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

Heinrich Voss
voss@tuhh.de

Joint work with Jiacong Yin and Pu Chen, Peking University

Hamburg University of Technology
Institute of Mathematics

TUHH
Technische Universität Hamburg-Harburg

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.

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.

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 talk we take advantage of the automated multi-level sub-structuring (AMLS) to construct an accurate initial subspace for SIM.

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 talk 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.

Outline

Outline

Problem definition

Determine all eigenvalues $0 < \lambda_1 \leq \lambda_2 \leq \ldots$ not exceeding a given value λ_{max} and the corresponding eigenvectors of a generalized eigenvalue problem

$$Kx = \lambda Mx \tag{1}$$

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.

Problem definition

Determine all eigenvalues $0 < \lambda_1 \leq \lambda_2 \leq \ldots$ not exceeding a given value $\lambda_{\max}$ and the corresponding eigenvectors of a generalized eigenvalue problem

$$Kx = \lambda Mx \tag{1}$$

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.

A robust and efficient methods for solving huge generalized eigenvalue problems in structural dynamics is the subspace iteration method.

Problem definition

Determine all eigenvalues $0 < \lambda_1 \leq \lambda_2 \leq \ldots$ not exceeding a given value $\lambda_{\max}$ and the corresponding eigenvectors of a generalized eigenvalue problem

$$Kx = \lambda Mx \tag{1}$$

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.

A 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 (1957) and Rutishauser (1969) it was proposed and so named in the 1970s by Bathe for computing eigenfrequencies and mode shapes of structures such as buildings and bridges using finite element discretizations.

Problem definition

Determine all eigenvalues $0 < \lambda_1 \leq \lambda_2 \leq \ldots$ not exceeding a given value $\lambda_{\max}$ and the corresponding eigenvectors of a generalized eigenvalue problem

$$Kx = \lambda Mx \tag{1}$$

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.

A 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 (1957) and Rutishauser (1969) it was proposed and so named in the 1970s by Bathe 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.

Problem definition

The basic subspace iteration method for determining p eigenvalues corresponding to the p smallest eigenvalues $\lambda_1 \leq \cdots \leq \lambda_p$ of problem (1) reads as follows:

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

1: **for** $k = 1, 2, \ldots$ **do**
2: inverse iteration step: solve $KZ_k = MX_{k-1}$ for Z_k
3: compute projected stiffness and mass matrix $K_k := Z_k^H KZ_k$ and $M_k := Z_k^H MZ_k$
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

Problem definition

The basic subspace iteration method for determining p eigenvalues corresponding to the p smallest eigenvalues $\lambda_1 \leq \cdots \leq \lambda_p$ of problem (1) reads as follows:

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

1: **for** $k = 1, 2, \ldots$ **do**
2: inverse iteration step: solve $KZ_k = MX_{k-1}$ for Z_k
3: compute projected stiffness and mass matrix $K_k := Z_k^H KZ_k$ and $M_k := Z_k^H MZ_k$
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 converges linearly (Bathe 1977) where the convergence rate for the ith smallest eigenvalue λ_i is

$$\lambda_i/\lambda_{q+1}, \quad i = 1, \ldots, q.$$

Problem definition

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 ith column of Q_k, then it holds

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

Problem definition

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 ith 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 (Bathe & Ramaswamy 1980), and aggressive shifts (Zhao et al. 2007), using refined Ritz vectors (Jia 2000), combining SIM with Chebyshev polynomials (Yamamozo & Ohtsudo 1976), selective repeated inverse iteration (Lam & Bertolini 1994), SIM in conjunction with sub-structure techniques (Nguyen & Arora 1980), and multi-level sub-structuring (Lu et al. 1989, Xian et al. 1989), to name just a few.

Problem definition

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 $KZ_{k+1} = MX_k$ efficiently.

Problem definition

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 $KZ_{k+1} = MX_k$ 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 (Chen et al. 1987).

Problem definition

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 $KZ_{k+1} = MX_k$ 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 (Chen et al. 1987).

In the original development of the subspace iteration procedure based on the experience on seeking a small number of eigenpairs Bathe 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.

Problem definition

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 $KZ_{k+1} = MX_k$ 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 (Chen et al. 1987).

In the original development of the subspace iteration procedure based on the experience on seeking a small number of eigenpairs Bathe 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 (assuming that $\lambda_j - \lambda_1$ is a known polynomial of j of degree less than or equal to 4) Bathe (2013) developed a formula for an optimal q. If the explicit form of the polynomial is not known he suggested to use the very simple formula $q := \max\{p + 8, 2p\}$.

Problem definition

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

$$KX_k = MX_{k-1}, \quad k = 1, 2, \ldots, m$$

and subsequent projection of (1) to $\text{span}(X_m)$ to extract the wanted eigeninformation.

Problem definition

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

$$KX_k = MX_{k-1}, \quad k = 1, 2, \ldots, m$$

and subsequent projection of (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 remedy this deficiency.

