

Research Article

Some Progress on Parallel Modal and Vibration Analysis Using the JAUMIN Framework

Xuanhua Fan,¹ Keying Wang,¹ Shifu Xiao,¹ Qingkai Liu,² and Zeyao Mo²

¹Institute of Systems Engineering, CAEP, Mianyang 621900, China

²Institute of Applied Physics and Computational Mathematics, Beijing 100094, China

Correspondence should be addressed to Zeyao Mo; zeyao.mo@iapcm.ac.cn

Received 26 August 2014; Accepted 3 October 2014

Academic Editor: Song Cen

Copyright © 2015 Xuanhua Fan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In the development of large and complex equipment, a large-scale finite element analysis (FEA) with high efficiency is often strongly required. This paper provides some progress on parallel solution of large-scale modal and vibration FE problems. Some predominant algorithms for modal and vibration analysis are firstly reviewed and studied. Based on the newly developed JAUMIN framework, the corresponding procedures are developed and integrated to form a parallel modal and vibration solution system; the details of parallel implementation are given. Numerical experiments are carried out to evaluate the parallel scalability of our procedures, and the results show that the maximum solution scale attains ninety million degrees of freedom (DOFs) and the maximum parallel CPU processors attain 8192 with favorable computing efficiency.

1. Introduction

In many engineering applications such as aerospace, automotive, and numerous other large equipment, the FE modal and vibration analysis are important ways to characterize their dynamic properties. However, the FE computing scale and the efficiency of commercial software often become a bottle to solve these complex problems. Therefore, developing large-scale and high efficient FE procedures is strongly required.

Mathematically, modal analysis is treated as computing a number of lower eigenpairs of the generalized eigenvalue problems:

$$\mathbf{K}\mathbf{x} = \lambda\mathbf{M}\mathbf{x}, \quad (1)$$

where \mathbf{K} and \mathbf{M} are stiffness and mass matrix, respectively. For the FE problems of small DOFs (below one million, for example), the equation can be easily solved via traditional methods provided by commercial software, but for larger scale cases, the commercial software is inaccessible. Both algorithms and parallel computing techniques should be therefore considered.

Modal analysis takes an important role in dynamic computation, which is also the base of many other vibration analysis types [1, 2]. For large scale vibration analysis

problems, the mode superposition method [3] is usually a feasible and effective way to obtain the dynamic response of structures. We ourselves focus on the random vibration analysis in this paper.

Considering the parallel implementation of these methods for modal and vibration analysis, the design of framework is very important. Currently, many frameworks, such as SIERRA [4], PHG [5], and JASMIN [6], have been successfully developed or are being actively developed. The basic idea of these frameworks is providing common data structures, parallel strategies, and interface for various application programs. Framework has been proven to be an effective approach to develop parallel programs for different massive computers.

Aimed at large scale parallel computation of modal and vibration analysis, this paper gives some progress which is based on our newly developed JAUMIN (J Adaptive Unstructured Mesh application INfrastructure) framework. The rest of the paper is organized as follows. Section 2 gives a brief review of the eigenvalue algorithms and mode-based random vibration theory. Section 3 introduces the parallel implementation, including the data structures, the parallel strategy, and programming interface of JAUMIN framework.

A representative example with different scales is given to verify the parallel scalability of our application procedures and JAUMIN framework in Section 4. Finally, the paper ends with a brief conclusion in Section 5.

2. Algorithm and Theory for Modal and Vibration Analysis

Since only a few lower eigenpairs are of interest for modal analysis, subspace-based algorithms are usually the best choice for large sparse eigenvalue problems. Subspace-based methods are mainly based on matrix-vector multiplications using the original sparse matrix so that the sparse matrix storage and structures can be used to advantage [7]. When modal analysis is finished, the mode superposition method can be adopted to carry out the following vibration analysis.

2.1. A Brief Survey of Subspace-Based Methods for Modal Analysis. Subspace-based methods differ from each other in the ways the subspaces are generated. The dimension of subspaces can be either fixed or variable. The classical algorithms working with fixed dimensions mainly include power method, Rayleigh quotient iteration, and subspace iteration [7, 8].

A further class of subspace methods is the class of those whose dimensions increase as the iteration proceeds. Usually one starts with a subspace of dimension one and increases the dimension at each iteration step. These methods are in general more efficient than fixed-dimension ones and have become the mainstream for large-scale eigenvalue problems. This class mainly includes two subclasses, the Krylov-based subspace methods and the Davidson-based subspace methods [7–9].

