High Performance Hybrid Direct-Iterative Solution Method for Large Scale Structural Analysis Problems

Min-Ki Kim* and Seung-Jo Kim**
Mechanical Aerospace Engineering, Seoul National University,
Seoul, Korea 151-742

Abstract

High performance direct-iterative hybrid linear solver for large scale finite element problem is developed. Direct solution method is robust but difficult to parallelize, whereas iterative solution method is opposite for direct method. Therefore, combining two solution methods is desired to get both high performance parallel efficiency and numerical robustness for large scale structural analysis problems. Hybrid method mentioned in this paper is based on FETI-DP (Finite Element Tearing and Interconnecting-Dual Primal method) which has good parallel scalability and efficiency. It is suitable for fourth and second order finite element elliptic problems including structural analysis problems. We are using the hybrid concept of theses two solution method categories, combining the multifrontal solver into FETI-DP based iterative solver. Hybrid solver is implemented for our general structural analysis code, IPSAP.

Key Word : hybrid solver, domain decomposition, FETI-DP, parallel computing, large scale structural analysis

Introduction

After finite element method appeared, necessity of structural analysis based on finite element has grown up in the various scientific and industrial fields. Rapid and optimal design of products needs high-fidelity FE structural analysis technique. Not only airplanes, automobiles, ships and buildings but also electric machines should be designed by using structural analysis. So high-reliable large scale structural analysis is very significant nowadays and its necessity is larger over the time is going.

High fidelity structural analysis always demands huge amount of computational power. Fortunately, cost of computation resources is cheaper via Mohr's law. However, more important thing is the software which is able to exploit high performance computing.

High performance computing is going to be on the distributed parallel computer environment passed by vector computer architecture in the several decades. Therefore, it is important to develop the structural analysis software to support new computational environment. Commercial structural analysis codes such as NASTRAN, ABAQUS, and ANSYS are widely used in the academic research and industrial fields. These codes have been developed without consideration of distributed parallel computing, so their parallel performance is very poor.

Other structural analysis codes, SALINAS in USA, GeoFEM and ADVENTURE in Japan are famous for large scale parallel structural analysis software. They use the iterative solution methods to archive good parallel efficiency and high scalability. SALINAS adapts FETI algorithm

* Ph. D. Candidate
** Corresponding author, Professor
E-mail : sjkim@snu.ac.kr Tel : 02-880-7388 FAX : 08-887-2662

[1], ADVENTURE adapts balancing Neumann-Neumann technique [2], and GeoFEM adapts ICCG(Incomplete Cholesky Conjugate Gradient) [3]. Iterative solution methods are better for the parallel computing and easier to implement than direct solution method. Thus large scale computational softwares prefer iterative solution methods rather than direct methods.

Until now, direct solution methods are preferred to iterative methods for the commercial FEM software because they have numerical robustness and guarantee to obtain solution of various complex problems in any cases. However, demand of iterative solution methods is growing up with the needs of high-fidelity large scale structural analysis problems. Direct solution methods require huge amount of computations and memory space for large scale problems larger than millions of DOF (degree of freedom). Moreover, direct solution methods are severely worse for parallel computing than iterative methods.

Although iterative solvers have good parallel efficiency, they are influenced by characteristics of material, element types, or subdomain, total problem size and its ratio, so they often converge very slowly or even fail to converge for certain structural analysis problems. Therefore hybrid concept of direct method and iterative method is worth to be considered to get accurate solution with both numerical robustness and parallel efficiency.

Our structural analysis code, IPSAP has very high performance direct solver, called multifrontal solver. Multifrontal solver is best direct solver in terms of computation and memory requirement, and parallel speedup [4]. So best performance direct solver is suitable for DD (Domain Decomposition) based iterative methods, such as Schwarz DD [5] or Balanced Neumann-Neumann DD [6]. In this work, we blend our multifrontal solver into FETI-DP [7] based iterative solver.

Hybrid Solution Method

Characteristics

Hybrid solver in IPSAP is based on FETI-DP method, which is a kind of iterative substructuring domain decomposition method. FETI type substructuring methods introduce Lagrange multiplier to enforce the continuity of subdomain interface. Physical meaning of multiplier is reaction force or flux between two substructures. It is shown in reference [8] that second order elliptic problems, the condition number of its interface problems is approximately

$$\kappa = O\left(1+\log^m\left(\frac{H}{h}\right)\right) \tag{1}$$

The most important thing is that FETI method has numerical scalability similar to algebraic multigrid methods if ratio of subdomain size to global domain size H/h is kept constant. In other words, convergence rate is nearly independent of the problem size.

