Performance Evaluation of a Multilevel Sub-structuring Method for Sparse Eigenvalue Problems*

Weiguo Gao[1], Xiaoye S. Li[1], Chao Yang[1], and Zhaojun Bai[2]

[1] Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720. {wggao, xsli, cyang}@lbl.gov
[2] Department of Computer Science, University of California, Davis, CA 95616. bai@cs.ucdavis.edu

1 Introduction

The automated multilevel sub-structuring (AMLS) method [2, 7, 3] is an extension of a simple sub-structuring method called *component mode synthesis* (CMS) [6, 4] originally developed in the 1960s. The recent work by Bennighof and Lehoucq [3] provides a high level mathematical description of the AMLS method in a continuous variational setting, as well as a framework for describing AMLS in matrix algebra notations. The AMLS approach has been successfully used in the vibration and acoustic analysis of very large scale finite element models of automobile bodies [7]. In this paper, we evaluate the performance of AMLS on other types of applications.

Similar to the domain decomposition techniques used in solving linear systems, AMLS reduces a large-scale eigenvalue problem to a sequence of smaller problems that are easier to solve. The method is amenable to an efficient parallel implementation. However, a few questions regarding the accuracy and computational efficiency of the method remain to be carefully examined. Our earlier paper [12] addressed some of these questions for a single-level algorithm. We developed a simple criterion for choosing spectral components from each sub-structure, performed algebraic analysis based on this mode selection criterion, and derived the error bounds for the approximate eigenpair associated with the smallest eigenvalue. This paper focuses on the performance of the multilevel algorithm.

* This work was supported by the Director, Office of Advanced Scientific Computing Research, Division of Mathematical, Information, and Computational Sciences of the U.S. Department of Energy under contract number DE-AC03-76SF00098.

2 The algorithmic view of AMLS

We are concerned with solving the following algebraic eigenvalue problem

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

where K is symmetric, M is symmetric positive definite, and both are sparse. Using a graph partitioning software package such as METIS [8] we can permute the matrix pencil (K, M) into a multilevel nested block structure shown below:

$$K = \begin{pmatrix} K_{11} & & & & & & \\ & K_{22} & & & sym. & & \\ K_{31} & K_{32} & K_{33} & & & & \\ & & & K_{44} & & & \\ & & & & K_{55} & & \\ & & & K_{64} & K_{65} & K_{66} & \\ K_{71} & K_{72} & K_{73} & K_{74} & K_{75} & K_{76} & K_{77} \end{pmatrix}, M = \begin{pmatrix} M_{11} & & & & & & \\ & M_{22} & & & sym. & & \\ M_{31} & M_{32} & M_{33} & & & & \\ & & & M_{44} & & & \\ & & & & M_{55} & & \\ & & & M_{64} & M_{65} & M_{66} & \\ M_{71} & M_{72} & M_{73} & M_{74} & M_{75} & M_{76} & M_{77} \end{pmatrix} \tag{2}$$

Blocks K_{ij} and M_{ij} are of size n_i-by-n_j. A byproduct of this partitioning and reordering algorithm is a *separator tree* depicted in Figure 1. The separator tree can be used to succinctly describe the matrix structure (2), the computational tasks and their dependencies in the AMLS algorithm. The internal tree nodes (marked by $\square$) represent the separators (also known as the *interface* blocks, e.g. K_{33}, K_{66} and K_{77}), and the bottom leaf nodes (marked by $\bigcirc$) represent the sub-structures (e.g. K_{11}, K_{22}, K_{44} and K_{55}). The permutation

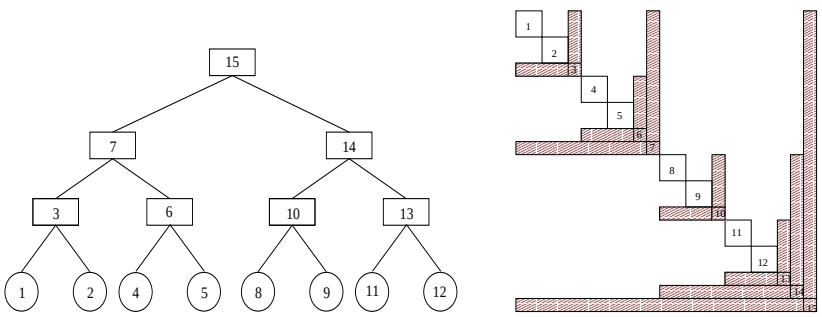

Fig. 1. Separator tree (left) and the reordered matrix (right) for a three-level dissection.

