of large dense sub-matrices of the transformed mass matrix.

We already mentioned that for any selection of the ordering the reduced matrix \tilde{K} becomes diagonal, the mass matrix is reduced to a generalized arrow-head form. Details of an implementation of AMLS are contained in [12, 14, 17, 27].

4. Preconditioning SIM with AMLS

Obviously, the modal reduction of problem (3.3) in AMLS does not modify the block matrices \tilde{K}_r and \tilde{M}_r but only those blocks of both matrices which are handled currently or have been

modified previously. Hence, we arrive at the same reduced problem if we first apply the variable transformations with (3.2) on all sub-structures, and then in a second step all modal reductions. Hence, if $U := U_1 U_2 \dots U_m$ denotes the product of all transformations U_c in (3.2), and if $Z := Z_1 Z_2 \dots Z_m$ denotes the product of all reduction matrices Z_c , then

$$\tilde{K} = Z^H \hat{K} Z := Z^H U^H K U Z \quad \text{and} \quad \tilde{M} = Z^H \hat{M} H := Z^H U^H M U Z.$$

Hence, the eigenvalue problem (1.1) is equivalent to

$$\hat{K} \hat{x} = \lambda \hat{M} \hat{x}, \quad \hat{K} := U^H K U, \quad \hat{M} := U^H M U, \quad U := U_1 U_2 \dots U_m \quad (4.1)$$

and \hat{K} is a block diagonal matrix with very small blocks on its diagonal, such that linear systems with system matrix \hat{K} can be solved very efficiently.

Unfortunately, the block matrix \hat{M} contains very many dense sub-matrices. Hence, along with the AMLS reduction the transformed matrix \hat{K} can be computed and stored cheaply, whereas the storage of \hat{M} has to be avoided because it would require a huge amount of storage. So, SIM for the transformed problem (4.1) is also not efficient.

The way out is to combine the benefits of both approaches, i.e. to apply subspace iteration to the transformed problem (4.1), but to evaluate $\hat{M}V$ taking advantage of the transformation matrix U and the sparse structure of the original mass matrix M .

We already pointed out that crucial for the success of SIM is the quality of the initial matrix and the number of its columns. From our experience [13, 27] we can deduce the following rule of thumb: if all eigenvalues not exceeding λ_{\max} are wanted, then AMLS with cut-off frequency $\omega_{\text{cutoff}} = 5 \cdot \lambda_{\max}$ usually yields an approximation to λ_{\max} with a relative error of about 10% and (much) smaller errors for the (much) smaller eigenvalues. Hence, from the result of AMLS with this cut-off frequency we can read off p as the number of eigenvalue approximations which are less than $1.1 \cdot \lambda_{\max}$ and according to Bathe's recommendation we choose $q = 2p$.

Algorithm 3.

Input: p and q as explained above,
the transformed eigenvectors \hat{V} ,
the transformed stiffness matrix \hat{K} ,
and the transformation matrix U from AMLS

- 1: initialize the iteration matrices $\hat{Q}_0 = \hat{V}$
- 2: **for** $k = 1, 2, \dots, m$ **do**
- 3: transform backward $Q_{k-1} = U \hat{Q}_{k-1}$
- 4: compute $R = M Q_{k-1}$
- 5: transform forward $\hat{R} = U^H R$
- 6: solve for \hat{Q}_k : $\hat{K} \hat{Q}_k = \hat{R}$
- 7: **end for**
- 8: project transformed stiffness matrix $\hat{K}_c = \hat{R}^H \hat{Q}_m$
- 9: transform backward $Q_m = U \hat{Q}_m$
- 10: project transformed mass matrix $\hat{M}_c = (Q_m)^H M Q_m$
- 11: solve projected problem $\hat{K}_c X_c = \hat{M}_c X_c \Lambda$
- 12: compute eigenvector approximations $V_m = Q_m X_c$.

Table I. Computation time of SIM and SIM-AMLS

AMLS reduction	100.2s
AMLS eigenvectors	7.6s
SIM-AMLS	196.8s
normal SIM	501.0s

5. Numerical Example

To evaluate the the performance of SIM preconditioned with AMLS we consider a finite element model of a blade of a 1.5 MW wind turbine with 117990 degrees of freedom, where the number of non-zeros in the stiffness matrix is $n_z(K) = 11243248$ and in the mass matrix is $n_z(M) = 5590256$. We are interested in all eigenvalues which are less than $2 \cdot 10^5$ (rad/s)².

An AMLS run with cut-off frequency $\omega_{\text{cutoff}} = 1 \cdot 10^6$ (rad/s)² suggests to choose $p = 175$ and to include $q = 350$ trial eigenmodes into our computations. The 350th eigenvalue of (1.1) is $\lambda_{350} \approx 6 \cdot 10^5$ (rad/s)², and therefore the convergence rate for the wanted eigenvalues is at most 1/3.

The computations were performed on a 64-bit Linux platform with an Intel Pentium D CPU (3.64 GHz, 2 Cores) and 7.7 GB memory. AMLS were coded in C employing METIS [18] to fix the sub-structuring and the Intel Math Kernel Library v10.3 optimized LAPACK to achieve high performance of computations.

