Large Scale Finite Element Analysis with a Balancing Domain Decomposition Method

Shioya, Ryuji
Department of Intelligent Machinery and Systems, Kyushu University

Ogino, Masao
Department of Intelligent Machinery and Systems, Kyushu University

Kanayama, Hiroshi
Department of Intelligent Machinery and Systems, Kyushu University

Tagami, Daisuke
Department of Intelligent Machinery and Systems, Kyushu University

https://hdl.handle.net/2324/6112

出版情報：Key Engineering Materials. 243-244, pp.21-26, 2003-01. Trans Tech Publications
バージョン：

Large Scale Finite Element Analysis with a Balancing Domain Decomposition Method

Ryuji SHIOYA, Masao OGINO, Hiroshi KANAYAMA and Daisuke TAGAMI

Graduate School of Engineering, Kyushu University,
6-10-1 Hakozaki Higashiku Fukuoka 812-8581 JAPAN
E-mail: shioya@mech.kyushu-u.ac.jp

ABSTRACT

This paper describes the parallel finite element analysis of large scale problems based on the Domain Decomposition Method with preconditioner using Balancing Domain Decomposition for a massively parallel processors. In order to solve the issue of memory shortage and computational time, the developed system employs a dynamic load balancing and hierarchical distributed data management technique. The present system is successfully applied to static elastic stress analyses, pressure vessel model of over one million with effective performances.

KEYWORDS

Finite Element Method, Domain Decomposition Method, Balancing Domain Decomposition, Parallel Computing, Large Scale Analysis, Elasticity Problems

INTRODUCTION

With the increase of the size and complexity of numerical simulation problems, such as a finite element method (FEM), more processing power and memory of computer are required. Using single processor computers, we encounter their physical limits. To save computational time and memory, it is well known that the parallel computers, particularly Multiple Instruction Multiple Data (MIMD) type computers including clustered workstation computers seem to be promising. A MIMD type computer has many processors with local memory, and can reduce computational time by distributing tasks among processors. However, we need special algorithms for parallel computing to solve problems with high performance using this kind of computer.

The iterative Domain Decomposition Method (DDM) is one of the most effective parallel methods for large scale problems due to its excellent parallelism and suitability for various kinds of parallel computers such as massively parallel processors and workstation/PC clusters [Yagawa and Shioya(1994), Shioya and Yagawa(1998)]. As the iterative DDM satisfies continuity among subdomains through iterative calculations such as the Conjugate Gradient (CG) method, it is indispensable to reduce the number of iterations with a preconditioning technique especially for large scale problems.

The Neumann-Neumann algorithm (N-N) is known to be an efficient domain decomposition preconditioner with unstructured subdomains for an iterative solution of finite element discretization of difficult problems with strongly discontinuous coefficients [Roeck and Tallec(1991)]. However, its convergence deteriorates with the increasing number of subdomains due to the lack of a coarse problem to propagate the error globally. The Balancing Domain Decomposition method (BDD) based on N-N introduced by Mandel [Mandel(1993)] shows that the equilibrium conditions for the singular problems on subdomains lead to the simple and natural construction of a coarse problem. The construction is purely algebraic. In this study, an implementation of BDD to static elastic stress analysis is presented and some numerical experiments are performed.

BALANCING DOMAIN DECOMPOSITION

Interface Problem

Consider a system of linear algebraic equations,

$$Ku = f \tag{1}$$

arising from a finite element discretization of a linear, elliptic, self adjoint boundary value problem in domain Ω.

The domain Ω is split into non-overlapping subdomains $\Omega^{(1)}$,...,$\Omega^{(k)}$ and union of all subdomains boundaries is $\Gamma = \cup_{i=1}^{k} \partial\Omega^{(i)}$. Let $V^{(i)}$ and V be the spaces of degrees of freedom on $\partial\Omega^{(i)}$ and Γ. Let $u^{(i)}$ be the vector of degrees of freedom corresponding to all elements in subdomain $\Omega^{(i)}$ and let $N^{(i)}$ denotes the $0-1$ matrix that maps the degrees of freedom $u^{(i)}$ into global degrees of freedom u.