FETI-DP takes advantages of original FETI method [9] and gets rid of difficulties to handle floating substructure mode from original FETI methods. The original FETI methods make effort to treat rigid body modes of substructures, whereas FETI-DP method can avoid this simply by introducing corner DOF.

Algorithm

Let Ω denote whole domain of 2^{nd} or 4^{th} order elliptic problems, Ω^s is its decomposed subdomain into N_s segments. Then, stiffness matrix, displacement and load vector of each subdomain can be represented as:

$$K^s = \begin{bmatrix} K^s_{rr} & K^s_{rc} \\ K^s_{cr} & K^s_{cc} \end{bmatrix}, \quad u^s = \begin{bmatrix} u^s_r \\ u^s_c \end{bmatrix}, \quad f^s = \begin{bmatrix} f^s_r \\ f^s_c \end{bmatrix} \tag{2}$$

The subscript ***c*** means 'corner DOF', and ***r*** means 'remainder DOF' all DOF except corner DOF of the each subdomain. Remainder DOF is partitioned into internal and interface DOF,

$$u_r^s = \begin{bmatrix} u_i^s \\ u_b^s \end{bmatrix} \tag{3}$$

The subscript ***i*** and ***b*** denote internal and interface DOF, respectively. And global vector of corner DOF is expressed as:

$$u_c = \begin{bmatrix} u_c^1 \\ u_c^2 \\ .. \\ u_c^{N_c} \end{bmatrix} \tag{4}$$

Restriction operator of the corner and remainder DOF is expressed as:

$$B_r^s u_r^s = \pm u_b^s, \qquad B_c^s u_c = u_c^s \tag{5}$$

where the sign is determined by a suitable convention. Fig. 1 shows the example of the 4 subdomains, red points are the corner DOFs, purple lines are the interface DOFs, and blue regions are the internal DOFs.

The main role of the corner DOF is to guarantee non-singularity of each subdomain. Determining corner DOF is to select DOF belonging to more than two subdomains.

Interface DOF must be satisfied continuity condition between two subdomains.

$$u_b^m - u_b^n = 0 \quad on \quad \partial\Omega^m \cap \partial\Omega^n, \quad m > n \tag{6}$$

This condition can be written as the constraint shown in (7).

$$\sum_{s=1}^{N_s} B_r^s u_r^s = 0 \tag{7}$$

Using these notations, the subdomain equilibrium equation with Lagrange multiplier is like this.

$$\begin{aligned} &K_{rr}^s u_r^s + K_{rc}^s B_c^s u_c + {B_r^s}^T \lambda = f_r^s \\ &\sum_{s=1}^{N_s} {B_c^s}^T \left({K_{rc}^s}^T u_r^s + K_{cc}^s B_c^s u_c \right) = f_c \end{aligned} \tag{8}$$

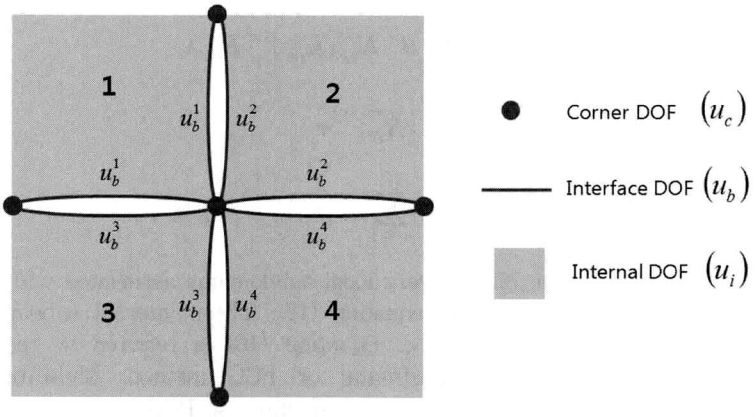

Fig. 1. example of the 4 subdomains

From equation (8), $\boldsymbol{u_r}$ is represented as

$$u_r^s = \left(K_{rr}^s\right)^{-1}\left(f_r^s - B_r^{s\,T}\lambda - K_{rc}^s B_c^s u_c\right) \tag{9}$$