We applied 3 steps of SIM. Table I compares the computing times of SIM for problem (1.1) and SIM-AMLS. The AMLS reduction and the computation of the eigenvector approximations by AMLS provide the initial approximation of eigenvectors, and are needed for both methods. Taking advantage of the AMLS information (i.e. the block diagonal \hat{K} , the transformation U , and the sparse structure of the original M) the computing time for SIM is reduced to about 40% by the preconditioning.

Figure 2 shows the relative errors of the eigenvalue approximations by AMLS und of three steps of SIM-AMLS demonstrating that the maximal relative error for all wanted eigenvalues not exceeding $2 \cdot 10^5$ (rad/s)² is reduced from $1.8 \cdot 10^{-2}$ for the sole AMLS method with cut off frequency $\omega_{\text{cutoff}} = 1 \cdot 10^6$ (rad/s)² in the first SIM step to $2.3 \cdot 10^{-3}$, in the second one to $1.4 \cdot 10^{-4}$, and in the third step to $1.1 \cdot 10^{-5}$. Notice that the approximate eigenvalues used as reference values were obtained with the block Lanczos package BLZPACK in double precision, but were on our disposal only in single precision. Hence, the given relative errors in Figure 2 in the order $1 \cdot 10^{-8}$ are essentially rounding errors of the reference values.

To measure the accuracy of the computed eigenvector approximations we use the modal error [2]

$$\varepsilon = \frac{\|Kx - \lambda Mx\|}{\|\lambda Mx\|}$$

of the eigenpairs (λ, x) . Figure 3 shows the modal errors of the wanted eigenpairs for the AMLS approximations and for the first three steps of SIM-AMLS. It demonstrates that the modal error for the sole AMLS approximation is about 10. But in structural dynamics, the modal error is required to be as low as 10^{-3} . Otherwise, no accurate strain or stress can be derived.

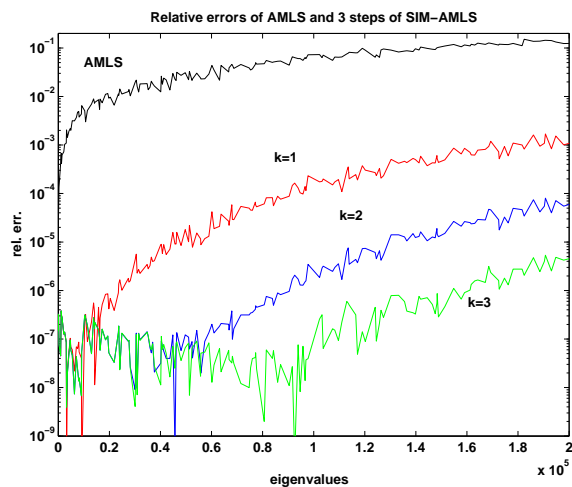


Figure 2. Relative errors of eigenvalue approximations

By three steps of SIM-AMLS this bound can be achieved for the smallest 125 eigenvalues, i.e. for all eigenvalues not exceeding $1.25 \cdot 10^5$.

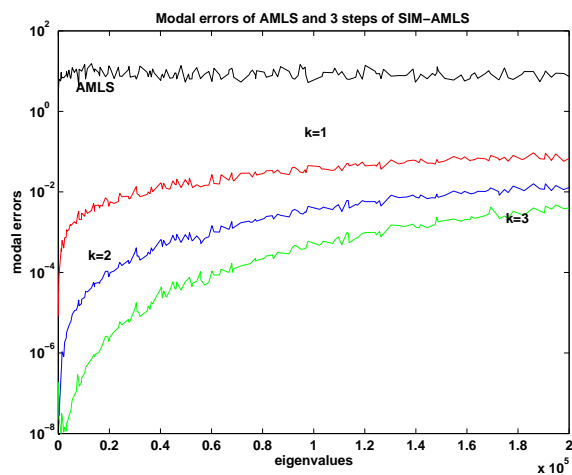


Figure 3. Modal errors of eigenpair approximations

6. Conclusions

The object of this paper is to combine the subspace iteration method (SIM) and the automated multi-level sub-structuring (AMLS) to derive a robust and efficient method for determining a

large number of eigenpairs at the lower end of the spectrum of a huge generalized eigenvalue problem. On the one hand, AMLS is used to determine an accurate initial basis for SIM. Moreover, the AMLS reduction yields a sequence of congruence transformations and a block diagonal matrix which is congruent to the mass matrix such that SIM can be performed in the transformed space whereas the right hand side is evaluated by multiplying the current iterate with the mass matrix in the original space. The efficiency is demonstrated by a finite element model of a blade of a rotor. A further example for a gyroscopic quadratic eigenvalue problem governing the eigenmodes of a rotating tire can be found [24].

Acknowledgement

This work was supported by the National Natural Science Foundation of China under the grant No.10972005 and the China Scholarship Council with the contract No.2010601199.

