

STATIC ELASTIC-PLASTIC STRESS ANALYSES OF TEN-MILLION DOFS PROBLEMS USING HDDM

Tomoshi MIYAMURA ^{*1} and Hirohisa NOGUCHI ^{*2}

^{*1} Department of Environmental Studies, Graduate School of Frontier Sciences
University of Tokyo

e-mail: miyamura@garlic.q.t.u-tokyo.ac.jp, web page: <http://garlic.q.t.u-tokyo.ac.jp/~miyamura>

^{*2} Department of System Design Engineering
Keio University

e-mail: noguchi@sd.keio.ac.jp

Key words: Parallel Computation, Elastic-Plastic Analysis, Domain Decomposition Method, Large Scale Solid and Structure.

1 INTRODUCTION

Various commercial codes for nonlinear finite element analysis of solids and structures have been developed, and widely used for practical problems such as design of artifacts and prediction of natural phenomena. However, much larger scale analysis models are required to precisely simulate, for example, behaviors of a whole part of aircraft or nuclear reactor, mechanism of earthquake etc. Recently, massively parallel processors (MPP) have been the main stream of high performance computers for scientific and engineering computations. Virtual parallel processors such as workstation (WS) cluster and personal computer (PC) cluster have been also widely used.

The hierarchical domain decomposition method (HDDM) that was proposed by Yagawa and Shioya [Yagawa and Shioya, 1993] is one of solvers for large scale algebraic equations in the finite element method, which is suitable for various kinds of parallel computers. Some basic studies for applying the domain decomposition method to the elastic plastic analysis were carried out by Yagawa et al. [Soneda et al., 1991][Yagawa and Uchida, 1994][Shioya et al., 1997] and the present authors [Miyamura et al., 1999]. The purpose of the present study is to extend the HDDM to large scale nonlinear finite element analyses of solids. Static elastic-plastic problem is considered here. As illustrative examples, a complex shape model with over 1.3 millions degrees of freedom (M DOFs), and a simple cube model with 10 M DOFs are analyzed by using 256 PEs (processing elements) and 1024 PEs, respectively, of HITACH SR2201.

A survey of parallel nonlinear dynamic analyses in 1980s and early 1990s can be found in Ref. [6] by Fahmy and Namini. In 1990s, however, distributed-memory parallel computers have been rapidly improved, and efficient methods of implementation of parallel implicit solvers have been developed. Farhat and Roux [Farhat and Roux, 1994] proposed the Finite

Element Tearing and Interconnecting (FETI) method, which is a kind of substructure, based iterative method or domain decomposition method. Recently, an analysis of a shell structure with 0.88 M DOFs was reported in Ref. [Farhat et al., 1998] by using 64 processing elements (PEs) of IBM SP2. Kacou and Parsons [Kacou and Parsons, 1993][Parsons, 1997] proposed a parallel multigrid method for the elastic-plastic analysis based on the Newton-Raphson method. They proposed convergence criteria of the multigrid method and the Newton-Raphson method. An example of 132 K DOFs problem was solved by using coarse-grained shared-memory parallel computers. Fish et al. [Fish et al., 1995] proposed an attractive implicit solver for large scale nonlinear problems in which the BFGS method and a multigrid method were combined, although it is not implemented in parallel environments. In this method, the two iterative procedures are unified, and only one iterative loop is required. An elastic-plastic problem with 80 K DOFs was solved by using a single Sun SS10 workstation. Nikishkov et al. [Nikishkov et al., 1998] developed a code for the sheet metal forming analysis by a parallelized substructure method. A sheet metal was modeled by shell elements with 20 K DOFs and analyzed by using 8 PEs of IBM SP2. Nakajima et al. [Nakajima and Okuda] developed an implicit parallel solver based on a domain decomposition method with an ILU(0) preconditioner, which was proposed by Venkatakrishnan [Venkatakrishnan, 1994]. Iizuka et al. [Iizuka, Okuda and Yagawa, 1999] analyzed an elastic-plastic problem of a complex structure with 1.3 M DOFs (the same model that is analyzed in Chapter 6 of the present paper) by using this solver. Garatani et al. [Garatani, 1999] analyzed a simple elastic problem with 100 M DOFs also by using the solver.

In the field of fluid dynamics, some studies concerning with parallel implicit solvers can be found. Tezduyar et al. [Tezduyar et al., 1994] illustrated examples of large scale flow problems including a problem with over 5 M equations. These problems were mainly solved by a Connection Machine CM-5 system, which is a distributed-memory massively parallel computer. A parallel implementation of the implicit finite element formulation used in these examples was illustrated by Kennedy et al. [Kennedy et al., 1994]. Keyes et al. [Keyes and Venkatakrishnan, 1996] proposed the Newton-Krylov-Schwarz method, which is a nonlinear implicit solver for fluid dynamics. This method consists of outer Newton iteration for solving nonlinear equations and inner Krylov iteration for solving the linear equations for the incremental method. Overlapping Schwarz type domain decomposition preconditioner is combined with the Newton-Krylov iterations. An aerodynamic problem with 11 M DOFs was solved by this method using 128 or 512 PEs of SGI T3E-900 [Keyes, 1998].

The dynamic analysis based on the explicit time integration procedure is another choice for solving large scale nonlinear problems, because it is suitable for parallelization. Johan et al. [Johan et al., 1994] showed an example of the elastic-viscoplastic analysis with 14 M DOFs by using CM-5. Note that CM5 is a less powerful computer comparing with the currently available MPPs such as SR2201 and T3E. Belytschko et al. [Belytschko et al., 1994] analyzed the shear band on 2D model with 64 K elements by using a parallel explicit scheme.

The contents of the rest of the present paper are as follows. In Chapter 2, the hierarchical domain decomposition method (HDDM) is briefly illustrated. In Chapter 3, a technique to improve computation speed of the original HDDM is presented, because large scale algebraic equations have to be solved many times in nonlinear analyses. In Chapter 4, some key techniques to combine the HDDM with the elastic-plastic algorithm are illustrated. In Chapter

5, the improved HDDM described in Chapter 3 is evaluated. Results of the large scale elastic-plastic analyses are presented in Chapter 6, and concluding remarks appear in Chapter 7.

2 HIERARCHICAL DOMAIN DECOMPOSITION METHOD

2.1 Domain decomposition method

In the domain decomposition methods (DDM), an analysis model is subdivided into subdomains. Although there are several kinds of DDM, a kind of substructure method is adopted in this study. This method is simply called the DDM here. In the followings, the boundaries between subdomains are called the internal boundaries. The DOFs concerning internal nodes of each subdomain are called the internal DOFs, while those concerning the nodes on the internal boundaries are called the interface DOFs. In the DDM adopted in this paper, the internal DOFs are statically condensed by using a direct solver such as the skyline method. The interface DOFs are then solved by using an iterative solver such as the conjugate gradient (CG) method. This method is sometimes called the ‘substructure-based conjugate gradient algorithm [Farhat and Roux, 1994], because it uses the same fundamental equations as the substructure method. Brief outline of the formulation is described in the following.