Each $u^{(i)}$ is split into degrees of freedom $u_B^{(i)}$ that correspond to the interface of the $\Omega^{(i)}$ with other subdomains, and the remaining degrees of freedom $u_I^{(i)}$. The subdomain stiffness matrices and $N^{(i)}$ are then split accordingly and the system (1) is

$$\begin{bmatrix} K_{II}^{(i)} & K_{IB}^{(i)} \\ {K_{IB}^{(i)}}^T & K_{BB}^{(i)} \end{bmatrix} \begin{bmatrix} u_I^{(i)} \\ u_B^{(i)} \end{bmatrix} = \begin{bmatrix} f_I^{(i)} \\ f_B^{(i)} \end{bmatrix} \tag{2}$$

After eliminating $u^{(i)}$, the system (1) becomes

$$Su_b = g \tag{3}$$

where S is the Schur complement that is the assembly of the local ones:

$$\begin{aligned} S &= \sum_{i=1}^{k} N_B^{(i)} S^{(i)} {N_B^{(i)}}^T && (4) \\ S^{(i)} &= K_{BB}^{(i)} - {K_{IB}^{(i)}}^T {K_{II}^{(i)}}^{-1} K_{IB}^{(i)} && (5) \end{aligned}$$

The local Schur complements $S^{(i)}$ are positive semidefinite.

A large number of domain decomposition (or substructuring) methods consist of solving the reduced system (3) iteratively. Since S is symmetric positive definite, the preconditioned CG method is the standard choice for iterative methods. This method requires at each step the solution of an auxiliary problem,

$$Mz = r \tag{6}$$

with a symmetric positive definite matrix M, called a preconditioner.

Neumann-Neumann Preconditioner

The method uses a collection of matrices $D^{(i)}$ that form a decomposition of unity,

$$\sum_{i=1}^{k} N_B^{(i)} D^{(i)} {N_B^{(i)}}^T = I \tag{7}$$

The simplest choice for $D^{(i)}$ is the diagonal matrix with diagonal elements equal to the reciprocal of the number of subdomains with which the degrees of freedom is associated. The N-N preconditioned operator M^{-1} is described by

$$M^{-1} = \sum_{i=1}^{k} N_B^{(i)} D^{(i)} S^{(i)\dagger} {D^{(i)}}^T {N_B^{(i)}}^T \tag{8}$$

where $S^{(i)\dagger}$ is the Moore-Penrose pseudoinverse of $S^{(i)}$.

Another drawback of N-N is the lack of a mechanism to exchange information between all subdomains in the preconditioning step and thus to prevent the growth of the condition number with the number of subdomains. BDD settled this matter by solving a coarse problem.

Balancing Domain Decomposition Preconditioner

Let $n^{(i)}$ be the dimension of $V^{(i)}$, let $m^{(i)}$ be the number with $0 \leq m^{(i)} \leq n^{(i)}$, let $Z^{(i)}$ be the $n^{(i)} \times m^{(i)}$ matrices of full column rank such that

$$Range Z^{(i)} \subset Null S^{(i)}, i = 1, ..., k \tag{9}$$

and let W be the coarse space defined by

$$W = \{v \in V / v = \sum_{i=1}^{k} N_B^{(i)} D^{(i)} u^{(i)}, u^{(i)} \in Range Z^{(i)}\} \tag{10}$$

Let P be the S-orthogonal projection onto W, then the BDD preconditioned operator M^{-1} is described by

$$M^{-1} = P + \sum_{i=1}^{k} (I - P) N_B^{(i)} D^{(i)} S^{(i)\dagger} {D^{(i)}}^T {N_B^{(i)}}^T (I - P) \tag{11}$$

Using BDD, we should determine an efficient $Z^{(i)}$ satisfying (9). For an elastic stress problem, $Null S^{(i)}$ can be considered to correspond to the degrees of freedom of rigid displacement. At the point $X(x_1, x_2, x_3)$ on the interface of the subdomain $\Omega^{(i)}$, let $Z_X^{(i)}$ be defined as:

$$Z_X^{(i)} \equiv \begin{pmatrix} 1 & 0 & 0 & 0 & x_3 & -x_2 \\ 0 & 1 & 0 & -x_3 & 0 & x_1 \\ 0 & 0 & 1 & x_2 & -x_1 & 0 \end{pmatrix} \tag{12}$$

and Z_i be defined by assembling $Z_X^{(i)}$,

$$Z^{(i)} = \sum_P B_X^{(i)} Z_X^{(i)} \tag{13}$$

where $B_X^{(i)}$ is the 0-1 matrix that maps the degrees of freedom X into global degrees of freedom of the interface of the subdomain $\Omega^{(i)}$.