The Krylov subspace method can be traced back to the Lanczos method [10] for symmetric matrices and the Arnoldi method [11] for nonsymmetric matrices. Since Lanczos and Arnoldi use subspaces of increasing dimension, storage and computing time is increased during the algorithm. In order to overcome this disadvantage, many restarted Lanczos and Arnoldi methods have been developed. A significant improvement of these methods appertains to Sorensen's implicitly restarted Arnoldi/Lanczos (IRA) method [12]. Later, Stewart [13] proposed the Krylov-Schur (K-S) method by expanding Arnoldi decomposition to a general Krylov decomposition. The implicitly restarted Arnoldi/Lanczos method and the Krylov-Schur method are mathematically equivalent, which are recognized as the most successful algorithms in Krylov subspace methods.

Instead of using Krylov subspace, the Davidson-based methods increase subspace dimension with a Newton iteration step or an approximate Newton iteration step. Based on the standard Davidson method [14], some generalized Davidson methods [7, 9, 15] were presented by using different preconditioning. In 1996, Sleijpen and Van der Vorst [16] presented a Jacobi-Davidson (J-D) algorithm, which speeded up the development of Davidson-based methods. So far, the Jacobi-Davidson method has been extended to various eigenvalue problems. For more about subspace-based methods in modal analysis, we refer to [17].

2.2. Some Discussion on Subspace-Based Methods. For the Krylov subspace methods, such as IRA and K-S, a spectral transformation is needed to get faster convergence to lower eigenpairs. With a Shift-Invert spectral transformation, (1) will become

$$(\mathbf{K} - \sigma\mathbf{M})^{-1}\mathbf{M}\mathbf{x} = \frac{1}{\lambda - \sigma}\mathbf{x}, \quad (2)$$

where σ denotes a user-defined shift value. Due to the spectral transformation, the solution of a linear system with the form $(\mathbf{K} - \sigma\mathbf{M})\mathbf{y} = \mathbf{M}\mathbf{z}$ is needed in each subspace iteration [17]. The coefficient matrix $\mathbf{K} - \sigma\mathbf{M}$ is usually ill-conditioned or even nearly singular because of the shift value σ . Thus an iterative solution to (2) becomes difficult [7, 9]; instead a direct method is often recommended. The direct matrix decomposition (such as LU or Cholesky) of $\mathbf{K} - \sigma\mathbf{M}$ brings both merits and shortcomings. Firstly, the matrix decomposition gives a fast computation of modal analysis with fewer subspace iterative steps. Secondly, the maximum matrix scale that can be decomposed is limited (about ten million DOFs [17]), and the matrix factorization needs a vast demand of memory storage and parallel communication, which results in a relatively poor parallel scalability.

For the Davidson-based methods, especially for the J-D method, the corresponding solution of modal analysis becomes an inner-outer iteration. The inner iteration, which is the kernel of J-D method, becomes the iterative solving of a "correction equation":

$$(\mathbf{I} - \mathbf{M}\mathbf{u}_k\mathbf{u}_k^*)(\mathbf{K} - \theta_k\mathbf{M})(\mathbf{I} - \mathbf{u}_k\mathbf{u}_k^*\mathbf{M})\mathbf{t} = -\mathbf{r}_k, \quad (3)$$

where θ_k and \mathbf{u}_k denote the Ritz value and Ritz vector, respectively, \mathbf{t} is an orthogonal component vector to be solved, $*$ denotes conjugate transpose, and \mathbf{r}_k is the corresponding residual vector; that is, $\mathbf{r}_k = (\mathbf{K} - \theta_k\mathbf{M})\mathbf{u}_k$. Thus, the vector \mathbf{t} in (3) is adopted to expand the search subspace (i.e., outer iteration).

Using inner-outer iterations, the J-D method takes lower memory storage since no matrix decomposition like Krylov space methods is carried out. Furthermore, the main mathematical operations in J-D are matrix-vector multiplications and vector inner products, which maintain the sparse characteristics of original matrices and greatly reduce the demand of parallel communication.

2.3. Mode-Based Random Vibration Theory. Once the required eigenpairs for modal analysis are obtained, various following vibration analyses can be carried out using mode superposition method. Here we ourselves focus on the random vibration analysis of single-point support motion; other kinds of vibration analysis will be carried out in future step by step.

The dynamic equation of random vibration analysis for support excitation via FE method can be described as

$$\mathbf{M}\ddot{\mathbf{x}}_r + \mathbf{C}\dot{\mathbf{x}}_r + \mathbf{K}\mathbf{x}_r = -\mathbf{M}\mathbf{d}\ddot{u}, \quad (4)$$

where \mathbf{C} denotes damping matrix, \mathbf{x}_r denotes the relative displacement against support motion, \mathbf{d} is a direction vector composed by the direction cosine between the motion

direction and the three coordinate axes for each node, and u is the displacement value of support motion. Thus, the real displacement \mathbf{x} can be expressed as

$$\mathbf{x} = \mathbf{x}_r + \mathbf{d}u. \quad (5)$$

Using the mode superposition method [3], (4) can be decoupled by the m eigenpairs (ω_i, ϕ_i) obtained by modal analysis (the eigenvectors are normalized with \mathbf{M}), which yields

$$\ddot{\eta}_i + 2\omega_i\xi_i\dot{\eta}_i + \omega_i^2\eta_i = -\gamma_i\ddot{u} \quad (i = 1, 2, \dots, m), \quad (6)$$

where η_i is the i th modal displacement, ξ_i is the i th modal damping ratio, and γ_i is the i th modal participation factor with the definition $\gamma_i = \phi_i^T \mathbf{M} \mathbf{d}$. Modal participation factors are indicators of the possible contributions of particular modes towards the stochastic response of the structure, and the corresponding computation concerns operations of matrix-vector multiplications and vector inner products, which can be implemented with parallel data structures.