First the stiffness matrix for each subdomain is partitioned as follows:

$$\mathbf{K}^k = \begin{bmatrix} \mathbf{K}_{II}^k & \mathbf{K}_{IB}^k \\ \mathbf{K}_{BI}^k & \mathbf{K}_{BB}^k \end{bmatrix}, \quad (1)$$

where k denotes the subdomain number, I shows the part concerning with the internal DOFs, B shows the part concerning with the interface DOFs. Equilibrium equations for the internal DOFs are described as

$$\mathbf{K}_{II}^k \mathbf{u}_I^k + \mathbf{K}_{IB}^k \mathbf{u}_B^k = \mathbf{p}_I^k, \quad (2)$$

and those for the interface DOFs are represented as follows:

$$\sum_{k=1}^{NDOM} \mathbf{K}_{BI}^k \mathbf{u}_I^k + \sum_{k=1}^{NDOM} \mathbf{K}_{BB}^k \mathbf{u}_B^k = \sum_{k=1}^{NDOM} \mathbf{p}_B^k, \quad (3)$$

where \mathbf{u} denotes the nodal displacement vector, \mathbf{p} denotes the equivalent nodal external force vector, and $NDOM$ shows the number of subdomains. Equation (2) can be solved independently for \mathbf{u}_I^k in each subdomain as follows:

$$\mathbf{u}_I^k = (\mathbf{K}_{II}^k)^{-1} (\mathbf{p}_I^k - \mathbf{K}_{IB}^k \mathbf{u}_B^k). \quad (4)$$

\mathbf{u}_I^k is eliminated (i. e., statically condensed) by substituting from Eq. (4) into Eq. (3) as

$$\begin{aligned} & \sum_{k=1}^{NDOM} \left\{ -\mathbf{K}_{BI}^k (\mathbf{K}_{II}^k)^{-1} \mathbf{K}_{IB}^k + \mathbf{K}_{BB}^k \right\} \mathbf{u}_B^k \\ & = \sum_{k=1}^{NDOM} \left\{ -\mathbf{K}_{BI}^k (\mathbf{K}_{II}^k)^{-1} \mathbf{p}_I^k + \mathbf{p}_B^k \right\} \end{aligned} \quad (5)$$

Equation (5) is a system of linear algebraic equations concerning with \mathbf{u}_B^k . In analyses of solid, the coefficient matrix of Eq. (5) is usually symmetric. Hence, the CG method can be adopted to solve Eq. (5) for \mathbf{u}_B^k . Iteration of the CG is continued until the following residual vector, \mathbf{g} becomes sufficiently small:

$$\mathbf{g} = \sum_{k=1}^{NDOM} \left\{ -\mathbf{K}_{BI}^k (\mathbf{K}_{II}^k)^{-1} \mathbf{K}_{IB}^k + \mathbf{K}_{BB}^k \right\} \mathbf{u}_B^k - \sum_{k=1}^{NDOM} \left\{ -\mathbf{K}_{BI}^k (\mathbf{K}_{II}^k)^{-1} \mathbf{p}_I^k + \mathbf{p}_B^k \right\} \quad (6)$$

2.2 Hierarchical domain decomposition method

Yagawa and Shioya proposed a hierarchical technique to implement the DDM on MPP [Yagawa and Shioya 1993]. This technique is called the hierarchical domain decomposition method (HDDM). In this method, a group of processing elements (PEs) is subdivided into three groups: one Grand Parent PE (Grand), several Parent PEs (Parent/Parents), and many Child PEs (Child/Children).

Figure 1 shows the schematic data flow among PEs in the HDDM. An analysis model is subdivided into several “parts” whose number is the same as the number of Parents. Each Parent stores in its memory the data for a part including nodal coordinates, connectivities of elements and material properties. In nonlinear analyses, stresses and displacements also have to be stored since they are used in calculating the tangent stiffness matrix and stresses.

The static condensation of the subdomain (i. e., the elimination of \mathbf{u}_I^k using Eq. (4)) is done by Children. The computation in a Child is almost the same as the ordinary finite element analysis. Each Child receives the analysis data from a Parent, and sends back the results to the Parent. Children are dynamically allocated to Parents that possess the subdomains whose static condensation is not finished.

Grand checks the status of each Parent and arranges Children. It also manages the whole computation. After the static condensation processes of all the subdomains in all the parts are finished, Grand and Parents update the solution vector based on the CG method. In this procedure, some communications between Grand and Parents are required. The series of analyses are iterated until the residual of the CG method becomes sufficiently small.

The features of the HDDM are summarized as follows:

- (1) Analysis data are stored in several Parents. The number of Children is variable though it has to be smaller than the number of subdomains. Therefore, robust operations can be achieved, i. e., a user can change the number of PEs, depending on analysis cases, without changing the domain decomposition of an analysis model.
- (2) Because Children do not store analysis data, the dynamic workload balancing can be implemented, which means that good parallel efficiency can be attained. This implementation is also suitable for heterogeneous parallel environments such as PC/WC clusters.
- (3) Any large scale analysis data can be handled by changing the number of Parents.

When the HDDM is used in static nonlinear analyses of solid, incremental loop and the Newton-Raphson iteration loop are necessary. These loops are placed outside the CG loop. Figure 2 illustrates the analysis flow of the HDDM-based nonlinear analysis.