Because the submatrix $\boldsymbol{K_{rr}}^s$ is invertible unlike original FETI methods. After substituting (9) into (7) and second of (8), we get a symmetric algebraic equation like this:

$$\begin{bmatrix} F_{bb} & F_{bc} \\ F_{bc}^T & -\overline{K_{cc}} \end{bmatrix}\begin{bmatrix} \lambda \\ u_c \end{bmatrix} = \begin{bmatrix} d_r \\ -\overline{f_c} \end{bmatrix} \tag{10}$$

where

$$\begin{aligned} F_{bb} &= \sum_{s=1}^{N_s} B_r^s\left(K_{rr}^s\right)^{-1} B_r^{s\,T} \\ F_{bc} &= \sum_{s=1}^{N_s} B_r^s\left(K_{rr}^s\right)^{-1} K_{rc}^s B_c^s \\ \overline{K_{cc}} &= \sum_{s=1}^{N_s}\left(K_{cc}^s - \left(K_{rc}^s B_c^s\right)^T\left(K_{rr}^s\right)^{-1}\left(K_{rc}^s B_c^s\right)\right) \\ d_r &= \sum_{s=1}^{N_s} B_r^s\left(K_{rr}^s\right)^{-1} f_r^s \\ \overline{f_c} &= f_c - \sum_{s=1}^{N_s} B_c^{s\,T} K_{rc}^{s\,T}\left(K_{rr}^s\right)^{-1} f_r^s \end{aligned} \tag{11}$$

By eliminating corner DOF $\boldsymbol{u_c}$ from (10), a symmetric positive definite interface problem follows

$$\left(F_{bb} + F_{bc}\left(\overline{K_{cc}}\right)^{-1} F_{bc}^T\right)\lambda = d_r - F_{bc}\left(\overline{K_{cc}}\right)^{-1}\overline{f_c} \tag{12}$$

Since equation (12) is symmetric positive definite, we can solve (12) by PCG (Preconditioned Conjugate Gradient). At each PCG iteration step, the matrix-vector multiplication of interface problem must be evaluated. It can be calculated following two steps:

$$\sigma = F_{bb}\lambda = \sum_{s=1}^{N_s} B_r^s\left(K_{rr}^s\right)^{-1} B_r^{s\,T}\lambda \tag{13}$$

$$\sigma = \sigma + F_{bc}\left(\overline{K_{cc}}\right)^{-1} F_{bc}^T\lambda \tag{14}$$

Second step in (14) can be split into three substeps:

$$\tau = F_{bc}^T\lambda = \sum_{s=1}^{N_s} B_c^{s\,T} K_{rc}^{s\,T}\left(K_{rr}^s\right)^{-1} B_r^{s\,T}\lambda \tag{15}$$

$$\eta = \left(\overline{K_{cc}}\right)^{-1}\tau \tag{16}$$

$$\sigma = \sigma + F_{bc}\eta = \sigma + \sum_{s=1}^{N_s} B_r^s\left(K_{rr}^s\right)^{-1} K_{rc}^s B_c^s\eta \tag{17}$$

It contains two times solution of the every local subdomains associated with $(\boldsymbol{K_{rr}}^s)^{-1}\boldsymbol{B_r}^s\boldsymbol{\lambda}$ in equation (13), (15) and $(\boldsymbol{K_{rr}}^s)^{-1}\boldsymbol{K_{rc}}^s\boldsymbol{B_c}^s\boldsymbol{\eta}$ in equation (17). These internal subdomain solution is solved by serial multifrontal solution method. Equation (16) is referred to coarse problem in FETI-DP, which can be solved by multifrontal or PCG method. Multifrontal method is numerically robust but has some weak for parallel computing, so PCG is also considered in coarse problems for large scale parallel computation.