of the pencil (K, M) is followed by a block factorization of the K matrix, i.e., $K = LDL^T$, where

$$D = L^{-1}KL^{-T} = \mathrm{diag}(K_{11}, K_{22}, \widehat{K}_{33}, K_{44}, K_{55}, \widehat{K}_{66}, \widehat{K}_{77}) \stackrel{def}{=} \widehat{K} \ . \tag{3}$$

and L is given by:

$$L = \begin{pmatrix} I_{n_1} & & & & & & \\ & I_{n_2} & & & sym. & & \\ K_{31}K_{11}^{-1} & K_{32}K_{22}^{-1} & I_{n_3} & & & & \\ & & & I_{n_4} & & & \\ & & & & I_{n_5} & & \\ & & & K_{64}K_{44}^{-1} & K_{65}K_{55}^{-1} & I_{n_6} & \\ K_{71}K_{11}^{-1} & K_{72}K_{22}^{-1} & K_{73}\widehat{K}_{33}^{-1} & K_{74}K_{44}^{-1} & K_{75}K_{55}^{-1} & K_{76}\widehat{K}_{66}^{-1} & I_{n_7} \end{pmatrix} \tag{4}$$

Applying the same congruence transformation defined by L^{-1} to M yields:

$$L^{-1}ML^{-T} \stackrel{def}{=} \widehat{M} = \begin{pmatrix} M_{11} & & & & & & \\ & M_{22} & & & & sym. & \\ \widehat{M}_{31} & \widehat{M}_{32} & \widehat{M}_{33} & & & & \\ & & & M_{44} & & & \\ & & & & M_{55} & & \\ & & & \widehat{M}_{64} & \widehat{M}_{65} & \widehat{M}_{66} & \\ \widehat{M}_{71} & \widehat{M}_{72} & \widehat{M}_{73} & \widehat{M}_{74} & \widehat{M}_{75} & \widehat{M}_{76} & \widehat{M}_{77} \end{pmatrix}. \tag{5}$$

Note that $\widehat{M}$ has the same block structure as M, and only the diagonal blocks associated with the leaves of the separator tree are not altered; all the other blocks are modified. Moreover, the altered blocks of $\widehat{M}$ typically contain more non-zero elements than those in M. The eigenvalues of $(\widehat{K}, \widehat{M})$ are identical to those of (K, M), and the corresponding eigenvectors $\widehat{x}$ are related to those of the original problem (1) through $\widehat{x} = L^T x$.

Instead of computing eigenvalues of (K, M) directly, AMLS solves a number of subproblems defined by the diagonal blocks of $\widehat{K}$ and $\widehat{M}$. Suppose S_i contains eigenvectors associated with k_i desired eigenvectors of (K_{ii}, M_{ii}) (or $(\widehat{K}_{ii}, \widehat{M}_{ii})$), then, AMLS constructs a subspace in the form of

$$S = \mathrm{diag}(S_1, S_2, \ldots, S_N) \; . \tag{6}$$

The eigenvectors associated with (K_{ii}, M_{ii}) will be referred to as the *sub-structure modes*, and those associated with $(\widehat{K}_{ii}, \widehat{M}_{ii})$ will be referred to as the *coupling modes*. The approximation to the desired eigenpairs of the pencil $(\widehat{K}, \widehat{M})$ are obtained by projecting the pencil $(\widehat{K}, \widehat{M})$ onto the subspace spanned by S, i.e., we seek θ and $q \in \mathbb{R}^{\bar{k}}$, where $\bar{k} = \sum_{i=1}^{N} k_i$, such that

$$(S^T \widehat{K} S)q = \theta (S^T \widehat{M} S)q. \tag{7}$$

It follows from the Rayleigh-Ritz theory [11, page 213] that θ serves as an approximation to an eigenvalue of (K, M), and the vector formed by $z = L^{-T}Sq$ is the approximation to the corresponding eigenvector. Algorithm 1 summarizes the major steps of the AMLS algorithm.

Note that when the interface blocks are much smaller than the sub-structures, we can include all the coupling modes by replacing S_i with I_{n_i} in (6). As a result, the projected problem (7) is simplified while its dimension is still kept small.