Problem definition

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

$$KX_k = MX_{k-1}, \quad k = 1, 2, \ldots, m$$

and subsequent projection of (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 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}, $j = 1, \ldots, p$ 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:

Problem definition

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

1: **for** $k = 1, 2, \ldots$,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

Outline

Automated Multi-Level Substructring

A method which is capable to determine a large number of eigenpairs of a huge symmetric problem with moderate accuracy is the Automated Multi-Level Substructuring (AMLS).

Automated Multi-Level Substructring

A method which is capable to determine a large number of eigenpairs of a huge symmetric problem with moderate accuracy is the Automated Multi-Level Substructuring (AMLS).

AMLS was introduced by Bennighof (1998) and was applied to huge problems of frequency response analysis.

Automated Multi-Level Substructring

A method which is capable to determine a large number of eigenpairs of a huge symmetric problem with moderate accuracy is the Automated Multi-Level Substructuring (AMLS).

AMLS was introduced by Bennighof (1998) and was applied to huge problems of frequency response analysis.

The large finite element model is recursively divided into very many substructures on several levels based on the sparsity structure of the system matrices.

Automated Multi-Level Substructring

A method which is capable to determine a large number of eigenpairs of a huge symmetric problem with moderate accuracy is the Automated Multi-Level Substructuring (AMLS).

AMLS was introduced by Bennighof (1998) and was applied to huge problems of frequency response analysis.

The large finite element model is recursively divided into very many substructures on several levels based on the sparsity structure of the system matrices.

Assuming that the interior degrees of freedom of substructures depend quasistatically on the interface degrees of freedom, and modeling the deviation from quasistatic dependence in terms of a small number of selected substructure eigenmodes the size of the finite element model is reduced substantially yet yielding satisfactory accuracy over a wide frequency range of interest.

Automated Multi-Level Substructring

A method which is capable to determine a large number of eigenpairs of a huge symmetric problem with moderate accuracy is the Automated Multi-Level Substructuring (AMLS).

AMLS was introduced by Bennighof (1998) and was applied to huge problems of frequency response analysis.

The large finite element model is recursively divided into very many substructures on several levels based on the sparsity structure of the system matrices.

Assuming that the interior degrees of freedom of substructures depend quasistatically on the interface degrees of freedom, and modeling the deviation from quasistatic dependence in terms of a small number of selected substructure eigenmodes the size of the finite element model is reduced substantially yet yielding satisfactory accuracy over a wide frequency range of interest.

Recent studies in vibro-acoustic analysis of passenger car bodies where very large FE models with more than six million degrees of freedom appear and several hundreds of eigenfrequencies and eigenmodes are needed have shown that AMLS is considerably faster than Lanczos type approaches.

Automated Multi-Level Substructring

Consider

$$Kx = \lambda Mx$$

where K and M are symmetric and positive definite (typically the stiffness and mass matrices of a FE model of a structure).

Automated Multi-Level Substructring

Consider

$$Kx = \lambda Mx$$

where K and M are symmetric and positive definite (typically the stiffness and mass matrices of a FE model of a structure).

Similarly as in the component mode synthesis method (CMS) the structure is partitioned into a small number of substructures based on the sparsity pattern of the system matrices, but more generally than in CMS these substructures in turn are sub-structured on a number of levels yielding a tree topology for the substructures.