For (6), the corresponding modal power spectral density (PSD) $S_{\eta_i\eta_j}(\omega)$ can be given as

$$S_{\eta_i\eta_j}(\omega) = \gamma_i\gamma_j H_i(\omega) H_j^*(\omega) S_{\ddot{u}\ddot{u}}(\omega), \quad (7)$$

where

$$H_i(\omega) = \frac{1}{\omega_i^2 - \omega^2 + i(2\zeta_i\omega\omega_i)}, \quad (8)$$

and $S_{\ddot{u}\ddot{u}}(\omega)$ denotes PSD of acceleration, which is the input load on the base. Equation (7) builds the relationship between modal PSD and input PSD.

Applying the mode combination in mode superposition method, we get the PSD of \mathbf{x}_r for the k th DOF as

$$S_{x_r k x_r k}(\omega) = \left| \sum_{i=1}^m \gamma_i \phi_{ik} H_i(\omega) \right|^2 S_{\ddot{u}\ddot{u}}(\omega), \quad (9)$$

where m is the modal orders for combination and ϕ_{ik} denotes the value of the k th DOF in the i th modal shape ϕ_i .

Considering relationship between the relative displacement and the absolute displacement in (5), we deduce the expression of PSD of \mathbf{x} for the k th DOF as

$$\begin{aligned} S_{x_k x_k}(\omega) &= S_{x_r k x_r k}(\omega) \\ &+ \frac{2d_k}{\omega^2} \sum_{i=1}^m \gamma_i \phi_{ik} H_i(\omega) \Big|_{\text{Re}} S_{\ddot{u}\ddot{u}}(\omega) \\ &+ \frac{d_k^2}{\omega^4} S_{\ddot{u}\ddot{u}}(\omega) \\ &= \left(\left| \sum_{i=1}^m \gamma_i \phi_{ik} H_i(\omega) \right|^2 + \frac{2d_k}{\omega^2} \sum_{i=1}^m \gamma_i \phi_{ik} H_i(\omega) \Big|_{\text{Re}} \right. \\ &\quad \left. + \frac{d_k^2}{\omega^4} \right) S_{\ddot{u}\ddot{u}}(\omega), \end{aligned} \quad (10)$$

where d_k denotes the value of the k th DOF in direction vector \mathbf{d} . Equation (10) gives the relationship between response PSD of the k th freedom and the input PSD of base motion.

Equation (10) is a perfect expression. Firstly, the implementation of (10) mainly becomes computing a component or subclass:

$$\sum_{i=1}^m \gamma_i \phi_{ik} H_i(\omega), \quad (11)$$

which physically represents the transfer function between output relative displacement and input base acceleration; secondly, the response of the k th DOF is absolutely independent of others. Thus, the PSD computation of each DOF in (10) is a natural parallel pattern, without any parallel communication occurring. In other words, the parallel communication for random vibration analysis only occurs at the stage of computing modal participation factors.

3. Parallel Implementation in JAUMIN Framework

The algorithm and theory mentioned in Section 2 are the kernel to realize the modal and vibration analysis of large engineering structures. The parallel implementation of modal analysis mainly depends on the eigenvalue solver package SLEPc [18] integrated in JAUMIN [19] framework, and the parallel implementation codes of random vibration are developed by ourselves.

3.1. JAUMIN Framework. JAUMIN is a parallel computing framework which has been developed by China Academy of Engineering Physics since 2011. JAUMIN framework supports the large scale parallel simulations on adaptive unstructured meshes using massively parallel processing (MPP) machines for the FEA of engineering structures; it is a complement to the formerly developed JASMIN [6] framework, which aims at structured mesh applications in fluid mechanics.

The software architecture of JAUMIN is based on multi-layered, modularized, and object-oriented design thoughts, mainly being written with C++ and MPI, including about 12 thousand of lines so far. JAUMIN can be installed on personal computer, cluster, and MPP machines with a UNIX or LINUX system. The architecture of JAUMIN framework can be described in Figure 1.

In JAUMIN framework, the bottom layer is high performance computation supporting layer, which encapsulates geometry, mesh, and field data structures, providing parallel communication, load balancing, adaptive mesh refinement, and basic mathematical operations; the middle layer is numerical general layer, which provides fast solvers, mature numerical methods and time integrators, and so forth. The top layer provides programming interfaces for different applications.