A straightforward implementation of Algorithm 1 is not very cost-effective. The amount of memory required to store the block eliminator L and the matrix $\widehat{M} = L^{-1}ML^{-T}$ is typically high due to fill-ins. We used the following strategies to reduce this cost: (1) Since computing the desired eigenvalues and the eigenvectors does not require $\widehat{M}$ explicitly, we project M into the subspace spanned by the columns of $L^{-T}S$ incrementally as L and S are being computed in an order defined by a bottom-up traversal of the separator tree. In another word, we *interleave* Steps (2) to (5) of Algorithm 1; (2) We

use a *semi-implicit* scheme to store L. We only explicitly compute and store the blocks in the columns associated with the separator nodes. The blocks in the columns associated with the leaf nodes are not computed explicitly. Whenever needed, $K_{ji}K_{ii}^{-1}$ is applied to a matrix block directly through a sequence of sparse triangular solves and matrix-matrix multiplications.

More implementation details can be found in our longer report [5].

Algorithm 1 *Algebraic Multilevel Sub-structuring (AMLS)*

Input:	A matrix pencil (K, M), where K is symmetric and nonsingular and M is symmetric positive definite
Output:	$\theta_j \in R^1$ and $z_j \in R^n$, $(j = 1, 2, ..., k)$ such that $Kz_j \approx \theta_j M z_j$
(1)	Partition and reorder K and M to be in the form of (2)
(2)	Perform block factorization $K = LDL^T$
(3)	Apply the congruence transformation defined by L^{-1} to (K, M) to obtain $(\widehat{K}, \widehat{M})$ defined by (3) and (5)
(4)	Compute a subset of the eigenpairs of interest for the subproblems (K_{ii}, M_{ii}) (or $(\widehat{K}_{ii}, \widehat{M}_{ii})$). Then, form the matrix S in (6)
(5)	Project the matrix pencil $(\widehat{K}, \widehat{M})$ into the subspace $span\{S\}$
(6)	Compute k desired eigenpairs (θ_j, q_j) from $(S^T\widehat{K}S)q = \theta(S^T\widehat{M}S)q$, and set $z_j = L^{-T}Sq_j$ for $j = 1, 2, ..., k$

3 Performance evaluation

We evaluate the performance of AMLS on two applications. Our first problem arises from a finite element model of a six-cell damped detuned accelerator structure [9]. The eigenvalues of this generalized eigenvalue problem correspond to the cavity resonance frequencies and the eigenvectors represent the electromagnetic accelerating field. We will refer to this problem as DDS6. Our second problem arises from the normal mode vibrational analysis of a 3000-atom polyethylene (PE) particle [13]. In this application, we are interested in the low frequency vibrations of the PE molecule. We will refer to this problem as PE3K.

Our platform is a single Power3 processor with a clock speed of 375Mhz and 2 MB of level-2 cache. We use *nev* to denote the number of wanted eigenvalues. The accuracy tolerance for each subproblem is denoted by τ_{sub}, and the accuracy tolerance for the projected problem is denoted by τ_{proj}. We use *nmodes* to denote the number of modes chosen from each sub-structure.

DDS6

The dimension of this problem is 65740, and the number of non-zero entries in $K + M$ is 1455772. Table 1 shows the AMLS timing and memory usage measurements. We experimented with different partitioning levels. For a single level partitioning, we set *nmodes* to 100. When we increase the number of

levels by one, we reduce *nmodes* by half to keep the total number of sub-structure modes roughly a constant. Since the separators in this problem are small, all the coupling modes are included in the subspace (6). Column 3 shows that the total memory usage does not increase too much with increasing number of levels. By using the semi-implicit representation for L, we save some memory but need extra time for recomputing some off-diagonal blocks. This tradeoff between memory reduction and extra runtime is shown in Columns 4 and 5, which indicate that we save up to 50% of the memory with only 10-15% extra runtime. This is very attractive when memory is at a premium. Column 6 shows the time spent in the first phase of AMLS, which consists of various transformations (Steps (2)-(5) of Algorithm 1). The time spent in the second phase of the algorithm, Step (6), is reported in Column 7. The total time is reported in the last column. As the number of levels increases, the transformation time decreases, whereas the projected problem becomes larger and hence requires more time to solve. The variation of the total CPU time is small with respect to the number of levels.

Table 1. Problem DDS6, $nev = 100$, $\tau_{sub} = 10^{-10}$, $\tau_{proj} = 10^{-5}$.

levels	nmodes	mem (MB)	mem-saved (MB)	recompute (sec)	phase 1 (sec)	phase 2 (sec)	total (sec)
2	100	319	199 (38.4%)	9.2 (1.5%)	457.7	137.2	594.8
3	50	263	263 (50.0%)	51.5 (11.0%)	287.7	178.8	466.5
4	25	325	248 (43.3%)	60.7 (13.3%)	220.2	235.4	455.6
5	12	392	228 (36.8%)	64.0 (13.2%)	194.0	291.9	485.9
6	6	480	192 (28.6%)	55.3 (10.9%)	151.9	352.4	504.2