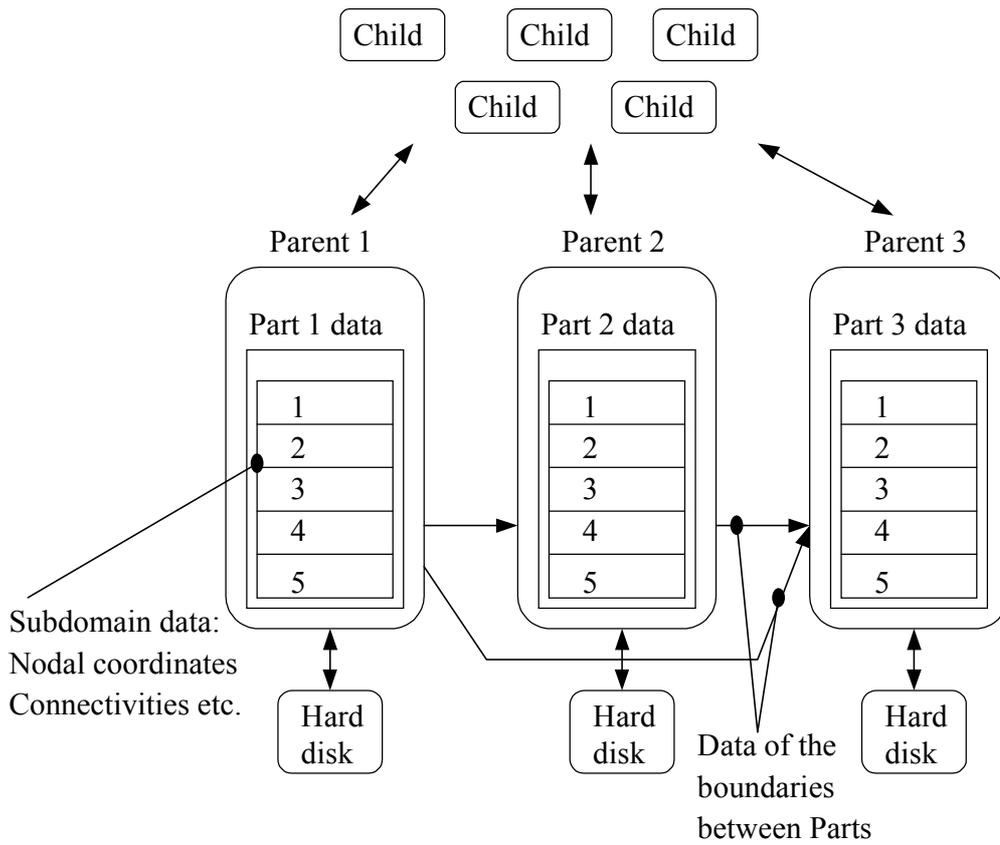


Fig. 1 Schematic data flow among PEs in the HDDM

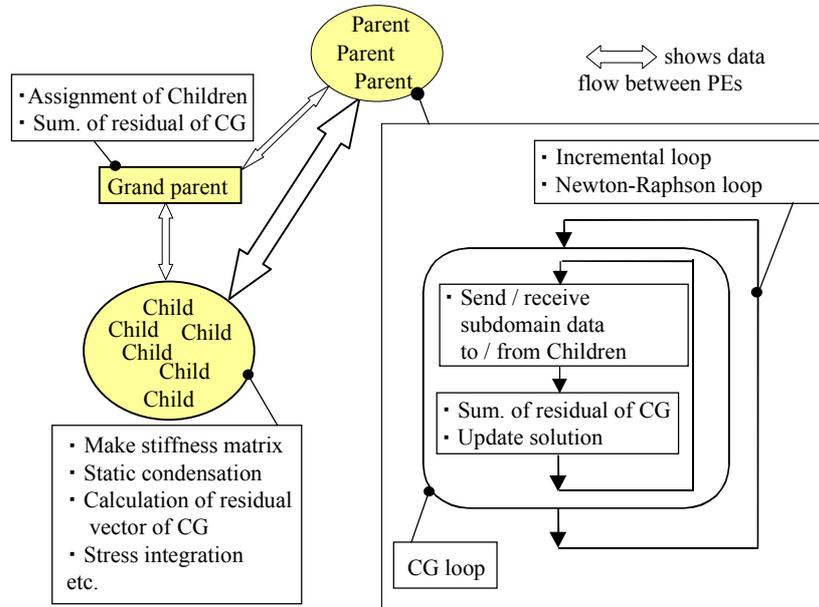


Fig. 2 Nonlinear finite element analysis of solids using the HDDM

3. IMPROVEMENT OF CALCULATION SPEED OF THE HDDM

3.1 Calculation speed versus memory use

A characteristic of the original HDDM was that large scale analysis models can be solved without consuming much memory, i. e., \mathbf{K}^k in Eq. (1) and $(\mathbf{K}_{II}^k)^{-1}$ in Eq. (4) are not stored and re-calculated in each CG step (Fig. 3 (a)). However, in recent MPPs or PC/WS clusters, each PE has a large amount of memory. For example, each PE of HITACHI SR2201 in the University of Tokyo has 224 M Bytes memory. Therefore, in the present method, \mathbf{K}^k and $(\mathbf{K}_{II}^k)^{-1}$ are stored in memory. Note that $(\mathbf{K}_{II}^k)^{-1}$ is actually the LU or LDL^T decomposition of \mathbf{K}_{II}^k in practical codes.

When this strategy is employed, (1) a feature of dynamic workload balancing is restricted, as described in the following section, however, (2) analysis data still managed by some Parents and a feature of robust data operation is still preserved. Comparisons of these two strategy, i.e., memory-oriented strategy and the computation-speed-oriented strategy can be found in Ref. [Yagawa and Soneda, 1991].

3.2 Hybrid workload balancing

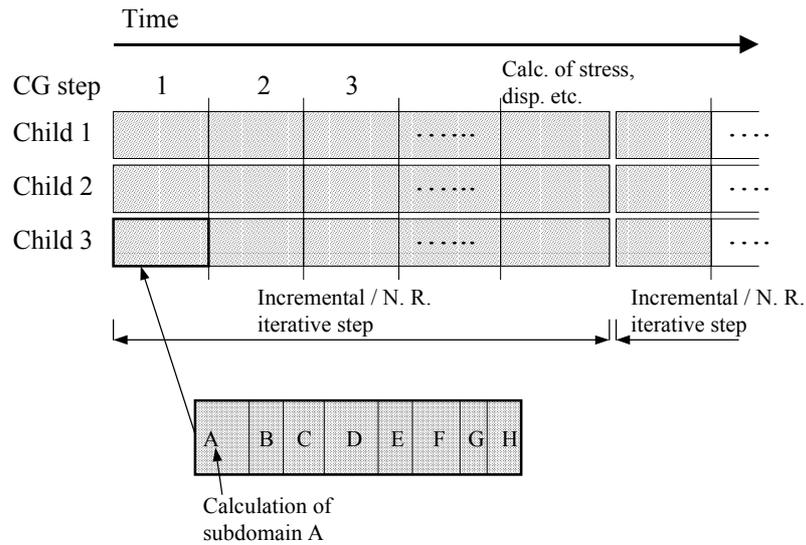
As described in Chapter 2, almost all analysis data are stored in Parents. Few amount of memory in Children is used, which means that \mathbf{K}^k and $(\mathbf{K}_{II}^k)^{-1}$ can be stored in Children. However, when sets of subdomain data in a Parent is assigned to Children dynamically, a Child that stores data of a subdomain does not always receive the same subdomain's data in the next CG step. To avoid this problem, a hybrid workload balancing technique, i. e., a combination of the dynamic workload balancing and static workload balancing, is adopted. In this technique, the dynamic workload balancing used in the original HDDM is adopted in the first CG step. Each Child calculates \mathbf{K}^k s and the LU decomposition of $(\mathbf{K}_{II}^k)^{-1}$ s concerning some subdomains and stores them in its memory. Subdomain number and task ID of the Parent that possesses the data for this subdomain are also stored in each Child. After the first CG step, Children are statically allocated to the related Parents, i. e., the workload balancing is static. Computation flow in Children is shown in Fig. 3 (b).