3.2. Data Structures and Parallel Communications. The element types in JAUMIN framework include most of polygons

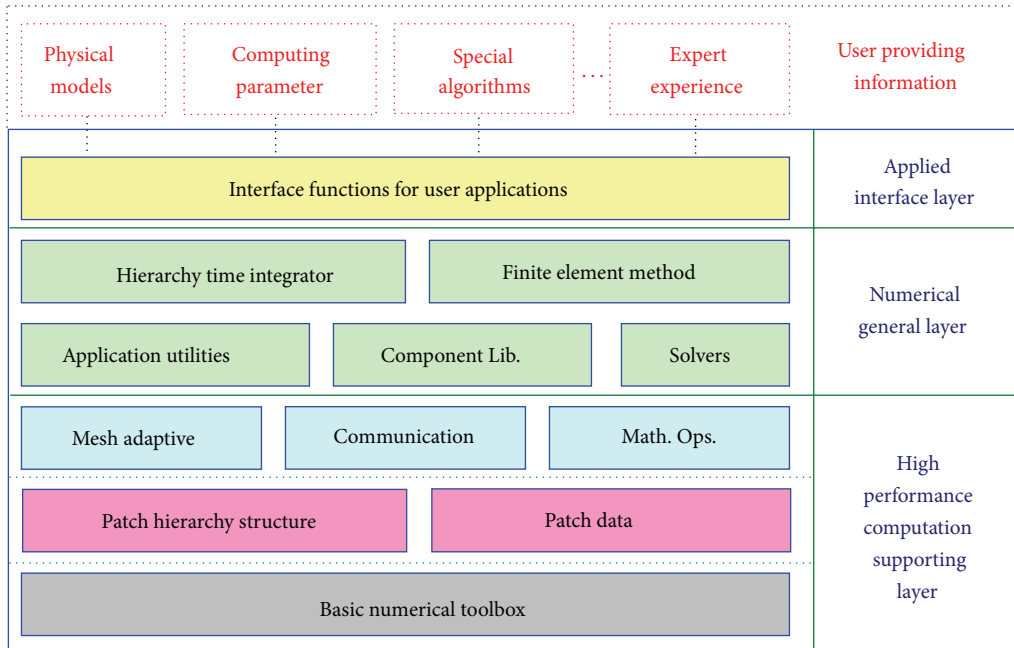


FIGURE 1: The architecture of JAUMIN framework.

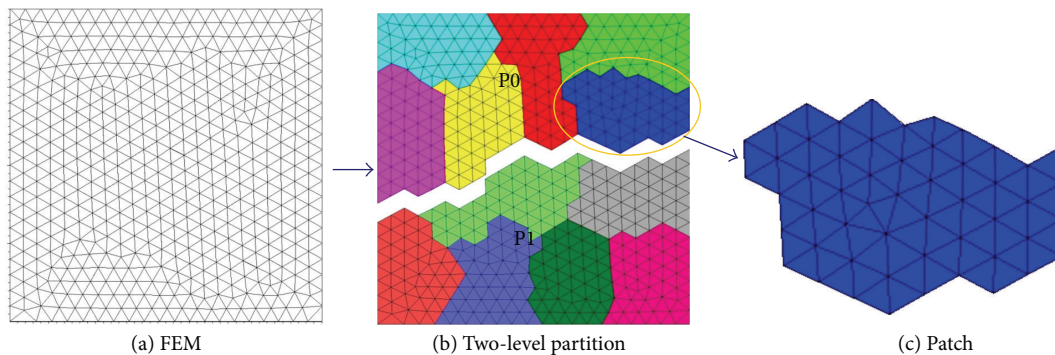


FIGURE 2: A two-level partitioning strategy in JAUMIN framework.

for two-dimensions and most of polyhedrons for three-dimensions, such as triangle, quadrangle, tetrahedron, pyramid, and hexahedron. Furthermore, hybrid meshes are also supported in JAUMIN.

Considering parallel computing environment, JAUMIN adopts a two-level partitioning strategy to partition the FEM, shown as in Figure 2. Firstly, the FEM is partitioned into submeshes considering load balance; each CPU processor is assigned with a submesh. Secondly, each submesh is again partitioned into a group of smaller submeshes, named as “patch.” Instead of element, patch is the basic object for memory management and numerical computation in JAUMIN, which make it easier to implement MPI + OpenMP hybrid parallel programming and computing. On the other hand, the hierarchical data structure can be well matched with the predominant multicore computer architectures. The first level (P0, P1 in Figure 2) is partitioned and the corresponding

information is sent to each CPU processor; and the second level (i.e., patch) is portioned based on the first level and corresponding information can be sent to the inner cores of each CPU processor if necessary. Each patch includes some basic information such as patch geometry, patch topology, and patch data, which constitutes the core data structures of the framework.

