Massively Parallel Computations by a General-Purpose Finite Element Analysis Code – *IPSAP*

Seung Jo Kim[1], Chang Sung Lee[2], Jeong Ho Kim[3], Jin Woo Park[4]

Summary

Introduction is made of the parallel finite element analysis code called *IPSAP* which is based on the parallel domain-wise multifrontal solver. The domain-wise multifrontal solver of *IPSAP* can overcome most of difficulties and disadvantages associated with direct solution methods in massively parallel finite element computations. By using *IPSAP*, we solved the largest problem ever solved by a direct solvers while sustaining 191 Gflop/s with 256 CPUs on our self-made cluster system, *Pegasus*. We also implemented and incorporated a block Lanczos algorithm based on the domain-wise multifrontal solver into *IPSAP* and solved eigen problems with 7 millions of DOFs within an hour. The characteristics of Active Fiber Composites was investigated through the Direct Numerical Simulation and vibration analysis of aerospace launch vehicle was carried out using *IPSAP*.

Introduction

In the finite element computations, solving of the sparse linear system of equations arising from the finite element discretization, is the most time-consuming part. To handle the huge sparse matrices effectively, numerous types of solvers have been developed which can conveniently be classified into direct and iterative solvers. Direct solvers perform direct factorization of the global stiffness matrix considering the non-zero sparse patterns of the matrix, followed by the solution of the equation by the substitution procedure. On the other hand, iterative solvers perform matrix–vector or vector–vector computations repeatedly until the solution converges within prescribed error bound.

In general, iterative methods can show greater performance than direct methods for certain types of problems. However, iterative methods do not guarantee the numerical robustness of direct solvers. This is important especially for those applications characterized by stiff systems such as finite element applications in structural and solid mechanics. Also, for general finite element applications in industries and academia, it is very important that the solution can be obtained within estimated time. For these reasons, most of the commercial finite element packages have adopted direct solvers. Direct solvers are also considerably more efficient than iterative solvers in handling problems with multiple right-hand sides such as structural analysis problems with multiple load cases and implicit time integration problems with constant stiffness matrices. The block Lanczos algorithm for eigenvalues problems is also a good example that can benefit from the efficiencies inherent in direct solvers.

[1]Professor, Department of Aerospace Engineering, Seoul National University, Korea
[2]Department of Aerospace Engrg., Seoul Nat. Univ. (currently at Samsung Electronics Co. Ltd.)
[3]Korea Institute of Science and Technology Information, Korea
[4]Post Doctoral Researcher , Dep. of Aerospace Engrg., Seoul National University, Korea

However, when it comes to large-scale parallel computations, direct solvers suffer from poor characteristics such as relatively large storage requirements and higher rate of increase in operation counts compared to iterative methods. Furthermore, direct solvers are generally more difficult to parallelize compared to iterative solvers and require much more communications between processors resulting in poor scalability. Consequently, iterative solvers have been preferred in large-scale parallel finite element computations arena[1],[2],[3]. One of the most successful research codes for large-scale parallel finite element analysis with iterative solvers may be Salinas [3] with FETI [1], [2] style algorithm.

Despite the difficulties and disadvantages in adopting direct solvers for massively parallel finite element computations, direct methods are still more desirable for numerical robustness. In [4], an efficient implementation of the domain-wise multifrontal solver was proposed. The capability and the performance of the solver was tested and showed very high level of efficiency. In this paper, examples of massively parallel computations carried out using general finite element program, *IPSAP* (Internet Parallel Structural Analysis Program) are provided. An efficient implementation of the domain-wise multifrontal solver constitutes the solver module of the *IPSAP* to obtain high level of performance and numerical robustness. The characteristics of smart structures, AFC(Active Fiber Composites) [5] was investigated through the DNS (Direct Numerical Simulation) approach [6] and the vibration analysis of an aerospace launch vehicle was successfully carried out using the *IPSAP* on our self made cluster system *Pegasus*, with 400 Intel Xeon processors.