In the present method, if the workload of PEs is changed due to other processes of other users during the CG iteration, the workload balancing is affected by this fluctuation. Therefore, this technique is sometimes not suitable for the heterogeneous parallel environment. However, in nonlinear analyses, scales of analysis models are usually smaller than in case of linear analyses, and the HDDM solver is called many times because the incremental/iterative method is used. In this case, the dynamic workload balancing can be performed in the first step of CG in each incremental/iterative step. Thus, this technique can efficiently be used even on the unstable heterogeneous parallel environment in case of nonlinear analyses.

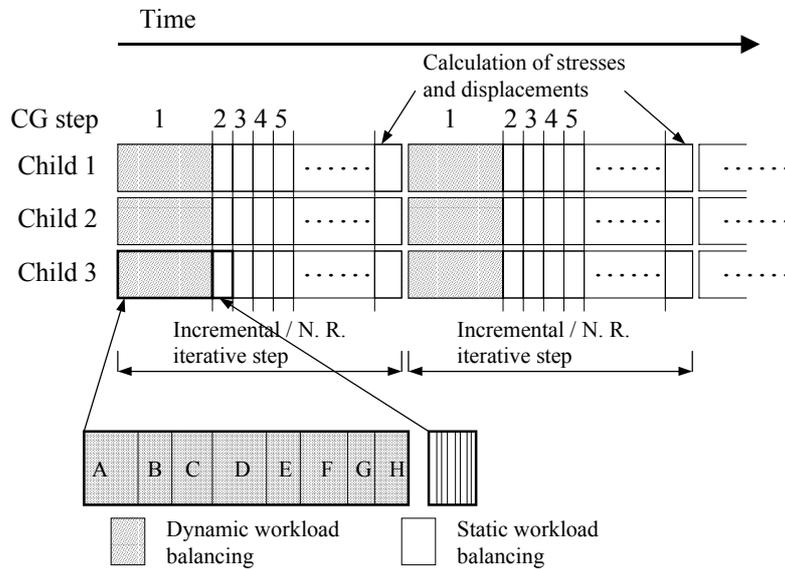
When a large scale elastic analysis is performed, restarts during CG iteration are often necessary. Because data of \mathbf{K}^k and $(\mathbf{K}_{II}^k)^{-1}$ are large, it is impossible to output them on hard disk as restart data. Instead of that, only vectors used in the process of the CG is output, and

\mathbf{K}^k and $(\mathbf{K}_{II}^k)^{-1}$ are rebuilt in the first CG step. The dynamic workload balancing can be attained in this procedure.

The original HDDM is sometimes effective especially when PEs in a parallel computer do not have enough memory, or, as mentioned above, when a large scale linear problem is solved on unstable parallel computers. Because the present technique can easily be implemented by slightly modifying the original HDDM code, a user can choose between these two methods considering the scale of an analysis model and the parallel environment he or she can utilize.



(a) Original HDDM



(b) Present HDDM

Fig. 3 Computation flow in Children; Comparison between original HDDM and present HDDM

4. ALGORITHM OF ELASTIC-PLASTIC ANALYSIS

4.1 Stress integration and consistent tangent

In a static elastic-plastic analysis, an incremental method combined with the Newton-Raphson method is usually employed. The modified Newton-Raphson method is sometimes efficient when a direct solver is used, because once the LU decomposition of the stiffness matrix is obtained, solutions for subsequent residual vectors can easily be calculated. However, this feature is not true when an iterative solver is used.

Here, the elastic predictor-radial corrector (radial return) method is employed as a stress integration scheme [Simo and Taylor, 1985] [Hisada and Noguchi, 1995] [Bathe, 1996]. By this method, accurate stresses can be calculated even when large increment is applied. In the Newton-Raphson iteration, the tangent stiffness matrix that is consistent with the stress integration scheme is employed [Simo and Taylor, 1985]. Hence, the quadratic convergence of the Newton-Raphson method will be attained.

In the following, brief outline of the formulation is described to illustrate how to use the HDDM in nonlinear analyses. Note that quantities at time t is known, and those at time $t + \Delta t$ is unknown and will be calculated.

The stress tensor at time $t + \Delta t$ is obtained by using the known stress tensor ${}^t\boldsymbol{\sigma}$, the yield stress ${}^t\sigma_Y$ at time t and the incremental displacement vector ${}^{t+\Delta t}{}_t\mathbf{u}$ from time t to $t + \Delta t$ as follows:

$${}^{t+\Delta t}\boldsymbol{\sigma} = {}^{t+\Delta t}\boldsymbol{\sigma}({}^t\boldsymbol{\sigma}, {}^t\sigma_Y, {}^{t+\Delta t}{}_t\mathbf{u}). \quad (7)$$

Note that the incremental strain tensor is a function of the incremental displacement when the displacement-based formulation is employed. Also, note that the stress integration should be performed from time t to $t + \Delta t$ during the Newton-Raphson iteration.

To calculate the consistent tangent stiffness, the constitutive equations that are consistent with the stress integration scheme should be used as

$$d^{t+\Delta t}\boldsymbol{\sigma} = \mathbf{C}^{\text{EP}*} : d^{t+\Delta t}\mathbf{e}, \quad (8)$$

where $d^{t+\Delta t}\boldsymbol{\sigma}$ is the increment of the stress tensor at time $t + \Delta t$, $d^{t+\Delta t}\mathbf{e}$ is that of the strain tensor, and $\mathbf{C}^{\text{EP}*}$ is the fourth order tensor concerning with the consistent constitutive equation. $\mathbf{C}^{\text{EP}*}$ is a function of the stress ${}^t\boldsymbol{\sigma}$ at time t and ${}^{t+\Delta t}\boldsymbol{\sigma}$ calculated in the previous Newton-Raphson iterative step as follows:

$$\mathbf{C}^{\text{EP}*} = \mathbf{C}^{\text{EP}*}({}^t\boldsymbol{\sigma}, {}^{t+\Delta t}\boldsymbol{\sigma}, {}^t\sigma_Y). \quad (9)$$

When unloading occurs during the Newton-Raphson iteration, elastic tensor is used instead of $\mathbf{C}^{\text{EP}*}$. As shown in Fig. 1, in the static elastic-plastic (or generally nonlinear) analysis combined with the HDDM, the stress in Eq. (7), the constitutive tensor in Eq. (9), and the tangent stiffness matrix are computed by Children in parallel. However, the cost of these calculations is much smaller than that of the CG iteration.

4.2 Use of HDDM in incremental/Newton-Raphson iterative steps

There are two residuals: i. e., the residual vector of the CG method and the out-of-balance force vector of the Newton-Raphson method. In this paper, the former is called the ‘residual of CG’ and the latter is called the ‘out-of-balance force of NR’.