REFERENCES

1. K.J. Bathe. Formulations and Computational Algorithms in Finite Element Analysis. MIT Press, Cambridge, MA, 1977.
2. K.J. Bathe. Finite Element Procedures. Prentice-Hall, New Jersey, 1996.
3. K.J. Bathe. The subspace iteration method – revisited. Computers & Structures, 126:177 – 183, 2013.
4. K.J. Bathe and J. Ramaswamy. An accelerated subspace iteration method. Comp. Meth. Appl. Mech. Engrg., 23:313 – 331, 1980.
5. K.J. Bathe and E.L. Wilson. Large eigenvalue problems in dynamic analysis. J. Engrg. Mech. Div., 98:1471 – 1485, 1972.
6. F.L. Bauer. Das Verfahren der Treppeniteration und verwandte Verfahren zur Lösung algebraischer Eigenwertprobleme. Zeitschrift für Angewandte Mathematik und Mechanik, 37:214–235, 1957.
7. J.K. Bennighof. Adaptive multi-level substructuring method for acoustic radiation and scattering from complex structures. In A.J. Kalinowski, editor, Computational methods for Fluid/Structure Interaction, volume 178, pages 25 – 38, New York, 1993. ASME.
8. J.K. Bennighof and C.K. Kim. An adaptive multi-level substructuring method for efficient modeling of complex structures. In Proceedings of the AIAA 33rd SDM Conference, pages 1631 – 1639, Dallas, Texas, 1992.
9. J.K. Bennighof and R.B. Lehoucq. An automated multilevel substructuring method for the eigenspace computation in linear elastodynamics. SIAM J. Sci. Comput., 25:2084 – 2106, 2004.
10. T.-C. Cheu, C.P. Johnson, and R.R. Craig Jr. Computer algorithms for calculating efficient initial vectors for subspace iteration method. Internat. J. Numer. Meth. Engrg., 24:1841 – 1848, 1987.
11. R.R. Craig Jr. and M.C.C. Bampton. Coupling of substructures for dynamic analysis. AIAA J., 6:1313–1319, 1968.
12. K. Elssel. Automated Multilevel Substructuring for Nonlinear Eigenvalue Problems. PhD thesis, Institute of Numerical Simulation, Hamburg University of Technology, 2006.
13. K. Elssel and H. Voss. Reducing huge gyroscopic eigenproblem by Automated Multi-Level Substructuring. Arch. Appl. Mech., 76:171 – 179, 2006.
14. W. Gao, X.S. Li, C. Yang, and Z. Bai. An implementation and evaluation of the AMLS method for sparse eigenvalue problems. ACM Trans. Math. Softw., 34, 2008.
15. W.C. Hurty. Vibration of structure systems by component-mode synthesis. J. Engrg. Mech. Div., ASCE, 86:51–69, 1960.
16. Z. Jia. A refined subspace iteration algorithm for large sparse eigenproblems. Appl. Numer. Math., 32:35 – 52, 2000.
17. M.F. Kaplan. Implementation of Automated Multilevel Substructuring for Frequency Response Analysis of Structures. PhD thesis, Dept. of Aerospace Engineering & Engineering Mechanics, University of Texas at Austin, 2001.
18. G. Karypis and V. Kumar. Metis. a software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices. version 4.0. Technical report, University of Minnesota, Minneapolis, 1998.

19. Y.C. Lam and A.F. Bertolini. Acceleration of the subspace iteration method by selective repeated inverse iteration. Finite Elements in Analysis and Design, 18:309 – 317, 1994.
20. X. Lu, J.H. Lin, and W.X. Zhong. Subspace iteration method in multi-level substructure systems. Comput. Struct., 33:459 – 462, 1989.
21. D.T. Nguyen and J.S. Arora. Eigensolution for large structural systems with sub-structures. Int. J. Numer. Meth. Engrg., 4:333 – 341, 1980.
22. H. Rutishauser. Computational aspects of F.L. Bauer’s simultaneous iteration method. Numer. Math., 13:4 – 13, 1969.
23. R.S. Sedeh, M. Bathe, and K.-J. Bathe. A subspace iteration method in protein normal mode analysis. J. Comput. Chem., 31:66 – 74, 2009.
24. H. Voss, J. Yin, and P. Chen. Enhancing eigenvector approximations of huge gyroscopic eigenproblems from AMLS with subspace iteration. Trans. Control Mech. Syst., 2, 2013.
25. L. Xian, L. Jiahao, and Z. Wanxie. Subspace iteration method in multi-level substructure systems. Computers & Structures, 33:459 – 462, 1989.
26. T. Yamamoto and H. Ohtsuda. Subspace iteration accelerated by using chebyshev polynomials for eigenvalue problems with symmetric matrices. Internat. J. Numer. Meth. Engrg., 10:935–944, 1976.
27. J. Yin, H. Voss, and P. Chen. Improving eigenpairs of automated multilevel substructuring with subspace iteration. Computers & Structures, 119:115 – 124, 2013.
28. Q. C. Zhao, P. Chen, W. B. Peng, Y. C. Gong, and M. W. Yuan. Accelerated subspace iteration with aggressive shift. Computers & Structures, 85:1562 – 1587, 2007.