According to the data structures in JAMIN framework, the parallel communication module is designed with two kinds of communication modes: one is the data transferring among the first levels, which can be implemented by MPI; the other is the data transferring among the patches for each level, which can be implemented by OpenMP. In our implementation of mode and vibration analysis, only MPI is used for the parallel communication among levels, and the patch data are used only to form the distributing FE matrices for each level.

3.3. Interface Service and Parallel Implementation for Dynamic Applications. Besides the basic data structures and parallel communication, the JAUMIN frame provides some interfaces to pre/postprocessing and the individual solvers. Using the preprocessing interface, the FEM of engineering structures is read to the JAUMIN framework. Currently, we build the FEM of engineering structures via a self-developed procedure, named as SuperMesh, which can implement the FE meshing and support adaptive refinement.

Based on the FEM information, a two-level domain decomposition mentioned above is implemented in JAUMIN framework to partition the FEM into some smaller subdomains. The information of each subdomain is delivered to different processors for FE discretization and numerical integral, until forming the distributing matrices \mathbf{K} and \mathbf{M} .

Another important interface in JAUMIN framework is the solver interface for different applications. As far as FEA is concerned, JAUMIN provides interfaces to linear/nonlinear static analysis, modal analysis, harmonic response analysis, random vibration response analysis, and some kinds of shock response analysis currently. We ourselves focus on the modal and vibration analysis in this paper.

The implementation of modal analysis mainly depends on an important software package, that is, SLEPc [18], the Scalable Library for Eigenvalue Problem Computations, which is taken as a third solver library. SLEPc provides a collection of eigensolvers, including Krylov-Schur, Jacobi-Davidson, Arnoldi/Lanczos, and some other subspace methods. Most of the parallel implementations in SLEPc are vector operations, the matrix-vector product and linear equation solvers, which are supported by another numerical toolbox PETSc [20]. The JAUMIN framework generates parallel distributing stiffness matrix \mathbf{K} and mass matrix \mathbf{M} and makes them as the input for SLEPc to finish the parallel solution of modal analysis. The JAUMIN framework is responsible for the calling and managing of these software packages.

The codes for mode-based random vibration analysis are developed by ourselves using the corresponding theory shown in (4)~(11), which can be treated as a postcomputing process of modal analysis. A crucial step in random vibration is calculating the modal participation factors, which requires matrix-vector products and vector operations with the JAUMIN parallel data structures. Another parallel process in random vibration analysis lies in the computation of PSD for different nodes. As mentioned in Section 2.3, this process is a natural parallel one without any communication. Each processor is responsible for the corresponding computing nodes that locate in it. Besides the PSD of relative displacement and absolute displacement as shown in (9) and (10), PSD of velocity (denoted by $S_{\dot{x}_k \dot{x}_k}$) and that of acceleration (denoted as $S_{\ddot{x}_k \ddot{x}_k}$) can also be calculated by relationships:

$$S_{\dot{x}_k \dot{x}_k} = \omega^2 S_{x_k x_k}, \quad S_{\ddot{x}_k \ddot{x}_k} = \omega^4 S_{x_k x_k}, \quad (12)$$

which have been implemented in our codes.

We want to emphasize that the vibration analysis in this paper is limited to the type of single-point support random excitations. The case of single-point nodal load excitation is similar to the calculation of PSD of relative displacement in

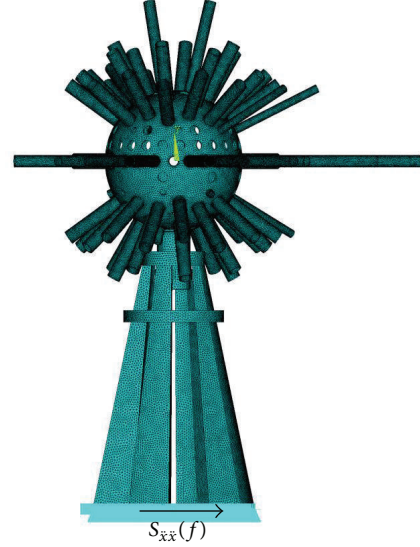


FIGURE 3: FEM of target ball.

(9), where the input PSD $S_{i\ddot{u}i}(\omega)$ on the right side is replaced with the PSD of force $S_{ff}(\omega)$, and the response PSD $S_{x_{rk}x_{rk}}(\omega)$ on the left side is replaced with $S_{x_kx_k}(\omega)$ since no relative displacement occurs. In this case, the implementation of single-point nodal load excitation is easier than single-point support excitations. For the implementations of some other cases such as multipoint excitation calculations [21, 22] are being developed in JAUMIN framework.