In the incremental step, the following linearized equilibrium equation is solved:

$${}^t \mathbf{K} {}^{t+\Delta t} \mathbf{u}^0 = {}^{t+\Delta t} \mathbf{R} - {}^t \mathbf{Q}, \quad (10)$$

where ${}^t \mathbf{K}$ is the tangential stiffness matrix at time t , ${}^{t+\Delta t} \mathbf{R}$ is the equivalent nodal external force vector at time $t + \Delta t$, ${}^t \mathbf{Q}$ is the equivalent nodal internal force vector at time t , and ${}^{t+\Delta t} \mathbf{u}^0$ is the predictor of the incremental nodal displacement vector from time t to $t + \Delta t$. On the other hand, in the Newton-Raphson iterative steps, the following equation is solved:

$${}^{t+\Delta t} \mathbf{K}^{i-1} \Delta \mathbf{u}^i = {}^{t+\Delta t} \mathbf{R} - {}^{t+\Delta t} \mathbf{Q}^{i-1}, \quad (11)$$

where i denotes the number of the Newton-Raphson iteration, and $\Delta \mathbf{u}^i$ is the corrector of the incremental nodal displacement. ${}^{t+\Delta t} \mathbf{K}^{i-1}$ and ${}^{t+\Delta t} \mathbf{Q}^{i-1}$ are calculated using the stresses obtained in the previous iterative step. By solving Eqs (10) and (11) for ${}^{t+\Delta t} \mathbf{u}^0$ and $\Delta \mathbf{u}^i$, respectively, the nodal displacement vector from time t to $t + \Delta t$ are calculated as follows:

$${}^{t+\Delta t} \mathbf{u} = {}^{t+\Delta t} \mathbf{u}^0 + \sum_i \Delta \mathbf{u}^i. \quad (12)$$

Only the right-hand side of Eq. (10) or (11) is different from that of the linear equilibrium equation. Hence, when the DDM is employed, the load terms in Eqs (2) and (3) are replaced as follows:

$$\begin{Bmatrix} \mathbf{P}_I^k \\ \mathbf{P}_B^k \end{Bmatrix} = \begin{Bmatrix} {}^{t+\Delta t} \mathbf{R}_I^k - {}^t \mathbf{Q}_I^k \\ {}^{t+\Delta t} \mathbf{R}_B^k - {}^t \mathbf{Q}_B^k \end{Bmatrix} \quad (\text{incremental step}) \quad (13)$$

$$\begin{Bmatrix} \mathbf{P}_I^k \\ \mathbf{P}_B^k \end{Bmatrix} = \begin{Bmatrix} {}^{t+\Delta t} \mathbf{R}_I^k - {}^{t+\Delta t} \mathbf{Q}_I^k \\ {}^{t+\Delta t} \mathbf{R}_B^k - {}^{t+\Delta t} \mathbf{Q}_B^k \end{Bmatrix} \quad (\text{iterative step}) \quad (14)$$

The right-hand side of Eq (13) is the incremental nodal load vector, and that of Eq. (14) is the nodal out-of-balance force vector. The out-of-balance force concerning with the internal DOFs of each subdomain can be calculated independently. On the other hand, it can be understood from Eq. (3) that the nodal out-of-balance force vector concerning with the interface DOFs is calculated by assembling Eq. (14) for all subdomains.

By substituting from Eq. (14) and $\mathbf{u}_B^k = \mathbf{0}$ into Eq. (6), the following residual of CG is calculated:

$$\begin{aligned} \mathbf{g} &= \sum_{k=1}^{NDOM} \left\{ -\mathbf{K}_{BI}^k (\mathbf{K}_{II}^k)^{-1} \mathbf{p}_I^k + \mathbf{p}_B^k \right\} \\ &\equiv \sum_{k=1}^{NDOM} \left\{ -\mathbf{K}_{BI}^k (\mathbf{K}_{II}^k)^{-1} ({}^{t+\Delta t} \mathbf{R}_I^k - {}^{t+\Delta t} \mathbf{Q}_I^k) + ({}^{t+\Delta t} \mathbf{R}_B^k - {}^{t+\Delta t} \mathbf{Q}_B^k) \right\}. \end{aligned} \quad (15)$$

Equation (15) is a residual vector of the CG in the first step of the CG in the Newton-Raphson iterative step, in which the initial vector of the CG is set to zero. This equation shows that the order of the residual vector of CG and that of the out-of-balance force of NR are the same.

4.3 Convergence criteria of CG and Newton-Raphson methods

A norm of the residual vector of CG normalized by a load vector is often used in the convergence criterion of the CG method. In the Newton-Raphson iterative procedure, the load term in Eq. (14) (i. e., the out-of-balance force vector) rapidly converges to zero vector. As shown in Eq. (12), $\Delta \mathbf{u}^i$ ($i = 1, 2, 3, \dots$) are summed to calculate ${}^{t+\Delta t} \mathbf{u}$, which means that

accuracy of ${}^{t+\Delta t}_i \mathbf{u}$ depends on the absolute error of $\Delta \mathbf{u}^i$. Therefore, the convergence of the CG is appropriately achieved when the residual norm of CG becomes sufficiently small compared with the norm of the incremental load vector. The norm of the out-of-balance force of NR does not have to be used.

In the DDM, the convergence of the Newton-Raphson method has to be judged carefully because only the interface DOFs are solved by the CG. The norm of the out-of-balance force of NR concerning with the interface DOFs cannot be smaller than the norm of the residual of CG, whereas that concerning with the internal DOFs may converge to a level of the round-off error. If the convergence criterion of the CG is relaxed compared with the order of the round-off error, the residual for subdomain internal DOFs converges to much smaller value than that of the interface DOFs.

In this study, a norm of vector is defined as a maximum absolute component of the vector. ε_{CG0} is the norm of the residual of CG in the first CG step of the first incremental step. ε_{CG} is the norm of the residual of CG at an iterative step of CG. When the initial vector of CG is zero vector, ε_{CG0} is equal to the norm of an incremental load vector including contributions of prescribed displacements.

The following convergence criterion for CG is adopted:

$$\frac{\varepsilon_{CG}}{\varepsilon_{CG0}} < \text{TOL}_{CG}, \quad (16)$$

where TOL_{CG} is a tolerance. Then, the following convergence criterion for the Newton-Raphson method is proposed. The norm of the out-of-balance force of NR concerning with the interface DOFs and that concerning with the internal DOFs are denoted by ε_{NR-INB} and ε_{NR-INN} , respectively. As described earlier, when TOL_{CG} in Eq. (16) is relaxed, ε_{NR-INB} converges to the order of TOL_{CG} . Hence, the following convergence criterion for the Newton-Raphson method using ε_{NR-INN} is adopted:

$$\frac{\varepsilon_{NR-INN}}{\varepsilon_{CG0}} < \text{TOL}_{NR}, \quad (17)$$

where TOL_{NR} is a tolerance. Since ε_{NR-INB} can not be smaller than TOL_{CG} , TOL_{NR} is set equal to TOL_{CG} , i. e.:

$$\text{TOL}_{CG} = \text{TOL}_{NR}. \quad (18)$$

Keyes et al., or Kacou and Parsons investigated on the convergence criteria of the above two iterations [Kacou and Parsons, 1993][Parsons, 1997][Keyes and Venkatakrishnan, 1996]. According to their studies, total computation time may become short when the criterion for the inner iteration, i.e., the CG iteration, is relaxed during the Newton-Raphson iteration steps. In the following Chapter 6, however, the CG iteration will be continued until Eq. (16) is satisfied since an objective of our study is to confirm the quadratic convergence of the Newton-Raphson method in large scale structural analyses.