Preconditioner

There are two preconditioners for interface problem (12). First one is so called Dirichlet preconditioner.

$$\overline{F_{bb}}^{-1} = \sum_{s=1}^{N_s} W^s B_r^s \begin{bmatrix} 0 & 0 \\ 0 & S_{bb}^s \end{bmatrix} B_r^{s\,T} W^s \tag{18}$$

where $S_{bb}{}^s$ is Schur complement of the interface DOF:

$$S_{bb}^s = K_{bb}^s - K_{ib}^{s\,T}\left(K_{ii}^s\right)^{-1} K_{ib}^s \tag{19}$$

And another one is lumed preconditioner.

$$\overline{F_{bb}}^{-1} = \sum_{s=1}^{N_s} W^s B_r^s \begin{bmatrix} 0 & 0 \\ 0 & K_{bb}^s \end{bmatrix} B_r^{s\,T} W^s \tag{20}$$

where $K_{bb}{}^s$ is subdiagonal matrix associated with interface DOF of subdomain stiffness matrix, and W^s is scaling diagonal subdomain matrix that accounts for eventual subdomain heterogeneties. Dirichlet preconditioner is more optimal in terms of mathematically than lumped preconditioner. However, lumped preconditioner is more computationally efficient than Dirichlet preconditioner, especially 3 dimensional second order problems. In this paper, lumped preconditioner is used for interface problem because of its computational efficiency.

Performance of hybrid solver

Numerical scalability

Serial performance of developed hybrid solver and multifrontal solver in IPSAP is compared in terms of elapsed time. Table 1 shows the number of iteration($N_{iteration}$) of hybrid solver and elapsed time comparison between multifrontal solver(T_{mfs}) and hybrid solver(T_{hyb}). Tested problems are 3 dimensional elasticity problems which consist of M x M x M 8 node hexahedron elements. Subdomains are partitioned geometrically into m x m x m for hybrid solver. These are run on a x64 machine which has Intel Core 2 Quad 2.66 GHz CPU and 8GB DDR2 main memory.

Table 1 represents the numerical scalability of hybrid solver if H/h is kept constant. Right of fig. 2 shows the ratio of elapsed time of multifrontal solver to hybrid solver. Left shows the scalability of multifrontal solver and hybrid solver. Fig. 2 says hybrid solver is more efficient computationally than multifrontal solver for large scale problems.

Table 1. performance comparison of multifrontal solver and hybrid solver

M (1/H)	m (1/h)	$N_{iteration}$	N_{DOF}	T_{mfs} (sec)	T_{hyb} (sec)	T_{mfs}/T_{hyb}
20	2	57	27,783	5.03	10.98	0.458
30	3	73	89,373	31.19	46.73	0.667
40	4	75	206,763	132.82	124.17	1.070
50	5	78	397,953	450.49	259.69	1.734
60	6	78	680,943	2261.78	468.83	4.824
70	7	78	1,073,733	10248.70	783.31	13.084
80	8	78	1,594,323	18582.30	1195.94	15.538
90	9	81	2,260,713	-	-	-
100	10	81	3,090,903	-	-	-

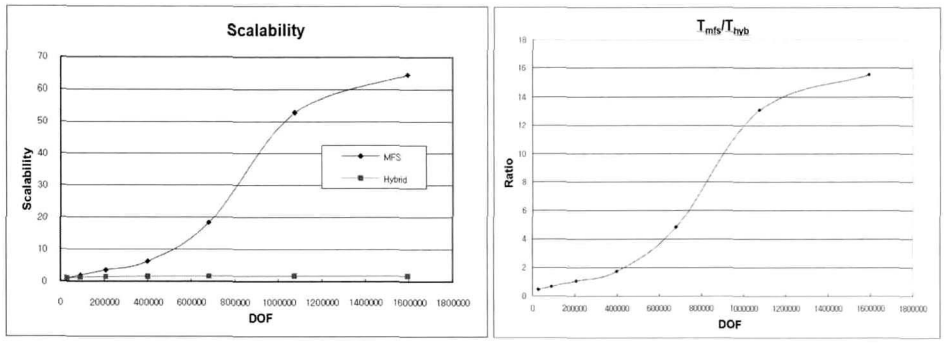

Fig. 2. Scalability (left) and time ratio (right) of multifrontal solver and hybrid solver

Parallel performance

Next, parallel scalability of hybrid solver is conducted up to 8 CPUs on PEGASUS cluster system built in 2004 by Aerospace Structure Laboratory. Benchmark problems and their results are shown in table 2 and figure 3, which is 3D structural analysis problems - X x Y x Z 8 node hexahedron element. Machines in PEGASUS system have Intel Xeon 2.2 GHz CPU and 3GB main memory. Subdomains are divided into some pieces whose geometrical profiles are 50x50x50. Coarse problem of the hybrid solution method is solved by parallel multifrontal method. In this case, solving coarse problem may be main bottleneck of parallel computing.