As a comparison, it took about 407 seconds and 308 Megabytes memory to compute the smallest 100 eigenpairs by a shift-and-invert Lanczos (SIL) method (using ARPACK and SuperLLT packages [10] with METIS reordering.) Thus when $nev = 100$, AMLS and SIL are comparable in both speed and memory usage. However, Figure 2 shows that AMLS is more efficient than SIL when more eigenvalues are needed. In AMLS, the time consumed by phase 1 (transformations) is roughly the same for different *nev*s. The increase in the total CPU time for a larger *nev* is mainly due to the increased cost associated with solving a larger projected problem (labeled as "AMLS-Ritz" in Figure 2), but this increase is far below linear. Linear increase in total CPU time is expected in SIL because multiple shifts may be required to compute eigenvalues that are far part. In our experiment, we set the number of eigenvalues to be computed by a single-shift SIL run to 100. Since the cost associated with each single-shift SIL run is roughly the same for each shift, the total cost for a multi-shift SIL run increases linearly with respect to *nev*.

Figure 3 shows the relative error of the smallest 100 eigenvalues returned from the AMLS calculation. As shown in the left figure, the accuracy deteriorates with increasing number of levels, which is true even for the first few eigenvalues. This is due to the limited number of modes selected in the sub-

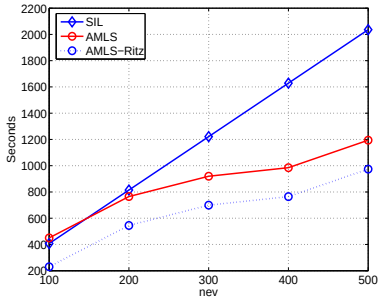

Fig. 2. Runtime of AMLS and SIL with increasing nev. Problem DDS6, $levels = 4$, $nmodes = 25$.

structures. In the right figure, we show the results with fixed number of levels (5 here) but different $nmodes$. Although the accuracy increases with more modes selected, as expected, this increase is very gradual. For example, the bottom curve is only about 1 digit more accurate than the top one, but the size of the projected problem (see (7)) for the bottom curve is almost twice as large as that of the top curve.

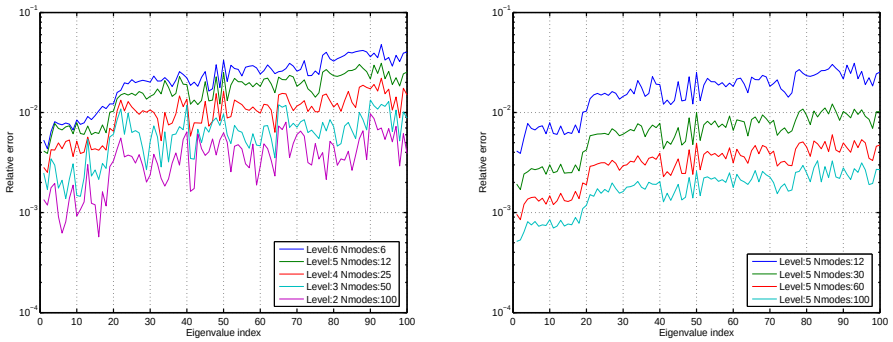

Fig. 3. Eigenvalue accuracy of DDS6. Left: increasing levels. Right: Fixed level, increasing $nmodes$.

PE3K

The low frequency vibrational modes of the PE molecule can be solved by computing the eigenvalues and eigenvectors associated with the Hessian of a potential function that describes the interaction between different atoms. For a 3000-atom molecule, the dimension of the Hessian matrix is 9000. Figure 4 shows the molecular structure of the PE particle and the sparsity pattern of the Hessian matrix after it is permuted by METIS. We observe that PE3K contains separators of large dimensions, resulting in excessive fills. This makes the SIL calculation memory intensive [13]. Our semi-implicit representation

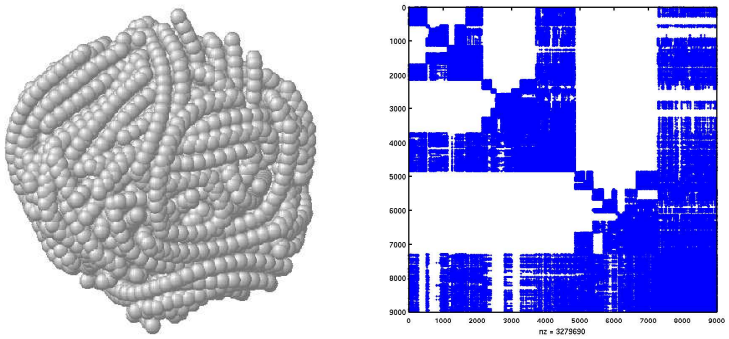

Fig. 4. The molecular structure of PE3K and the sparsity pattern of the Hessian after it is permuted by METIS.