4. Numerical Experiments

Many simple engineering examples have been adopted to validate the rightness of our programs, and some of them were compared with commercial software ANSYS or MSC.Nastran, in which the same computing results are obtained. Here we give a representative example in engineering structures to evaluate the rightness and parallel scalability of the modal and vibration analysis in JAUMIN framework. The computations were carried out on a supercomputer, which include thousands of blade computing nodes, each of them with twelve Intel processors and 48 GB of shared memory.

The engineering example is the target ball structure of a laser equipment, whose FEM is shown in Figure 3. The model is meshed with tetrahedron elements, and the node displacements at the bottom of the model are fixed as a boundary constraint. The initial DOFs of the model are about 1.1 million, and an adaptive mesh refinement technique is adopted to increase the numbers of DOFs. After the first refinement, the DOFs attain 12, 214, 455, and after the second refinement, the DOFs attain 91, 207, 359.

As there are a large number of similar pipes in the structures, there exist numerous dense modes in the modal analysis, which obviously enhances the difficulty of numerical computation. Here, the J-D method is adopted for modal

TABLE 1: Parallel performance of modal and vibration analysis for 12, 214, 455 DOFs.

Processors	64	128	256	512	1024	2048	4096	6144	8192
Time (s)	79166	47875	32014	16613	9524	5646	3992	3014	2708
% Efficiency	100	82.7	61.8	59.6	52.0	43.8	31.0	27.4	22.8

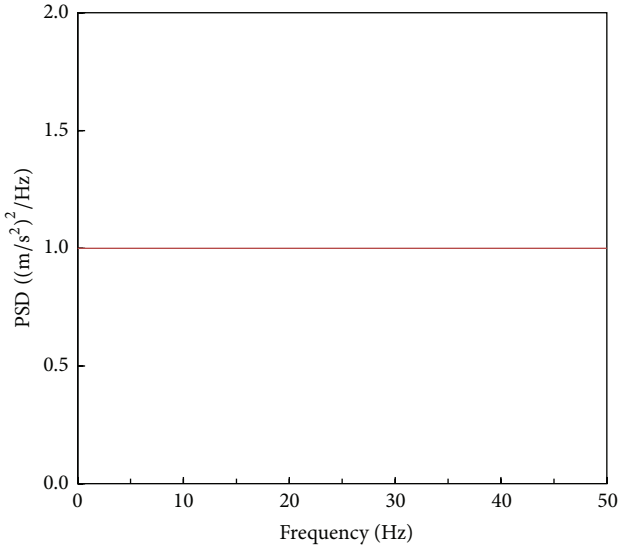


FIGURE 4: PSD of base acceleration.

analysis; the lowest 100 eigenpairs are desired for modal and random vibration analysis. For random vibration analysis, an acceleration excitation of base motion in the X direction is adopted, with the corresponding PSD shown in Figure 4.

4.1. Analysis and Validation of the Initial Model. For the initial meshes (1.1 million DOFs) of the model, the modal and vibration analysis are also carried out by ANSYS to validate the rightness of our programs. The results show that the 100 orders of eigenpairs calculated in JAUMIN framework are coincident with that in ANSYS; the largest relative error for eigenvalues is below 0.05%, and the modal shapes are identical. Figure 5 gives a comparison of the first-order modal shape calculated in ANSYS and JAUMIN.

Furthermore, using the results of modal analysis, the following mode-based random vibration analysis is also carried out and compared with ANSYS. Figure 6 gives the comparison of PSD of displacement in the X direction of the apogee. We can see from Figure 6 that the two curves are absolutely coincident in the range of analysis frequencies, which validates the rightness of both modal and vibration analysis from another point of view.

4.2. Testing on Parallel Scalability. Based on the initial FEM, the target ball is refined twice to increase the mesh density. As mentioned above, the total DOFs attain 12, 214, 455 after the first refinement (Case 1) and attain 91, 207, 359 after the second refinement (Case 2). For these two scales, the traditional commercial software is incapable of action.

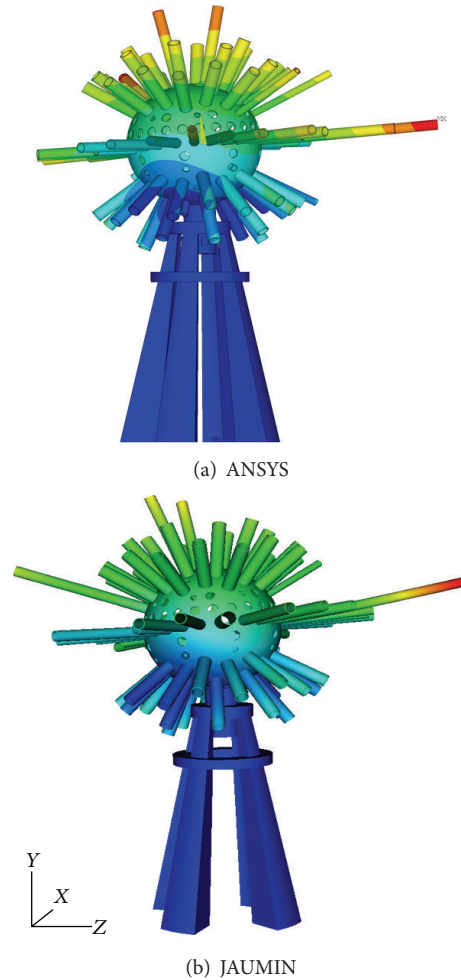


FIGURE 5: The comparison of the first-order modal shape.