Automated Multi-Level Substructring

Consider

$$Kx = \lambda Mx$$

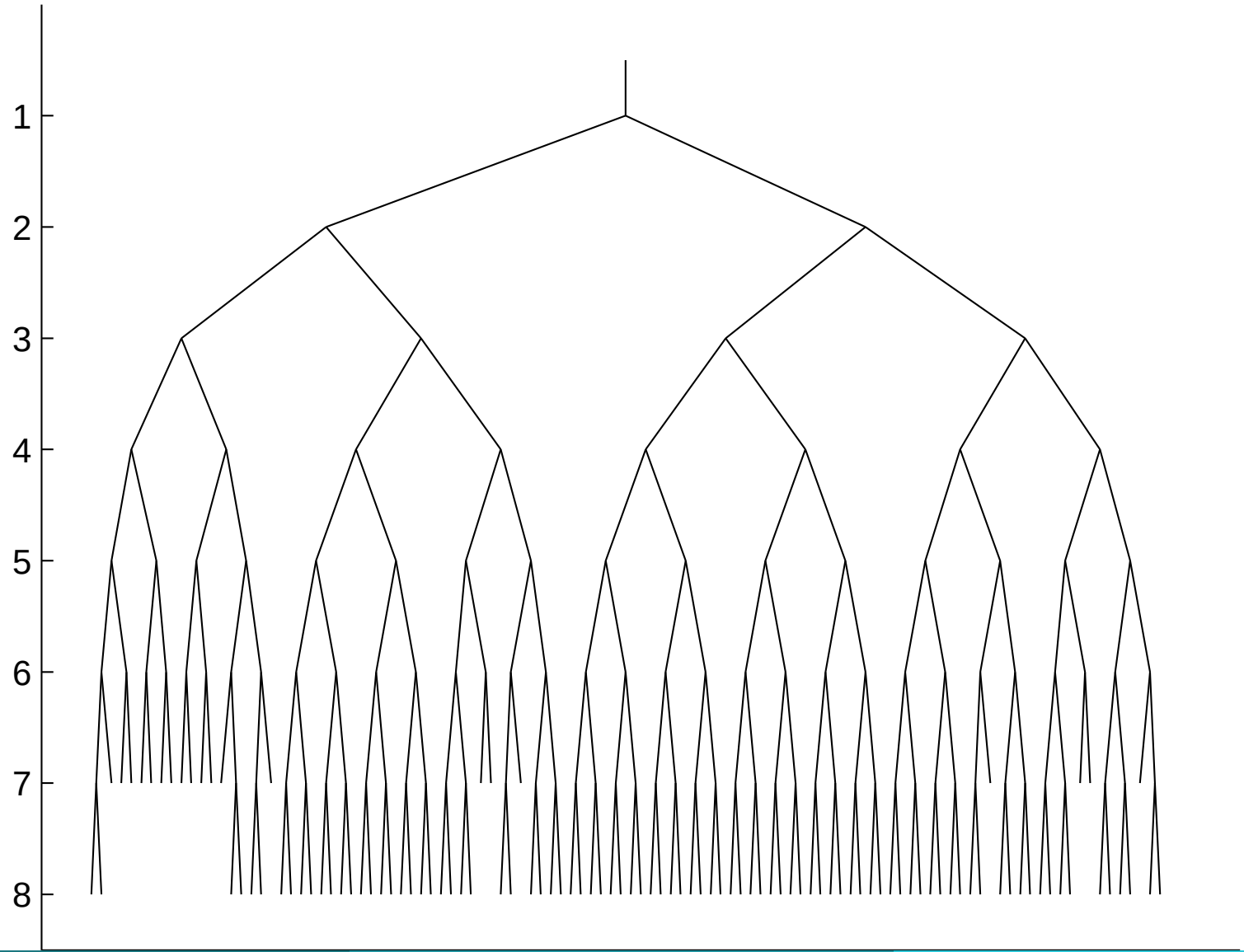
where K and M are symmetric and positive definite (typically the stiffness and mass matrices of a FE model of a structure).

Similarly as in the component mode synthesis method (CMS) the structure is partitioned into a small number of substructures based on the sparsity pattern of the system matrices, but more generally than in CMS these substructures in turn are sub-structured on a number of levels yielding a tree topology for the substructures.

AMLS consists of two ingredients.

1. First, based on the substructuring the stiffness matrix K is transformed to block diagonal form by block Gaussian elimination,
2. and secondly, the dimension is reduced substantially by modal condensation of the substructures.

Sub-structure tree



Decoupling sub-structures

After reordering the degrees of freedom the current reduced problem obtains the following form

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

where x_p denotes the degrees of freedom which where 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.

Decoupling sub-structures

After reordering the degrees of freedom the current reduced problem obtains the following form

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

where x_p denotes the degrees of freedom which where 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.

Eliminate K_{cr} by (symmetric) block-Gaussian elimination:

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

Reduction of dimension

To reduce the dimension of the eigenproblem we determine for every substructure (after decoupling it from the remaining degrees of freedom in the stiffness matrix as above, and neglecting connections to other substructures in the mass matrix) all eigenvalues λ_{cj} of $K_c z = \lambda_{cj} M_c z$ not exceeding a cut off frequency λ_{cutoff} and corresponding eigenvectors z_{sj}, $j = 1, \ldots, m_s$.

Reduction of dimension

To reduce the dimension of the eigenproblem we determine for every substructure (after decoupling it from the remaining degrees of freedom in the stiffness matrix as above, and neglecting connections to other substructures in the mass matrix) all eigenvalues λ_{cj} of $K_c z = \lambda_{cj} M_c z$ not exceeding a cut off frequency λ_{cutoff} and corresponding eigenvectors z_{sj}, $j = 1, \ldots, m_s$.

Then with $Z_s = [z_{s1}, \ldots, z_{sm_s}]$ and the global block diagonal projection matrix $Z = \text{diag}\{Z_1, \ldots, Z_m\}$ we finally get the reduced eigenvalue problem

$$K_c x_c = \lambda M_c x_c \tag{2}$$

where $K_c = Z^T \tilde{K} Z = Z^T U^T K U Z$ is a diagonal matrix and $M_c = Z^T \tilde{M} Z = Z^T U^T M U Z$ has generalized block arrowhead form.