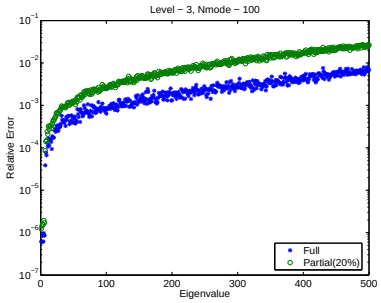

Fig. 5. Eigenvalue accuracy of PE3K, full or partial selection of interface modes.

of L greatly reduced the memory required in the AMLS calculation (saving 35% of memory). By choosing only a fraction of the coupling modes from each separator, we also reduced the dimension of the projected problem (7). In Figure 5, we compared the accuracy of a 3-level AMLS calculation in which 20% of coupling modes are computed and chosen from each separator with a 3-level calculation in which all coupling modes are selected. Both calculations used $nmodes = 100$ for each sub-structure. Figure 5 shows that the partial selection of the coupling modes does not affect the accuracy of the AMLS calculation significantly for this problem. It is important to note that choosing 20% of coupling modes enables us to reduce the AMLS runtime from 1776 to 581 seconds.

4 Conclusions and related work

When a large number of eigenvalues with a few digits of accuracy are wanted, the multilevel sub-structuring method is computationally more advantageous than the conventional shift-and-invert Lanczos algorithm. This is due to the fact that AMLS does not have the bottlenecks associated with the reorthogonalization and triangular solve. However, when the accuracy requirement is

high, AMLS becomes less appealing. Some research is underway to address the accuracy issue. We are developing better mode selection criteria so that the projected subspace retains better spectral information from (K, M) while its size is still restricted. Bekas and Saad [1] suggests to enhance the algorithm by using spectral Schur complements with higher order approximations. Further evaluation is needed to determine the effectiveness of these strategies.

References

1. C. Bekas and Y. Saad. Computation of smallest eigenvalues using spectral schur complements. Technical Report UMSI-2004-6, University of Minnesota, 2004.
2. 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, AMSE, New York, November, 1993.
3. J. K. Bennighof and R. B. Lehoucq. An automated multilevel substructuring method for eigenspace computation in linear elastodynamics. *SIAM Journal on Scientific Computing*, 25(6):2084-2106, 2004.
4. R. R. Craig and M. C. C. Bampton. Coupling of substructures for dynamic analysis. *AIAA Journal*, 6:1313–1319, 1968.
5. W. Gao, X. Li, C. Yang, and Z. Bai. Performance evaluation of an gebraic substructuring method for sparse eigenvalue problems. Technical Report LBNL-57438, Lawrence Berkeley National Laboratory, 2005.
6. W. C. Hurty. Vibrations of structure systems by component-mode synthesis. *Journal of the Engineering Mechanics Dvision, ASCE*, 86:51–69, 1960.
7. M. F. Kaplan. *Implementation of Automated Multilevel Substructuring for Frequency Response Analysis of Structures.* PhD thesis, University of Texas at Austin, Austin, TX, December 2001.
8. 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.* University of Minnesota, 1998.
9. K. Ko, N. Folwell, L. Ge, A. Guetz, V. Ivanov, L. Lee, Z. Li, I. Malik, W. Mi, C. Ng, and M. Wolf. Electromagnetic systems simulation - from simulation to fabrication. SciDAC report, Stanford Linear Accelerator Center, Menlo Park, CA, 2003.
10. Esmond G. Ng and Barry W. Peyton. Block sparse Cholesky algorithms on advanced uniprocessor computers. *SIAM J. Sci. Comput.*, 14(5):1034–1056, September 1993.
11. B. N. Parlett. *The Symmetric Eigenvalue Problem.* Prentice-Hall, 1980.
12. C. Yang, W. Gao, Z. Bai, X. Li, L. Lee, P. Husbands, and E. Ng. An algebraic sub-structuring method for large-scale eigenvalue calculation. Technical Report LBNL-55050, Lawrence Berkeley National Laboratory, May 2004.
13. C. Yang, B. W. Peyton, D. W. Noid, B. G. Sumpter, and R. E. Tuzun. Large-scale normal coordinate analysis for molecular structures. *SIAM Journal on Scientific Computing*, 23(2):563–582, 2001.