TABLE 2: Parallel performance of modal and vibration analysis for 91, 207, 359 DOFs.

Processors	4096	6144	8192
Time (s)	32316	23945	18867
% Efficiency	100	90.0	85.6

The modal and random vibration analyses are carried out for Cases 1 and 2 with a maximum number of processors of 8192. Since the modal analysis is included in the random vibration analysis, we count the total computing time for random vibration analysis. Tables 1 and 2 gave the parallel computing time and efficiency with different numbers of processors for Cases 1 and 2.

From Tables 1 and 2 we can find that the parallel computing time continuously decreases as the increase of

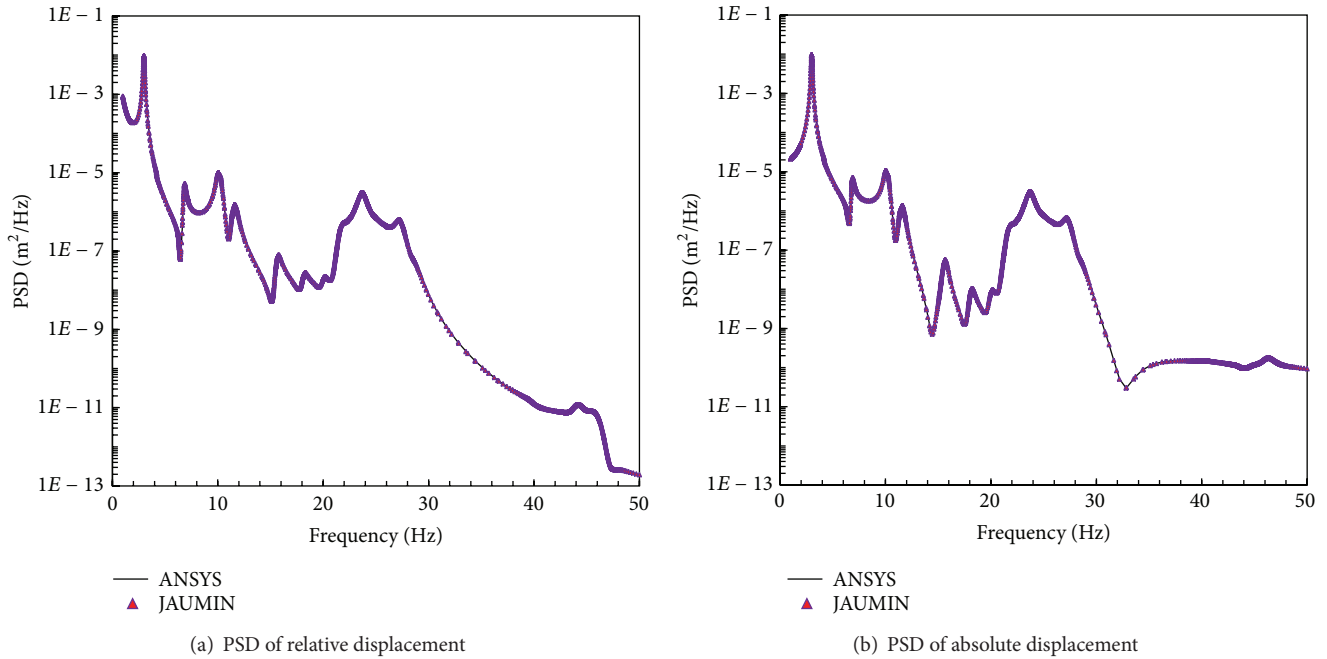


FIGURE 6: Comparison of PSD between ANSYS and JAUMIN framework.

processors, without computing inflexion, emerges. Based on the JAUMIN framework, our application programs behave a favorable parallel scalability. Furthermore, we believe that solving such a modal and vibration analysis problem near 100 million of DOFs is very challenging so far.

5. Conclusions

In this paper, we introduce some progress for the large-scale modal and vibration analysis based on the JAUMIN framework. The predominant algorithms and theory for modal and vibration analysis are firstly reviewed and deduced. A detailed introduction of JAUMIN framework is given and, based on which, an integrated parallel computing system for modal and vibration analysis is presented.