4.4 Initial vector for CG

Soneda and Yagawa showed that in the incremental analysis the number of CG iteration decreased if a solution vector of the previous incremental step is used for the initial vector of the CG method [Soneda et al., 1991]. This technique is adopted in the present study. The

analogous technique is also tested for the Newton-Raphson iterative steps in the present study. However, the number of CG iteration sometimes increased. Thus, in the following illustrative examples, the initial vector is always set to zero in the Newton-Raphson iteration steps.

5. EVALUATION OF MEMORY, COMMUNICATION AND COMPUTATION SPEED

5.1 Overview

This chapter describes the evaluation of amounts of required memory and communication, and computation speed of the present method in Chapter 3. In the following the method is called the improved method or the improved HDDM. Such an evaluation of the original HDDM can be found in Ref. [1]. Here the following four issues are evaluated:

- (1) Memory requirement in Parents in case of elastic-plastic analysis
- (2) Memory requirement in Children when using the improved method
- (3) Improvement of computation speed when using the improved method
- (4) Parallel efficiency

For these purposes, 1 M DOFs and 10 M DOFs simple cubes as shown in Tables 1 and 2, respectively, are analyzed. The lower surface of the cube is completely fixed, while uniform vertical displacements are prescribed on the upper surface in the upper direction. The material is steel.

5.2 Memory requirements in Parents and Children and amount of communication

Table 3 illustrates the total required memories in Parents and Children, and the number of DOFs in the interface nodes that are solved by the CG method. Those values are estimated for the cubes with different numbers of subdomains. As shown in Eqs (7) and (9), stresses and displacements have to be stored in the elastic-plastic analysis (or generally in nonlinear analyses). In the present method based on the HDDM, they are stored in Parents, and sent to and received from Children as shown in Fig. 1. The amount of required memory in Children is large because the data concerning with \mathbf{K}^k and $(\mathbf{K}_{II}^k)^{-1}$ in Eqs (1) and (4) are stored. The amount depends on both the number of subdomains and the skyline widths of the stiffness matrices of subdomains, i. e., it becomes smaller as the number of subdomains increases. It should be noted here that, when the improved method is not used, the amount of memory used in Children is very small.

Table 4 shows sizes of arrays in all Parents, and amounts of communication between Parents and Children in each stage of CG iteration. The technique presented in Chapter 3 is employed. There are the following three stages in CG iteration: (1) the first step where \mathbf{K}^k and $(\mathbf{K}_{II}^k)^{-1}$ are calculated, (2) the iterative steps, and (3) the last step where the displacements corresponding to the internal DOFs and stresses are recovered. The arrays needed for the elastic analysis and for the elastic-plastic analysis are separately illustrated. The total amount of memory needed for the elastic analysis is less than 100 M bytes. On the other hand, that for the elastic-plastic analysis is 5.1 times as large as that for the elastic analysis.

When the improved method is used, the stiffness matrices of subdomains are constructed only in the first CG step where a large amount of communication is needed especially in the elastic-plastic analysis. However, in other CG steps, only residual vectors updated displacements vectors and some index arrays are communicated. In these stages, there are no differences between the elastic analysis and the elastic-plastic analysis. When the CG is converged, a large amount of communication is again necessary for the recovered displacements for the internal DOFs and stresses etc.

5.3 Evaluation of computation speed

Elastic analyses of the 1 M DOFs and 10 M DOFs cubes are performed to evaluate computation speed. TOL_{CG} in Eq. (16), i. e., a tolerance of the convergence of the CG, is set to 10^{-6} . The 1 M DOFs model is subdivided into 2000 or 5000 subdomains, and the 10 M DOFs model is subdivided into 10000 or 20000 subdomains. These domain decomposition tasks are carried out by using an automatic domain decomposer developed in our project [Takubo et al., 1998]. This domain decomposer employs graph partitioning packages METIS [Karypis and Kumar, 1995] and PARMETIS [Karypis and Kumar, 1997] as basic modules. The following computations are performed on HITACHI SR2201 in the computer center of the University of Tokyo, which is a distributed-memory massively parallel computer with 1024 PEs. Each PE has 224 M Bytes memory. All PEs are connected to each other by 3D crossbar switch whose data transfer rate is 300 M Bytes per second. Peak performance of the SR2201 is 0.3 T Flops. This system is operated by a batch job system. Maximum elapsed time depends on the number of PEs concurrently used. For example, maximum elapsed time of a job with 1024 PEs is limited to 1 hour. The computation times described below are elapsed times except for file IO times.

Tables 5 and 6 illustrate computation speed for 1 M DOFs and 10 M DOFs cubes, respectively. Computation times are measured by changing the numbers of subdomains and PEs. The ratio of computation times at the first CG step and the other CG step clearly shows improvement of computation speed due to the improved method in Chapter 3. Figures 5 and 6 show a kind of speed up defined in the figures. These measures are adopted here because the problems cannot be solved by 1 PE, and computation times change due to the numbers of subdomains and parts.

When the number of subdomains increases, computation time for 1 CG step decreases and a whole computation time also decreases, although the number of CG steps increases. However, when the number of PEs is large and the number of subdomains is too large, computation time increases. This is because computation time for the static condensation of a subdomain is so short that concentration of communication occurs. This problem can be avoided by increasing the number of Parents. In Table 5, the numbers of Parents are increased when the numbers of PEs are increased.

Table 1 Cube model with 1 M DOFs

Total DOFs	1,073,733
Number of elements	343,000 (70^3)
Number of nodes (8-node hexahedral element)	357,911 (71^3)
Young's modulus	193 GPa
Poisson's ratio	0.275

Table 2 Cube model with 10 M DOFs

Total DOFs	10,125,000
Number of elements	3,307,949 (149^3)
Number of nodes (8-node hexahedral element)	3,375,000 (150^3)
Young's modulus	193 GPa
Poisson's ratio	0.275

Table 3 Total required memories in Parents and in Children, and DOFs in the interface nodes
(Unit of memory : Byte, 1 M = 10^6 Bytes, 1G = 10^9 Bytes)

Total DOFs	Number of parts	Number of subdomains	Parents (Elastic)	Parents (Elastic-Plastic)	Children	DOFs in the interface nodes
1 M	4	500	67 M	421 M	3.29 G	340010
1 M	4	2000	88 M	450 M	2.21 G	512009
1 M	4	4000	102 M	470 M	1.57 G	621222
10 M	40	10000	771 M	4220 M	30.81 G	4270245
10 M	40	20000	883 M	4374 M	21.05 G	5165553

Table 4 Amount of memory required in Parents and that of communication; 1 M DOFs model with 2000 subdomains; 1 M Byte = 10^6 Bytes