IPSAP based on the domain-wise multifrontal solver

The domain-wise multifrontal solver is at the core of the *IPSAP*, the finite element code for massively parallel structural finite element analysis. Therefore, the domain-wise multifrontal solver is discussed mainly.

Multifrontal techniques are generalization of the frontal technique, to deal with general sparse matrices with multiple frontal matrices [7]. Frontal methods were originally introduced as a solution procedure for the finite element method to minimize core storage requirements by eliminating the variables during the element assembly process [8]. There have been significant progress in the multifrontal methods and they are considered to be the most efficient direct solvers for the solution of general sparse system of linear equations. One of the most famous serial and parallel implementation is that of Gupta et al. [9]. However, by the generalization the advantages of frontal solvers used in conjunction with the finite element method of not having to assemble the global matrices has been lost. In addition, most of the multifrontal solvers available including [9] do not incorporate out-of-core algorithm which was used extensively in original frontal methods to minimize core memory requirements.

The idea of domain-wise multifrontal method by proposed by Kim et al. [10] extends the original frontal technique in finite element view point to use multiple fronts. Although algebraically equivalent to the generalized multifrontal methods, this approach eliminates the need to assemble the global stiffness matrix which is well suited for large-scale parallel finite element analysis. As each frontal matrix can be mapped to matrix from finite element

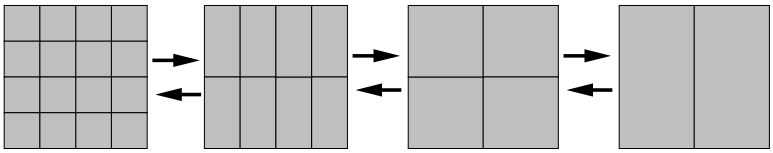

Figure 1: Illustration of the recursive substructuring procedure

domain, domain-wise parallelism can be accomplished by decomposing the finite element domain into multiple subdomains. This is much more advantageous than redistribution of the global matrix after assembly associated with multifrontal solvers for general sparse matrices. In finite element analysis, the domain-wise multifrontal method, divides the finite element domain into two subdomains recursively until appropriate number of elements are included in each domain. The fully assembled degrees of freedom (DOFs) in each domain are eliminated by the static condensation procedure, and neighboring domains are merged together in the reverse order to form a new domain. This process is repeated recursively until all of the element subdomains are merged. Figure 1 illustrates a simple example for a 2-D finite element domain. Thus, the domain-wise multifrontal method can be regarded as a recursive substructuring technique since the elimination of internal or fully assembled DOFs of the given domain is called substructuring in of structural analysis terms.

As domain-wise multifrontal solvers build frontal matrix equation for each domain assigned to the processor, the domain-wise multifrontal solver have the same level of domain-wise parallelism as iterative solvers based on domain decomposition method. Figure 2 illustrates the domain-wise parallelism of the multifrontal solver.

Serial and parallel implementations of the domain-wise multifrontal solver

To achieve high level of performance in modern cache-based systems, it is important that the algorithm be factored to perform dense matrix operations. The recursive substructuring procedure of the domain-wise multifrontal solver can be conveniently divided into two stages. The first step which in [9] is referred to as 'extend-add' operation, is the construction of a new frontal matrix by merging old frontal matrices from neighboring domains. In the second step, assembled DOFs from the newly constructed frontal matrix equation are eliminated. It should be noticed that the most of the time is consumed in the second step. The basic matrix equation of the new domain can represented as:

$$\begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix} \begin{Bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{Bmatrix} \tag{1}$$

where the subscripts 1 and 2 respectively represents quantities related to internal or assembled DOFs and those related to external or surface DOFs. After static condensation, equation 1 is transformed to:

$$\bar{\mathbf{K}}_{22}\mathbf{u}_2 = (\mathbf{K}_{22} - \mathbf{K}_{21}\mathbf{K}_{11}^{-1}\mathbf{K}_{12})\mathbf{u}_2 = \mathbf{f}_2 - \mathbf{K}_{21}\mathbf{K}_{11}^{-1}\mathbf{f}_1 = \bar{\mathbf{f}}_2 \tag{2}$$