Reduction of dimension

To reduce the dimension of the eigenproblem we determine for every substructure (after decoupling it from the remaining degrees of freedom in the stiffness matrix as above, and neglecting connections to other substructures in the mass matrix) all eigenvalues λ_{cj} of $K_c z = \lambda_{cj} M_c z$ not exceeding a cut off frequency λ_{cutoff} and corresponding eigenvectors z_{sj}, $j = 1, \ldots, m_s$.

Then with $Z_s = [z_{s1}, \ldots, z_{sm_s}]$ and the global block diagonal projection matrix $Z = \text{diag}\{Z_1, \ldots, Z_m\}$ we finally get the reduced eigenvalue problem

$$K_c x_c = \lambda M_c x_c \tag{2}$$

where $K_c = Z^T \tilde{K} Z = Z^T U^T K U Z$ is a diagonal matrix and $M_c = Z^T \tilde{M} Z = Z^T U^T M U Z$ has generalized block arrowhead form.

Important: In an implementation the block Gaussian eliminations and the condensations are performed in an interleaving way to avoid the storage of large dense sub-matrices of the transformed mass matrix which would occur in the course of the block elimination: as soon as a sub-matrix pencil (K_c, M_c) has been formed, the eigenproblem $K_c Z_c = M_c Z_c \Lambda_c$ is solved and the corresponding projection is executed.

Outline

Preconditioning SIM with AMLS

If $U := U_1 U_2 \ldots U_m$ denotes the product of all transformations in the elimination steps, the eigenvalue problem (1) is equivalent to

$$\hat{K}\hat{x} = \lambda \hat{M}\hat{x}, \quad \hat{K} := U^H K U, \ \hat{M} := U^H M U, \ U := U_1 U_2 \ldots U_m \tag{3}$$

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.

Preconditioning SIM with AMLS

If $U := U_1 U_2 \ldots U_m$ denotes the product of all transformations in the elimination steps, the eigenvalue problem (1) is equivalent to

$$\hat{K}\hat{x} = \lambda \hat{M}\hat{x}, \quad \hat{K} := U^H K U, \ \hat{M} := U^H M U, \ U := U_1 U_2 \ldots U_m \tag{3}$$

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 (3) is also not efficient.

Preconditioning SIM with AMLS

If $U := U_1 U_2 \ldots U_m$ denotes the product of all transformations in the elimination steps, the eigenvalue problem (1) is equivalent to

$$\hat{K}\hat{x} = \lambda \hat{M}\hat{x}, \quad \hat{K} := U^H K U, \ \hat{M} := U^H M U, \ U := U_1 U_2 \ldots U_m \tag{3}$$

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 (3) 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 (3), but to evaluate $\hat{M}V$ taking advantage of the transformation matrix U and the sparse structure of the original mass matrix M.

Preconditioning SIM with AMLS

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

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

Outline

Example 1

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 $117'990$ degrees of freedom, where the number of non-zeros in the stiffness matrix is $n_z(K) = 11'243'248$ and in the mass matrix is $n_z(M) = 5'590'256$. We are interested in all eigenvalues which are less than $2 \cdot 10^5$ $(\text{rad/s})^2$.

Example 1

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 $117'990$ degrees of freedom, where the number of non-zeros in the stiffness matrix is $n_z(K) = 11'243'248$ and in the mass matrix is $n_z(M) = 5'590'256$. We are interested in all eigenvalues which are less than $2 \cdot 10^5$ $(\text{rad/s})^2$.

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

Example 1

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 $117'990$ degrees of freedom, where the number of non-zeros in the stiffness matrix is $n_z(K) = 11'243'248$ and in the mass matrix is $n_z(M) = 5'590'256$. We are interested in all eigenvalues which are less than $2 \cdot 10^5$ $(\text{rad/s})^2$.

An AMLS run with cut-off frequency $\omega_{\text{cutoff}} = 1 \cdot 10^6$ $(\text{rad/s})^2$ suggests to choose $p = 175$ and to include $q = 350$ trial eigenmodes into our computations. The 350th eigenvalue of (1) is $\lambda_{350} \approx 6 \cdot 10^5$ $(\text{rad/s})^2$, 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 to fix the sub-structuring and the Intel Math Kernel Library v10.3 optimized LAPACK to achieve high performance of computations.

Example 1

Table: Computation time of SIM and SIM-AMLS

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

Example 1

Table: Computation time of SIM and SIM-AMLS

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

The AMLS reduction and the computation of the eigenvector approximations by AMLS provide the initial approximation of eigenvectors, and are needed for both methods.

Example 1

Table: Computation time of SIM and SIM-AMLS

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

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.