Array	Size (MByte)	TYPE*1	Commun.*2
\mathbf{g} (Eq. (6))	4.1	E	F<, I<, L<*3
$\sum_{k=1}^{NDOM} \mathbf{u}_B^k$ (Eq. (6))	4.4	E	F>, I>, L>*3
Preconditioner for CG	4.1	E	F<
Work arrays for CG	12.9	E	
Subdomain information	0.1	E	F>, I>, L>
Part information	2.7	E	
Index for interface DOFs	24.8	E	F><, I><, L><*4
Boundary conditions	0.4	E	F>, L>
Coordinates	8.9	E	F>, L>
Connectivity	11.0	E	F>, L>
Displacements	14.4	E	L<
${}^t\boldsymbol{\sigma}$ (Eqs (6) and (9))	132.0	P	F>, L>
${}^{t+\Delta t}\boldsymbol{\sigma}$ (Eq. (9))	132.0	P	F>, L><
${}^t\boldsymbol{\sigma}_Y$ (Eq. (6))	22.0	P	F>, L>
${}^{t+\Delta t}\boldsymbol{\sigma}_Y$ (Eq. (6))	22.0	P	F>, L><
Status of yielding	11.0	P	F>, L><
${}^{t+\Delta t}{}_t\mathbf{u}$ (Eq. (12))	14.4	P	L><
${}^{t+\Delta t}\mathbf{R}$ (Eqs (10) and (11))	14.4	P	F>, L><
${}^{t+\Delta t}\mathbf{Q}^{i-1}$ (Eq. (11))	14.4	P	F>, L><
Total (Elastic)	87.9		
Total (Elastic-Plastic)	449.5		
Total Communications (Elastic)	F> : 55.1, I> : 34.8, L> : 55.1,	F< : 19.0 I< : 14.9 L< : 29.3	
Total Communications (Plastic)	F> : 402.3, I> : 34.8, L> : 402.3,	F< : 19.0 I< : 14.9 L< : 237.3	

*1 E : Elastic, P : Plastic

*2 F : First CG step, L : Last CG step, I : Other steps

> : From Parents to Children, < : From Children to Parents

*3 Amount of communication becomes several times larger than amounts of data in memory, because the interface nodes are shared by some subdomains.

*4 1/5 of them are sent from Child to Parent.

Table 5 Computation time of 1 M DOFs model

Subdomains	PEs	Parents	Children	CG steps	Total time	1CG(1st)	1CG(other)	(1st)/(other)
1600	64	4	59	260	774	-	-	-
2000	16	2	13	241	2936	78.8	12.0	6.6
2000	16	4	11	Not enough memory in Children				
2000	32	4	27	265	1658	40.4	6.2	6.6
2000	64	4	59	265	724	16.8	2.7	6.2
2000	128	4	123	265	387	8.81	1.44	6.1
2000	128	10	117	269	377	8.82	1.38	6.4
2000	256	4	251	265	390	5.62	1.46	3.9
2000	256	10	245	255	202	4.9	0.78	6.3
4000	16	4	11	217	2901			
4000	16	2	13	231	2099			
4000	32	4	27	217	1001	28.7	4.5	6.4
4000	64	4	59	217	477	12.3	2.2	5.7
4000	128	4	123	217	582	6.7	2.7	2.5
4000	128	10	117	311	351			
4000	256	4	251	217	500	5.4	2.4	2.3
5000	16	2	13	282	2386	53.0	8.3	6.4
5000	32	4	27	283	1229	26.9	4.3	6.3
5000	64	4	59	283	589	11.5	2.1	5.6
5000	128	10	117	311	351	6.7	1.1	6.0
5000	256	10	245	311	380	3.9	1.2	3.2

Table 6 Computation time of 10 M DOFs model

Subdomains	PEs	Parents	Children	CG steps	Total time	1CG(1st)	1CG(other)	(1st)/(other)
10000	256	40	215	508	5318	85.2	10.3	8.2
10000	512	40	471	508	2481	37.3	4.8	7.7
10000	641	40	600	508	2203	33.6	4.1	8.3
10000	1024	40	983	508	1449	35.8	2.8	12.8
20000	256	40	215	539	4011	47.3	7.4	6.4
20000	512	40	471	539	1954	24.5	3.6	6.8
20000	641	40	600	539	1564	23.3	2.9	8.1
20000	1024	40	983	539	1973	30.2	3.6	8.3

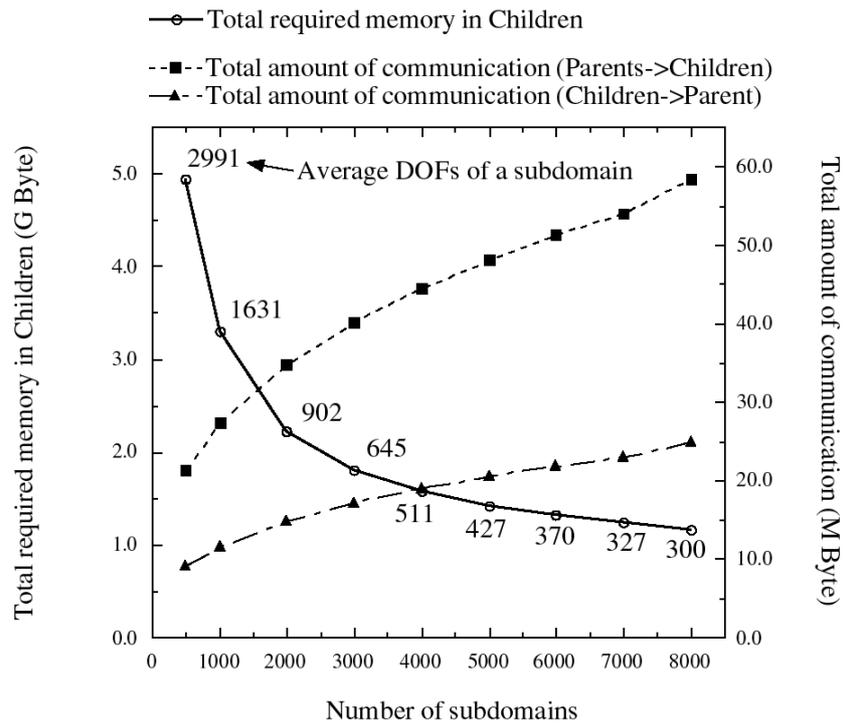


Fig. 4 Required memory in Children and total amounts of communication versus number of subdomains for 1M DOFs cube