The numerical experiments via a typical engineering structure validate the rightness and scalability of our procedures for large-scale modal and vibration analysis. In virtue of the developed parallel solving system, we have elevated the computing scale of modal and vibration analysis to a new level with favorable parallel performance. Moreover, our procedures of the modal and vibration analysis fit most elastic structures coming from various engineering realms.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this manuscript.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (no. 11472256) and the fundamental

development foundation of China Academy of Engineering Physics (2012A0202008). The corresponding author also thanks some very valuable suggestions from Professor Chen Pu in Peking University.

References

- [1] J. Humar, *Dynamics of Structures (3rd edition)*, CRC Press/Balkema, Leiden, The Netherlands, 2012.
- [2] D. E. Newland, *Mechanical Vibration Analysis and Computation*, Dover, New York, NY, USA, 2006.
- [3] T. Itoh, "Damped vibration mode superposition method for dynamic response analysis," *Earthquake Engineering & Structural Dynamics*, vol. 2, no. 1, pp. 47–57, 1973.
- [4] C. Edwards and J. R. Stewart, "SIERRA: a software environment for developing complex multi-physics applications," in *Proceedings of the 1st MIT Conference on Computational Fluid and Solid Mechanics*, K. J. Bathe, Ed., Elsevier Science, Amsterdam, The Netherlands, 2001.
- [5] L. B. Zhang, Parallel Hierarchical Grid, http://lsec.cc.ac.cn/phg/index_en.htm.
- [6] Z. Y. Mo, A. Q. Zhang, X. L. Cao et al., "JASMIN: a software infrastructure for large scale parallel adaptive structured mesh applications," *Frontiers of Computer Science*, vol. 4, no. 4, pp. 480–488, 2010.
- [7] Z. Bai, J. Demmel, J. Dongarra et al., *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*, vol. 11 of *Software, Environments, and Tools*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, Pa, USA, 2000.
- [8] F. Xue, *Numerical solution of Eigenvalue problems with spectral transformations [Ph.D. thesis]*, University of Maryland, Maryland, Md, USA, 2009.
- [9] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, Manchester University Press, Manchester, UK, 1992.

- [10] C. Lanczos, "An iteration method for the solution of the eigenvalue problem of linear differential and integral operators," *Journal of Research of the National Bureau of Standards*, vol. 45, pp. 255–282, 1950.
- [11] W. E. Arnoldi, "The principle of minimized iteration in the solution of the matrix eigenvalue problem," *Quarterly of Applied Mathematics*, vol. 9, pp. 17–29, 1951.
- [12] D. C. Sorensen, "Implicit application of polynomial filters in a k -step Arnoldi method," *SIAM Journal on Matrix Analysis and Applications*, vol. 13, no. 1, pp. 357–385, 1992.
- [13] G. W. Stewart, "A Krylov-Schur algorithm for large eigenproblems," *SIAM Journal on Matrix Analysis and Applications*, vol. 23, no. 3, pp. 601–614, 2001/02.
- [14] E. R. Davidson, "The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices," *Journal of Computational Physics*, vol. 17, no. 1, pp. 87–94, 1975.
- [15] R. B. Morgan and D. S. Scott, "Generalizations of Davidson's method for computing eigenvalues of sparse symmetric matrices," *Society for Industrial and Applied Mathematics. Journal on Scientific and Statistical Computing*, vol. 7, no. 3, pp. 817–825, 1986.
- [16] G. L. Sleijpen and H. A. Van der Vorst, "A Jacobi-Davidson iteration method for linear eigenvalue problems," *SIAM Journal on Matrix Analysis and Applications*, vol. 17, no. 2, pp. 401–425, 1996.
- [17] X. Fan, P. Chen, R. Wu, and S. Xiao, "Parallel computing study for the large-scale generalized eigenvalue problems in modal analysis," *Science China Physics, Mechanics and Astronomy*, vol. 57, no. 3, pp. 477–489, 2014.
- [18] V. Hernandez, J. E. Roman, and V. Vidal, "SLEPc: a scalable and flexible toolkit for the solution of eigenvalue problems," *Association for Computing Machinery. Transactions on Mathematical Software*, vol. 31, no. 3, pp. 351–362, 2005.
- [19] W. B. Zhao, Q. K. Liu, and G. M. Qin, "Parallel computation for finite element method based on hierarchical data structure," *Computer Engineering & Science*, vol. 34, no. 8, pp. 107–111, 2012 (Chinese).
- [20] S. Balay, K. Buschelman, V. Eijkhout et al., "PETSc users manual," Tech. Rep., Argonne National Laboratory, 2008.
- [21] A. M. Alkhaleefi and A. Ali, "An efficient multi-point support-motion random vibration analysis technique," *Computers and Structures*, vol. 80, no. 22, pp. 1689–1697, 2002.
- [22] A. K. Chopra, *Dynamics of Structures-Theory and Applications to Earthquake Engineering*, Prentice Hall, New Jersey, NJ, USA, 4th edition, 2012.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