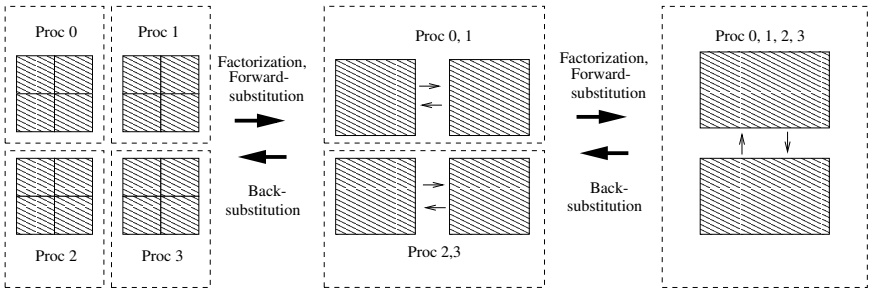

Figure 2: Domain-wise parallelism of the domain-wise multifrontal solver

The computations required to obtain $\bar{\mathbf{K}}_{22}$ can be implemented utilizing BLAS level 3 and LAPACK routines. This guarantees optimized performance along with portability.

In parallel domain-wise multifrontal solver, after the independent computations within each processor, the domains are merged across the processors. Then, dense matrix computations for substructuring are performed in parallel over related processors. Although PBLAS and ScaLAPACK are parallel implementation of BLAS and LAPACK, respectively, they cannot be adopted to the present implementation due to their requirements of the block-cyclic matrix distribution with constant block size. The parallel extend-add algorithm proposed in [9] is adopted for the domain-wise multifrontal solver. The parallel dense linear algebra subroutines are implemented using BLAS, LAPACK for dense matrix computations and BLACS for communications.

Block Lanczos eigensolver

Eigensolver is implemented using the domain-wise multifrontal solver taking use of the efficiency of handling multiple right-hand sides based on the block Lanczos algorithm. BLZPACK [11] was adopted which implements block Lanczos algorithm. For typical structural dynamics analysis represented by:

$$(\mathbf{K} - \lambda\mathbf{M})\mathbf{X} = 0 \tag{3}$$

where $\mathbf{K}$ and $\mathbf{M}$ are the stiffness and mass matrices, respectively, shifting-and-inverting technique is used to improve the solution through modifying the original equations to:

$$\mathbf{M}(\mathbf{K} - \delta\mathbf{M})^{-1}\mathbf{M}\mathbf{X} = \mu\mathbf{M}\mathbf{X} \tag{4}$$

with shifting value δ and shifted eigenvalue μ. The inverting of $(\mathbf{K} - \delta\mathbf{M})$ is replaced by factorization and substitution with multiple right-hand-side vectors.

Applications

Direct Numerical Simulation of Active Fiber Composites

Active Fiber Composites (AFC) composed of piezoelectric fibers and matrix, possess much potential for applications to adaptive structures. One major advantage of AFC are their ability to create anisotropic laminate layers for applications requiring off-axis or twisting motions. We have adopted microscopic modeling of AFC structures to accurately predict local failure mechanisms. Piezoelectric fiber and matrix were modeled separately using 3,369,600 eight-node solid elements with the total of 10,543,005 degrees-of-freedom.

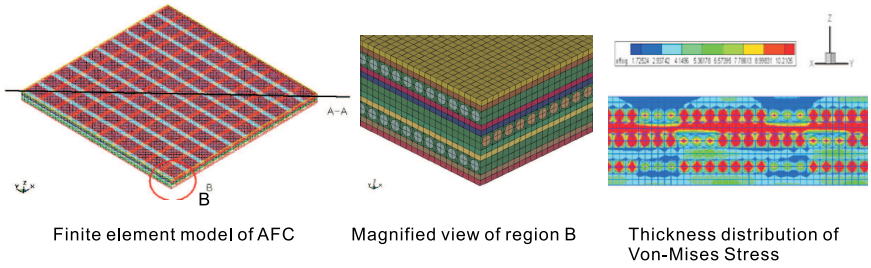

Figure 3: Finite element model and stress distribution of AFC

The elapsed time was 2,139 seconds with 256 CPUs on the *Pegasus* system. Figure 4 shows local stress distributions that could not be observed with homogenized models. Moreover, inhomogeneous non-periodic effects could be observed which cannot be observed with unit-cell approaches.