Example 1

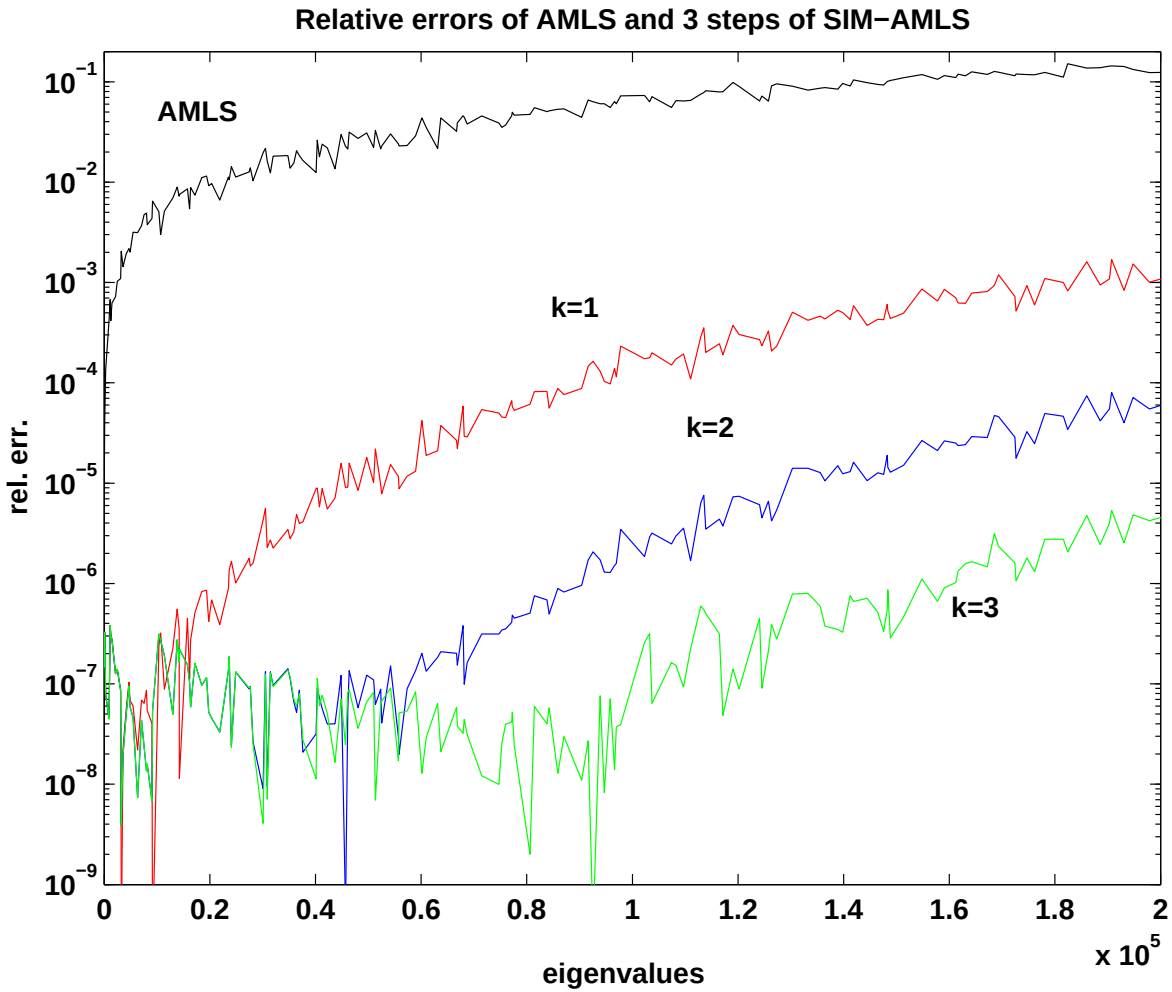


Figure: Relative errors of eigenvalue approximations

Example 1

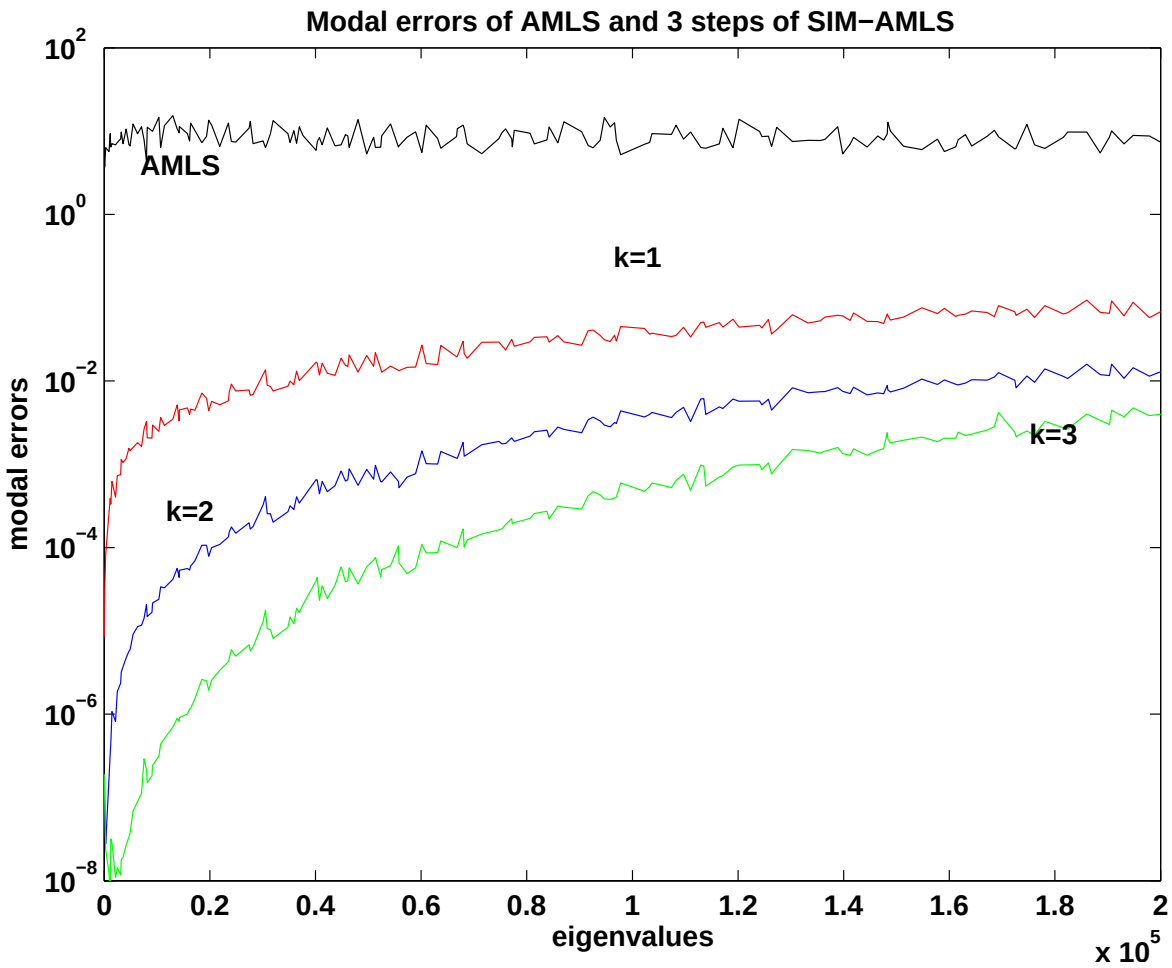


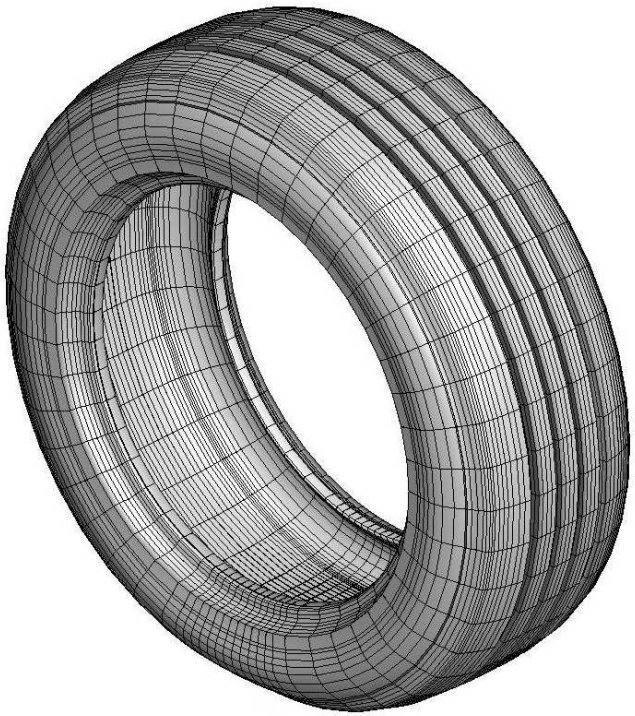
Figure: Modal errors of eigenpair approximations

Example 2

Consider the FE model of a rotating tire

$$Q(\omega)x := Kx + i\omega Gx - \omega^2 Mx = 0$$

which consists of 39204 brick elements with 124992 degrees of freedom and accounts for 20 different material groups. The speed is assumed to be 60 km/h. Our aim is to determine approximations to the smallest 100 eigenvalues (i.e. less than 1300 Hz) and corresponding eigenvectors.



Example 2

The numerical tests were performed on a 64-bit HP workstation with an Intel Xeon CPU (3.20 GHz, 2 cores) and 24GB memory (less than 1 GByte was used!). AMLS and the two subspace iteration algorithms were implemented with Matlab R2009a.

Example 2

The numerical tests were performed on a 64-bit HP workstation with an Intel Xeon CPU (3.20 GHz, 2 cores) and 24GB memory (less than 1 GByte was used!). AMLS and the two subspace iteration algorithms were implemented with Matlab R2009a.