Algorithm of Balancing Domain Decomposition is summarized as follows:

$$\begin{cases} step1: & Z^{(i)^T} D^{(i)^T} N_B^{(i)^T} (r - S\sum_{j=1}^{k} N_B^{(j)} D^{(j)} Z^{(j)} \lambda^{(j)}) = 0, i = 1, ..., k \\ step2: & s = r - S\sum_{j=1}^{k} N_B^{(j)} D^{(j)} Z^{(j)} \lambda^{(j)}, s^{(i)} = D^{(i)^T} N_B^{(i)^T} s, i = 1, ..., k \\ step3: & u^{(i)} = S^{(i)\dagger} s^{(i)}, i = 1, ..., k \\ step4: & Z^{(i)^T} D^{(i)^T} N_B^{(i)^T} (r - S\sum_{j=1}^{k} N_B^{(j)} D^{(j)} (u^{(j)} + Z^{(j)} \mu^{(j)})) = 0, i = 1, ..., k \\ step5: & z = \sum_{i=1}^{k} N_B^{(i)} D^{(i)} (u^{(i)} + Z^{(i)} \mu^{(i)}) \end{cases}$$

NUMERICAL EXPERIMENTS

BDD performances

The present system is applied to the pipe model shown in Figure 1 to estimate the effect of number of subdomains to the convergence. This model is expressed by 6,333 10-noded tetrahedral elements. The total degrees of freedom is 33,585 and for test analysis, it is divided into four sizes of subdomains, i.e. 20, 50, 80 and 160 subdomains. DDM without any preconditioner (DDM), DDM with diagonal scale preconditioner (DSCALE) and BDD are performed for these models.

The numbers of iterations until convergence are shown in Table1. Though that of DSCALE decrease in all cases compared with DDM, they increase with the number of subdomains. On the other hand, for the case of BDD, it is shown not only that the number of iterations drastically decreases compared with DDM but also that the increasing number of iterations for the number of subdomains is very little.

Large Scale Analysis

As a larger model, the present system is applied to the finite element analysis of an shaft bearing model shown in Figure 2. This model is expressed by 10-noded tetrahedral elements, divided into subdomains and divided again into parts. The sizes of this model are listed in Table 2. This problem is solved by Alpha Cluster (DEC AlphaAXP 533MHz) consisting of 18 processors, 1 Grand, 3 Parents and 14 Children were assigned for HDDM.

Table 1: NUMBER OF ITERATIONS FOR PIPE MODEL

Num. of Domains	DDM	DSCALE	BDD
20	1,423	822	36
50	1,823	1,097	40
80	2,170	1,090	38
160	2,580	1,143	45

Table 2: MESH SIZES FOR SHAFT BEARING MODEL

Nodes	Elements	Subdomains	Parts	Total DOFs	Interface DOFs
71,970	117,977	600	3	295,708	101,856

Table 3: CALCULATION PERFORMANCES WITH HDDM AND BDD

	Num. of Iter	CPU Time	Mem. for GP	Mem. for P	Mem. for C	CPU Usage
DDM	1,072	20.8[min]	0.001 [MB]	5.6 [MB]	50.1[MB]	68.5%
BDD	113	11.0[min]	0.03 [MB]	91.5 [MB]	46.2[MB]	42.5%

For this model, DDM with diagonal preconditioner and BDD are performed. The number of iterations, calculation times and required amount of memory are shown in Table 3. As shown in the table, although BDD requires more memory than DDM, reducing the number of iterations and speeding up the calculation time are achieved. The force imbalance measure at inter subdomain measure (residual value) against the number of CG iterations are shown in Figure 1. It is shown here that in case of BDD, the residual values decrease almost monotonically with the increase of the number of iterations.

CONCLUSION

The finite element system based on the DDM with preconditioner using BDD was developed in the current study. This system can be applied to structural analyses and effective performances were obtained. To apply for larger models like over ten million degrees of freedom problem, parallelizing of the system and reducing memory usage of BDD are required.

REFERENCES

J. Mandel. (1993). Balancing domain decomposition. Communications on Numerical Methods in Engineering **9**, 233-241.

Y. H. De Roeck and P. Le Tallec. (1991). Analysis and test of a local domain decomposition preconditioner. Fourth International Symposium on Domain Decomposition Methods for Partial Differential Equations, SIAM, Philadelphia, PA

R. Shioya and G. Yagawa. (1998). Parallel finite elements of ten-million DOFs based on domain decomposition method. WCCM IV, Computational Mechanics -New Trends and Applications- **VII:5-11**, 1-12.

G. Yagawa and R. Shioya. (1994). Parallel finite elements on a massively parallel computer with domain decomposition. Computing systems in Engineering **4:4-6**, 495-503.