Vibration analysis of a rocket model

A full-scale vibration analysis of an aerospace launch vehicle was carried out using *IPSAP*. The ATLAS V500 launch vehicle was modeled with 255,550 solid elements with the total number of DOFs of 1,201,511. The total of 20 eigenvalues and eigenvectors were extracted with 64 processors of the *Pegasus* system.

The Salinas program based on FETI-DP [1, 2] considered as the state-of-the-art among iterative solvers, took less than 10 minutes to solve a dozen of eigen modes of a million DOF model using 3,000 processors [3]. However, in practical view, such computing capacity is not easy to come by. The present direct solver based methods running on cost-effective cluster systems can be regarded as the practical alternative.

Concluding remarks

In this work, general finite element code, *IPSAP* based on the domain-wise multifrontal solver for massively parallel finite element analysis was presented. The domain-wise multifrontal solver has provided scalability and numerical robustness for large-scale parallel finite element analysis. The local stress distribution of an Active Fiber Composite (AFC) structure was investigated using the DNS approach. Also, vibration analysis of an aerospace launch vehicle was carried out using *IPSAP* on the self-made *Pegasus* cluster system.

Figure 4: Finite element model and natural mode shapes of ATLAS V500 launch vehicle

Reference

1. Farhat, C., Lesoinne, M. and Pierson, K. (2000), "A scalable dual-primal domain decomposition method," *Numer. Lin. Alg. Appl.*, 7, 687-714.
2. Farhat, C., Lesoinne, M., LeTallec, P., Pierson, K., and Rixen, D. (2001), "FETI-DP: a dual-primal unified FETI method - part I: a faster alternative to the two-level FETI method," *Internat. J. Numer. Meths. Engrg.*, 50, 1523-1544.
3. Bhardwaj, M., Pierson, K., Reese, G., Walsh, T., Day, D., Alvin, K., Peery, J. Farhat, C. and Lesoinne, M. (2002),"Salinas: A scalable software for high-performance structural and solid mechanics simulations," *Proceedings of the IEEE/ACM SC2002 Conference*, Baltimore, MD, 16-22.
4. SJ Kim, CS Lee, JH Kim, M Joh, S Lee (2003), "*IPSAP*: A high-performance parallel finite element code for large-scale structural analysis based on domain-wise multifrontal solver," *Proc. IEEE/ACM SC2003 Conference*, Phoenix, AZ
5. Seung Jo Kim, Jun Seok Hwang, Seung Hoon Paik (2003) "Direct Numerical Simulation of Active Fiber Composites", SPIE's 10th Annual Symposium on Smart Structures and Material, San Diego, CA.
6. Seung Jo Kim, Chang Sung Lee, Hea Jin Yeo, Jeong Ho Kim, and Jin Yeon Cho (2002) "Direct Numerical Simulation of Composite Structures", *J. Composite Materials*, Vol. 36, No. 24. pp. 2765-2785.
7. S. Duff and J. K. Reid (1973), "The multifrontal solution of indefinite sparse symmetric linear equations," *ACM Trans. Math. Software*, 9, 302-325
8. B. M. Irons (1970) "A frontal solution program for finite element analysis," *Int. J. Numer. Mthds. Engrg.*, 2, 5-32
9. A Gupta, G Karypis and V Kumar (1989), "Highly scalable parallel algorithms for sparse Cholesky factorization on a hypercube," *Parallel Computing*, 10, 287-298.
10. J.H. Kim and S.J. Kim (1999) "A Multifrontal Solver Combined with Graph Partitioners", *AIAA Journal*, Vol 38, No.8,pp.964-970.
11. O. A. Marques, "BLZpack User's Guide," http://www.nersc.gov/~osni/#Software