The AMLS method addressing the linear eigenvalue problem $Kx = \lambda Mx$ costs 881.2 seconds for the AMLS reduction and 270.4 seconds for solving the projected linear eigenvalue problem of dimension 2263 by `eig`.

Example 2

Table: Computation time of NormalSIM and AMLS-SIM with 200 iteration vectors

Computational Steps	NormalSIM(s)	AMLS-SIM(s)
Compute initial eigenvectors from AMLS	9.2	2.4
1 Iteration	2401.0	67.8
2 Iterations	Not computed	141.3
3 Iterations	Not computed	216.8
4 Iterations	Not computed	289.4
Compute $\hat{K}$ and $\hat{M}$ if $n_k = 1$	10.0	6.2
Compute $\hat{K}$ and $\hat{M}$ if $n_k > 1$	Not computed	8.7
Solve $\hat{K}Z = \hat{M}V\hat{\Lambda}$ by `eig`	2.6	2.6
Compute final eigenvectors	0.9	25.6

Numerical example

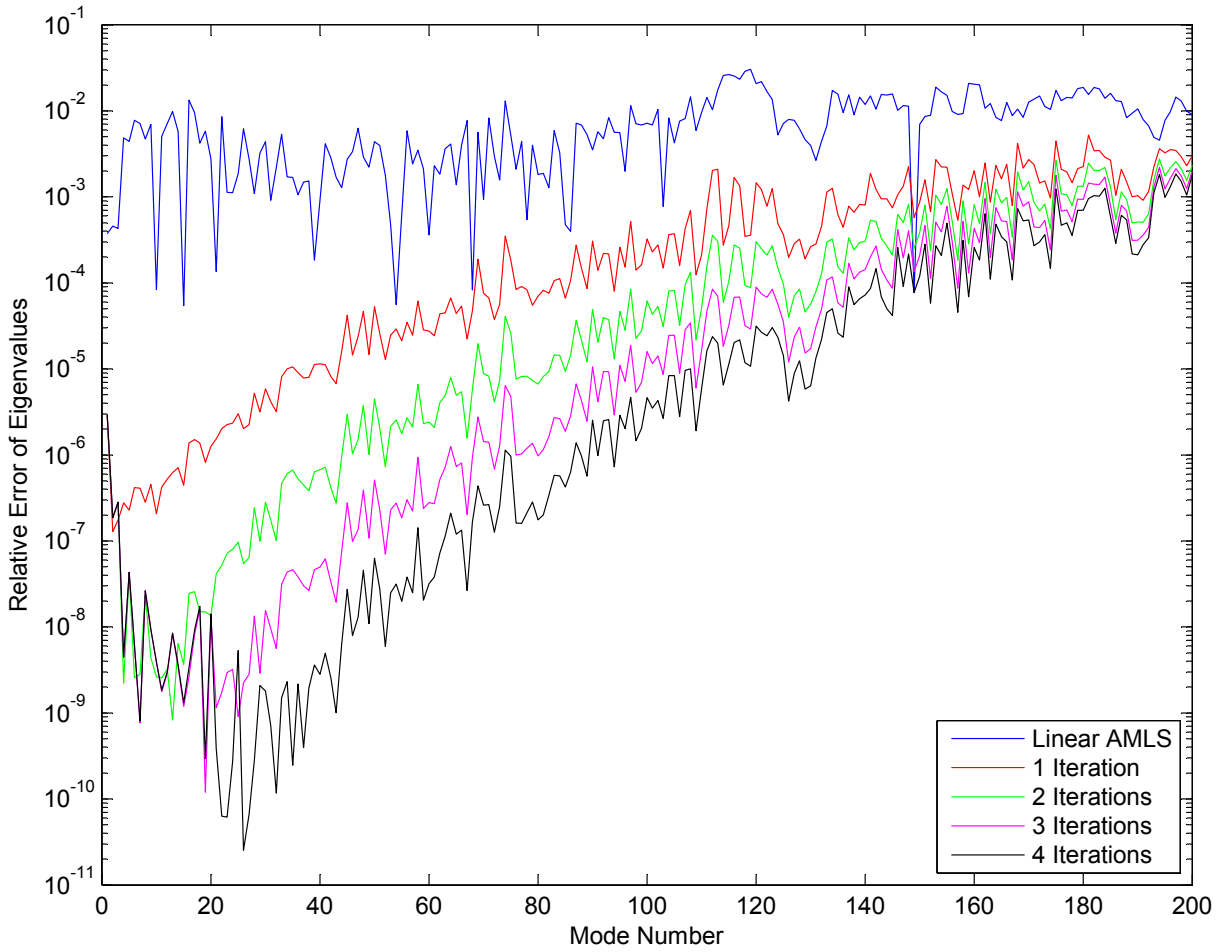


Figure: Relative errors of eigenvalues computed with subspace iteration with AMLS utilizing 200 iteration vectors