Extremely large scale problem near 100 million DOF

Final test problem is analysis of 320x320x320 3D structure which have near 100 million DOFs. It is conducted also on PEGASUS cluster, 64 Intel Xeon 2.2 GHz CPUs are used to solve this problem. Coarse problem of this is solved by PCG with point Jacobi preconditioner. Fig. 4 shows the convergence history of relative residual norm.

Table 2. Parallel scalability of hybrid solver

CPUs	XxYxZ	N_{DOF}	T (sec)	N_{DOF}/T
1	50x50x50	397,953	750.11	1
2	50x100x50	788,103	758.45	1.96
4	100x100x50	1,560,753	805.42	3.65
8	100x100x100	3,090,903	1,073.68	5.43

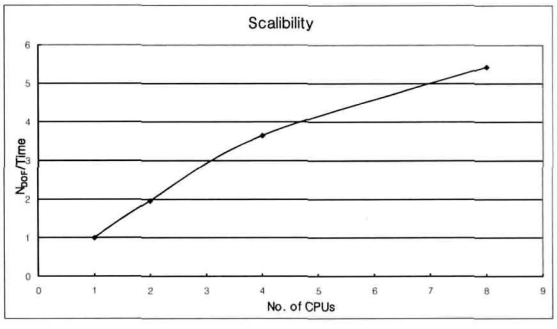

Fig. 3. Parallel scalability of hybrid solver

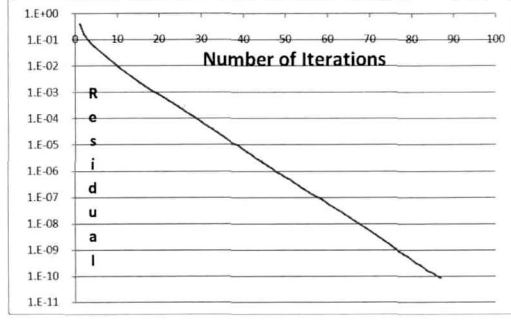

Fig. 4. Convergence of relative residual norm of 3D structure near 100 million DOFs

Applications for large scale structures

Main applications of hybrid solver are Direct Numerical Simulation (DNS). DNS is to simulate directly microscopic behaviors of the structure without any assumption such as beam or shell. Thus DNS requires huge amount of computation compared to shell or beam. First example has about 450 thousands of DOFs, solved by single CPU in PEGASUS system. This is divided into 128 pieces of subdomains. Fig. 5 shows the convergence of residual (left) and stress plot (right).

Second example is eight times larger than first example, it has about 3,713,328 DOFs. Fig. 6 shows the finite element model (left) and σ_{xx} stress components plot (right) and fig. 7 represents the convergence of second example. It is solved by just 8 CPUs in PEGASUS system, whereas 64 CPUs should be used if multifrontal solver is adapted to analyze this problem.

