A FAST SUBSPACE ITERATION PROCEDURE FOR IMPROVING AUTOMATED MULTI-LEVEL SUBSTRUCTURING

Jiacong Yin*, Heinrich Voss** & Pu Chen*[a)]
*State Key Laboratory for Turbulence and Complex Systems & Department of Mechanics and Aerospace Engineering, College of Engineering, Peking University, Beijing, 100871, China
**Institute of Numerical Simulation, Hamburg University of Technology, Hamburg, D-21071, Germany

Summary The Automated Multi-Level Substructuring (AMLS) proposed by Bennighof and Kaplan et.al. is a substitutive method that was recently developed for extracting thousands of eigenvalues of very large problems with millions of degrees of freedom, which cannot be solved by standard Lanczos algorithm or subspace iteration. Although AMLS has been widely used in the Noise-Vibration-Harshness (NVH) analysis of automobile industry, as a novel eigen solution algorithm, the accuracy of corresponding eigenvectors computed by AMLS is rarely to be checked. In order to extend AMLS into wider area of dynamical analysis, this paper investigates the procedure of eigenvectors extracting from AMLS and then improve precision of the eigenpairs through an implicit fast subspace iteration. Numerical results show that the precision of AMLS eigenpairs could be greatly improved with quite a few iteration steps.

AUTOMATED MULTI-LEVEL SUBSTRUCTURING

The Automated Multi-Level Substructuring (AMLS) method, which was recently proposed by Bennighof and Kaplan et.al. [1, 2, 3], is a substitutive method for extracting thousands of eigenvalues of very large problems

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

with millions of degrees of freedom (DOFs). AMLS is essential a condensation method, which could be seen as an extension of the Component Mode Synthesis (CMS) method [4, 5]. The standard AMLS algorithm described in [2] contains following five phases:

Algorithm 1 Five phases of the standard AMLS algorithm

1: Generate the Finite Element (FE) model and output the FE stiffness and mass matrices ($K \in \mathbb{R}^{n\times n}$ and $M \in \mathbb{R}^{n\times n}$).
2: Utilize graph partition tools, e.g. METIS [6], to automatically divide the original structure into N substructures with multi-levels. In the viewpoint of matrices, K and M are reordered according to their sparse patterns.
3: Execute the block Gaussian elimination transformation $U \in \mathbb{R}^{n\times n}$ to transform the reordered K and M into $\tilde{K} = U^T K U$ and $\tilde{M} = U^T M U$; then project $\tilde{K}$ and $\tilde{M}$ onto the so-called *AMLS subspace* to obtain $K_c = Z^T \tilde{K} Z \in \mathbb{R}^{m\times m}$ and $M_c = Z^T \tilde{M} Z \in \mathbb{R}^{m\times m}$ ($m \ll n$). Here $Z = \mathrm{diag}[Z_1, \ldots, Z_N] \in \mathbb{R}^{n\times m}$ contains N calculated eigenvector matrices Z_i ($i = 1, \ldots, N$) with a modal truncation for each substructure i.
4: Solve the reduced eigensystem $K_c X_c = M_c X_c \Lambda_c$, where $\Lambda_c \in \mathbb{R}^{n_p\times n_p}$ contains the n_p extracted AMLS approximate eigenvalue of Eq.(1) ($n_p \ll m$); $X_c \in \mathbb{R}^{m\times n_p}$ is the corresponding eigenvector matrix in the AMLS subspace.
5: Compute frequency responses in the AMLS subspace if requested.

Phase 3 is the kernel of AMLS. In implementation, the Gaussian transformation U and the reduction Z are interweaved to perform on each substructure i by the component matrices U_i and $\hat{Z}_i = \mathrm{diag}[I_1, \ldots, I_{i-1}, Z_i, I_{i+1}, \ldots, I_N]$, respectively. This interweaved operation follows the sequence of *postorder traversal* of the *AMLS tree*. If the N substructures are numbered in accordance with the postorder traversal, then $U = U_1 U_2 \ldots U_N$ and $Z = \hat{Z}_1 \hat{Z}_2 \ldots \hat{Z}_N$. $T = UZ$ is named as the global *transform matrix* of AMLS. Nevertheless, T is not explicitly formed in part nor whole unless the calculated frequency responses in AMLS subspace are required to transform back to the original FE space.

A FAST SUBSPACE ITERATION PROCEDURE FOR AMLS

Computation of AMLS Eigenvectors

The standard AMLS only extracts approximate eigenvalues of Eq.(1). In fact, the corresponding AMLS approximate eigenvectors are $V = UZX_c$. This equation is equal to $V = TX_c$. But we emphasize that $V = TX_c$ is inefficient for computation because of the time consuming assemblage of the entire T and Gigabytes for saving it. Actually, a faster way of obtaining V is to compute $\tilde{V} = ZX_c$ firstly, and then transform $\tilde{V}$ back to the original space following the *preorder traversal* of the AMLS tree, i.e. computation starting from the root of the tree by $\tilde{V}^{(N-1)} = U_N \tilde{V}$, and then addressing its offsprings by $\tilde{V}^{(N-2)} = U_{N-1} \tilde{V}^{(N-1)}$, etc. Our tests reveal that the computation of V with the preorder traversal is 25 times faster than the postorder traversal for a problem with 117,990 DOFs.

[a)]Corresponding author. E-mail: chenpu@pku.edu.cn. 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.