Numerical example

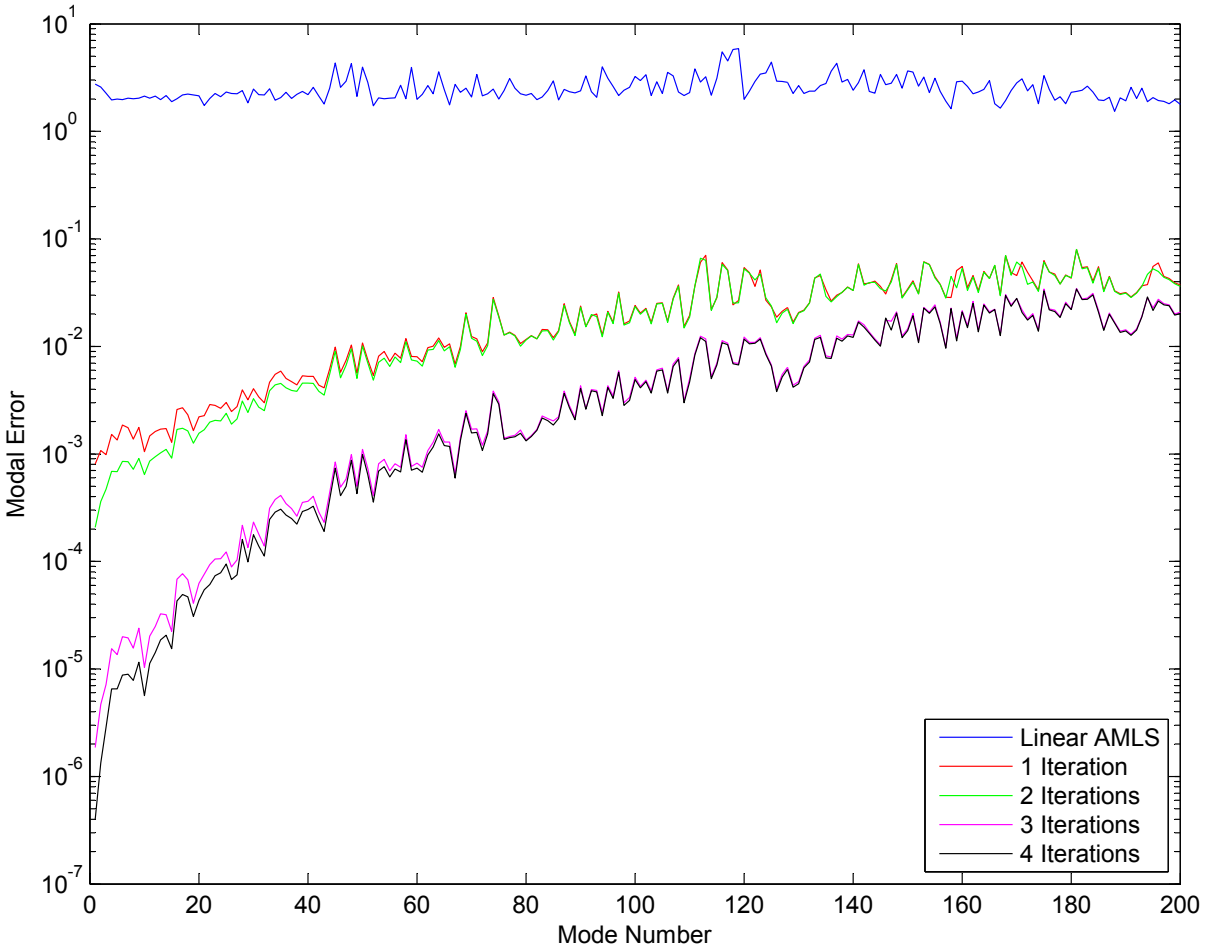


Figure: Modal errors computed with subspace iteration with AMLS utilizing 200 iteration vectors

Outline

Conclusions

- The object of this talk 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.

Conclusions

- The object of this talk 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.

Conclusions

- The object of this talk 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 stiffness 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.

Conclusions

- The object of this talk 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 stiffness 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, and for a gyroscopic quadratic eigenvalue problem governing the eigenmodes of a rotating tire.

Thanks for your attention!

References

- J. Yin, H. Voss, P. Chen.
  Improving eigenpairs of automated multilevel substructuring with subspace iteration.
  Computers & Structures 119, 115 – 124 (2013)
- H. Voss, J. Yin, P. Chen.
  Enhancing eigenvector approximations of huge gyroscopic eigenproblems from AMLS with subspace iteration.
  Trans. Control Mech. Syst. 2, 1 – 8 (2012)
- H. Voss, J. Yin, P. Chen.
  Preconditioning subspace iteration for large eigenvalue problems with Automated Multi-Level Sub-structuring.
  Report 172, Institute of Mathematics, Hamburg University of Technology, 2013

Thanks for your attention!