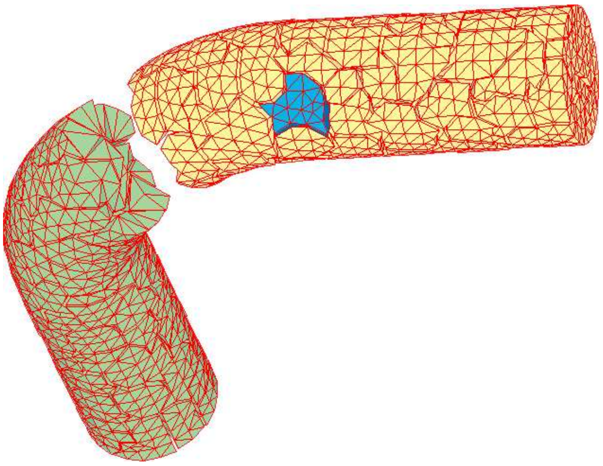

Figure 1: Mesh of Pipe Model

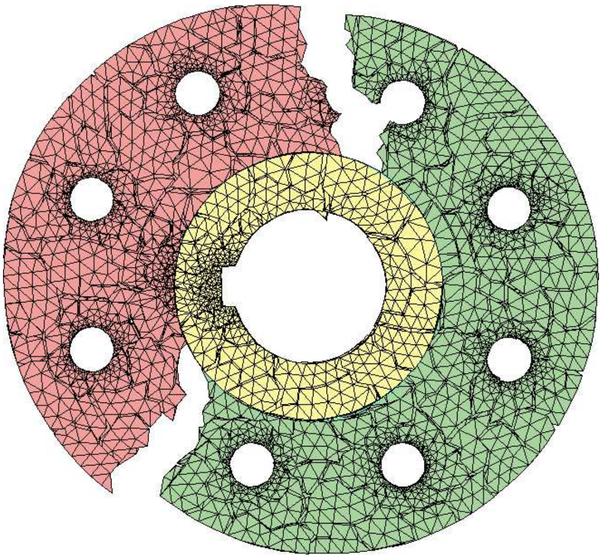

Figure 2: Shaft Bearing Model

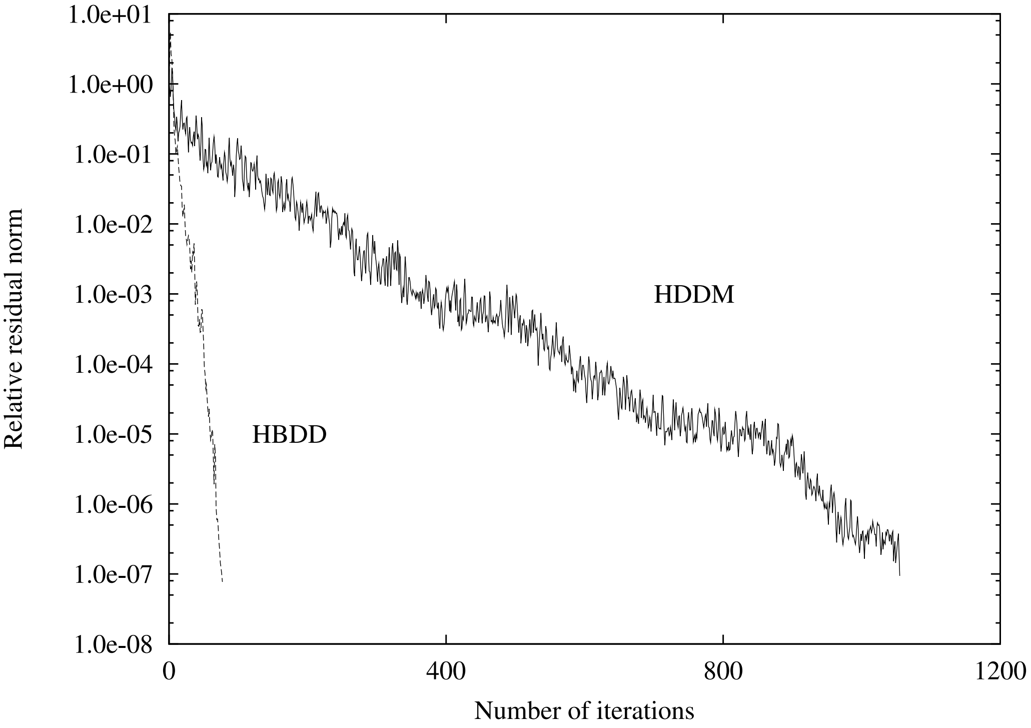

Figure 3: Profile of Residual Norm