Table 1. Time cost of the standard AMLS (t_{AMLS}), the computation of eigenvectors (t_{eigvec}), a single iteration of the fast subspace iteration procedure ($t_{\mathrm{SI-FAST}}$) and the normal subspace iteration ($t_{\mathrm{SI-Normal}}$), as well as the modal errors (ϵ_{AMLS} and ϵ_{SI})

Examples	n	n_p	t_{AMLS}	t_{eigvec}	$t_{\mathrm{SI\text{-}FAST}}$	$t^{*}_{\mathrm{Factorization}}$	$t_{\mathrm{SI\text{-}Normal}}$	ϵ_{AMLS}	ϵ_{SI} after the first iteration
bcsstk24	3,562	201	0.78s	0.05s	0.53s	0.56s	0.75s	$O(10^{2})$	$O(10^{-2}) \sim O(10^{0})$
wtaoc01k	23,328	144	11.66s	0.46s	4.35s	1.28s	7.70s	$O(10^{0})$	$O(10^{-4}) \sim O(10^{-1})$
wtnh01k	117,990	350	100.21s	7.60s	69.55s	7.99s	167.03s	$O(10^{1})$	$O(10^{-4}) \sim O(10^{-1})$
bs6_k	517,161	300	308.39s	33.08s	210.07s	27.17s	385.71s	$O(10^{0})$	$O(10^{-6}) \sim O(10^{-2})$

* $t_{\mathrm{Factorization}}$ represents the time cost of the factorization ($K = LDL^T$ or $K = LL^T$) required for the normal subspace iterations.

Fast Subspace Iteration Procedure

However, we found that AMLS eigenvectors achieve worse precision than the Lanczos method in terms of the *modal error*

$$\epsilon = \frac{\|Kx - \lambda Mx\|_2}{\|\lambda Mx\|_2}. \tag{2}$$

Our computations demonstrate $\epsilon_{\mathrm{AMLS}} \sim O(10^{0} - 10^{1})$, while $\epsilon_{\mathrm{Lanczos}}$ could be as low as $O(10^{-10})$ in some commercial codes. In order to improve the AMLS eigenvectors, we developed a fast subspace iteration procedure for AMLS by taking advantage of the block diagonal property of the transformed stiffness $\tilde{K}$ (see Algorithm 2).

Algorithm 2 A fast subspace iteration procedure for improving AMLS.

Require: the transformed eigenvectors $\tilde{V}$, the transformed stiffness matrix $\tilde{K}$ and the transformation matrix U by AMLS.

1: initialize the iteration vector $\tilde{Q}^{(0)} = \tilde{V} = ZX_c$ and determine the maximum iteration number q.
2: **for** $k = 1, 2, \ldots, q$ **do**
3: transform backward $Q^{(k-1)} = U\tilde{Q}^{(k-1)}$ and then compute $R = MQ^{(k-1)}$.
4: transform forward $\tilde{R} = U^T R$ and then solve $\tilde{Q}^{(k)}$: $\tilde{K}\tilde{Q}^{(k)} = \tilde{R}$.
5: **end for**
6: reduce stiffness matrix by $K_c^{(q)} = \tilde{R}^T\tilde{Q}^{(q)}$ and then transform backward $Q^{(q)} = U\tilde{Q}^{(q)}$.
7: reduce mass matrix by $M_c^{(q)} = (Q^{(q)})^T MQ^{(q)}$.
8: solve $K_c^{(q)}X_c^{(q)} = M_c^{(q)}X_c^{(q)}\Lambda_c^{(q)}$ to obtain the improved eigenvalues $\Lambda_c^{(q)}$.
9: compute the improved eigenvectors by $V^{(q)} = Q^{(q)}X_c^{(q)}$.

To reveal the improvement and efficiency of the proposed algorithm, computational results of four eigenproblems with different scales are listed in Table 1. All numerical experiments were tested on a Linux platform with an Intel Pentium D CPU(3.46GHz, 4 Cores) and 7.7GB memory. METIS and the Intel Math Kernel Library optimized BLAS, LAPACK, PARDISO were employed for programming a AMLS code with high performance .

CONCLUSIONS

A fast subspace iteration procedure for AMLS is presented by taking advantage of the block diagonal property of the transformed AMLS stiffness $\tilde{K}$. Numerical results show that the precision of AMLS eigenpairs could be greatly improved even after one subspace iteration step. In addition, the proposed fast subspace iteration procedure with $Q^{(k+1)} = U\tilde{K}^{-1}U^T MQ^{(k)}$ saves more computational costs than the normal subspace iteration with $Q^{(k+1)} = K^{-1}MQ^{(k)}$ for larger problems. This research may extend AMLS into wider FE dynamical analysis with a higher accuracy.

References

[1] Bennighof J. K., Kim C. K.: An adaptive multi-level substructuring method for efficient modeling of complex structures. *In Proceedings of the AIAA 33rd SDM Conference*:1631–1639. Dallas, Texas, USA, 1992.

[2] Kaplan M. F.: Implementation of automated multilevel substructuring for frequency response analysis of structures. *PhD Thesis*, The University of Texas at Austin, Texas, USA, 2001.

[3] Bennighof J. K., Lehoucq R. B.: An automated multilevel substructuring method for eigenspace computation in linear elastodynamics. *SIAM J. Sci. Comput* **25**:2084–2106, 2004.

[4] Hurty W. C.: Vibrations of structural systems by component-mode synthesis. *Journal of the Engineering Mechanics Division*. **86**:51–69, 1960.

[5] Craig R. R., Bampton M. C. C.: Coupling of substructures for dynamic analysis. *AIAA Journal* **6**:1313–1319, 1968.

[6] Karypis G., Kumar V.: Metis - a software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices. version 4.0. *Technique Report*, University of Minnesota, Minneapolis, USA, 1998.