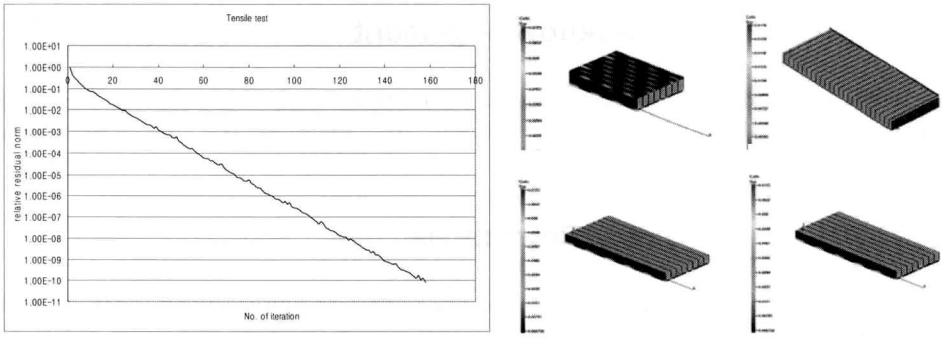

Fig. 5. results of DNS example 1

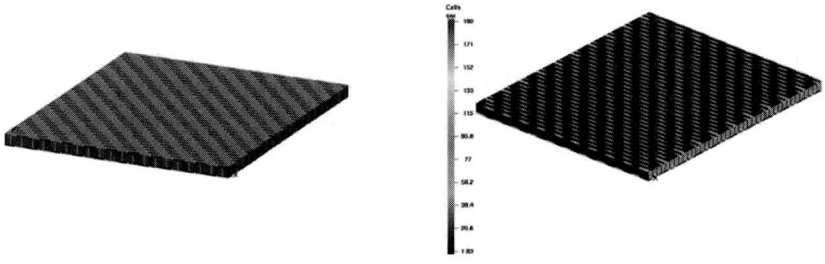

Fig. 6. Finite element model (left) and σxx stress components (right)

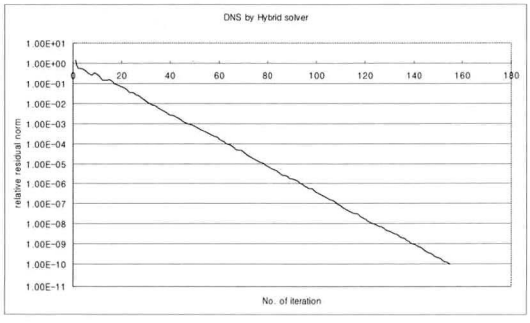

Fig. 7. convegence of DNS example 2

Conclusions

In this paper, efficiency and performance of hybrid solution method is presented. Hybrid solver we developed has good numerical scalability shown by the ratio of subdomain to global mesh size being kept constant. Parallel scalability of hybrid solver is also shown, but it has a little weak since there is some performance drop to treat the coarse problem. Next it should be studied to remove the difficulties of handling coarse problem.

Hybrid solution method is very useful to analyze large scale 3D structures. Hybrid method has better computing and memory efficiency than multifrontal solution method, it has been shown the 3D scalability test. Moreover hybrid method has ability to solve 3D near 100 million DOFs problems by just only 64 CPUs. Direct numerical simulation of woven composite is shown to verify the hybrid solution method, these tests say hybrid solver is efficient and reliable method for large size 3D structures.

Acknowledgement

This work is supported by Korea Science and Engineering Foundation (KOSEF) grant number R0A-2006-000-10178-0, contacted through the Institute of Advanced Aerospace Technology at Seoul National University.

References

1. M. Bhardwaj, D. Day, C. Farhat, M. Lesoinne, K. Pierson and D. Rixen, 2000, "Application of the FETI method to ASCI problems - scalability results on one thousand processors and discussion of highly heterogeneous problems", Int. J. Numer. Meth. Eng., Vol. 47, pp. 513 - 535.

2. ADVENTURE User's Manual, 2001, ver 0.81.

3. H. Okuda, G. Yagawa and H. Nakamura, 2003, "Parallel FE Solid Earth Simulator Tiger and Snake", SC2003.

4. J. H. Kim, C. S. Lee and S. J. Kim, 2005, "High-performance Domain-wise Parallel Direct Solver for Large-scale Structural Analysis", AIAA Journal, Vol. 43, No. 3, pp. 662-670.

5. B. F. Smith, P. E. Bjorstad and W. D. Gropp, 1996, Domain Decomposition - Parallel Multilevel Methods For Elliptic Partial Differential Equations, Cambridge Univ. Press, New York, pp. 1-18.

6. P. Le Tallec, 1994, "Domain decomposition methods in computational mechanics", Computational Mechanics Advances, Vol. 1, No. 2, pp. 121-220.

7. C. Farhat, M. Lesoinne and K. Pierson, 2000, "A scalable dual-primal domain decomposition method", Numer. Linear Algebra Appl., Vol. 7, pp. 687-714.

8. Mandel J and Tezaur R, 1996, "Convergence of a substructuring method with Lagrange multipliers", Numerische Mathematik, Vol. 73, pp. 473 - 487.

9. C. Farhat, 1991, "A Method of Finite Element Tearing and Interconnecting And Its Parallel Solution Algorithm", Int. J. Numer. Meth. Eng., Vol. 32, pp. 1205-1227.