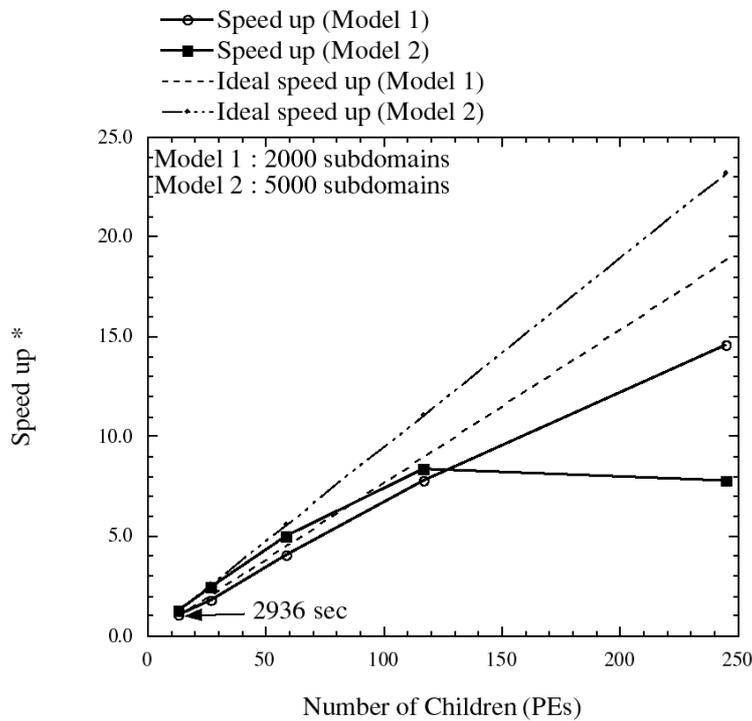


Fig. 5 Speedup for 1M DOFs cube; * Note that the speedup is defined as follows:

$$\text{Speedup} = \frac{\text{Computation time of Model 1 by 13 Children}}{\text{Computation time}}$$

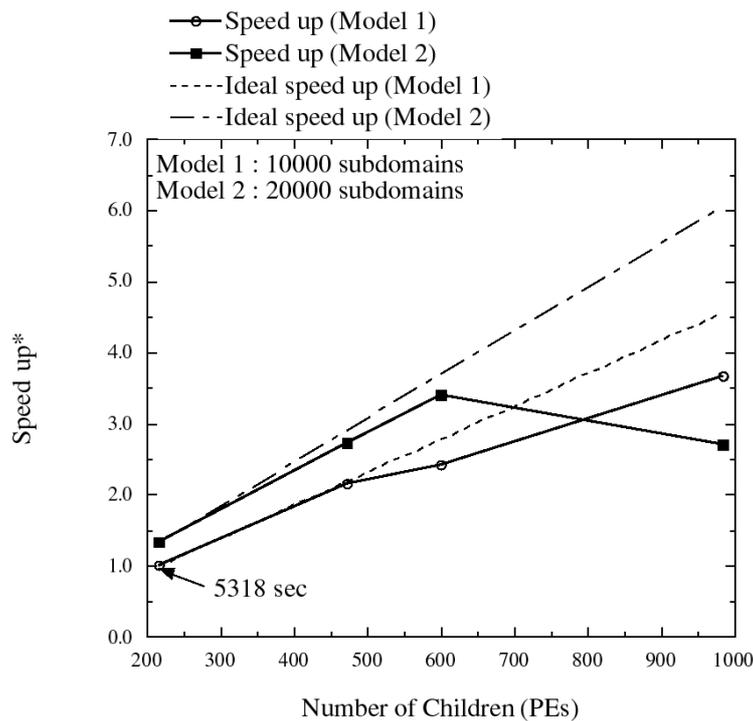


Fig. 6 Speedup for 10M DOFs cube; * Note that the speedup is defined as follows:

$$\text{Speed up*} = \frac{\text{Computation time of Model1 by 215 Children}}{\text{Computation time}}$$

6. NUMERICAL EXAMPLES AND DISCUSSIONS

6.1 Elastic-plastic analysis of HTTR model with 1.3 M DOFs

Figure 7 shows the analysis model of a support structure of the high temperature engineering test reactor (HTTR) [Arai et al., 1992]. The shape of model is rather simplified, although the real structure has more holes. This model is specially prepared to examine fundamental performance of the developed analysis system. A basic coarse mesh with 3924 DOFs is first automatically generated by the intelligent local approach proposed by Wada et al. [Wada et al., 1999]. Then, each hexahedral element in the coarse mesh is recursively subdivided into eight hexahedral elements. The first subdivision leads to a model with 24999 DOFs, and the second subdivision a model with 175833 DOFs. After three times subdivision, a model with 1.31 M DOFs that is shown in Fig. 7 and Table 7 is obtained.

Since the structure is made of graphite, the creep analysis in high temperature considering thermal strain is necessary from a practical point of view. In this study, however, the material of the structure is assumed to be steel, and the elastic-plastic analysis is performed as a preliminary computation. As shown in Fig. 7, the right-hand surface of the model is fixed, while uniform displacements, whose direction is depicted by arrows in the figure, are prescribed in the left-hand surface. Figure 8 shows the hierarchically decomposed model that is subdivided by the automatic domain decomposer [Takubo et al., 1998]. Schematic flow of

the present parallel finite element analysis system including pre- and post-processes is illustrated in Ref. [Miyamura et al., 1999].

The von Mises yield criterion with the linear isotropic hardening and the associated flow rule are assumed as the elastic-plastic constitutive equations. As a preliminary analysis, the small model with 0.175 M DOFs is analyzed to check convergence of the Newton-Raphson method. Tolerances for the convergence criteria of the CG and the Newton-Raphson methods are set as $TOL_{CG} = 10^{-15}$ and $TOL_{NR} = 10^{-15}$, respectively. A PC cluster consisting of 16 PCs with DEC alpha 21164 533 MHz processors (OS : Linux) is utilized in this analysis. Figure 10 illustrates the convergence of the CG method in each incremental/Newton-Raphson iterative step, and the convergence of the Newton-Rahson method. As described in Section 4.3, the order of the residual of CG and that of the out-of-balance force of NR are closely related to each other. Therefore, $\varepsilon_{CG}/\varepsilon_{CG0}$, $\varepsilon_{NR-INN}/\varepsilon_{CG0}$ and $\varepsilon_{NR-INB}/\varepsilon_{CG0}$, whose definitions are given in Section 3.3 are plotted in the same figure. Since the HDDM solver is called several times, the horizontal axis shows the accumulated number of the CG iteration steps. The quadratic convergence of $\varepsilon_{NR-INN}/\varepsilon_{CG0}$ and $\varepsilon_{NR-INB}/\varepsilon_{CG0}$ is clearly observed. Total computation time until the Newton-Raphson method converges is 5676 seconds.

Finally, the elastic-plastic analysis of the 1.31 M DOFs model is performed. Figure 9 shows the convergence of the CG iteration in an elastic analysis. The elastic analysis of this model takes about 20 minutes using 256 PEs of HITACHI SR2201. The maximum elapsed time of a batch job for 256 PEs on the MPP in the University of Tokyo is limited to two hours, and the Newton-Raphson method does not converge within a batch job. To finish the computation in a batch job, the convergence criterion of the CG is slightly relaxed, i.e., $TOL_{CG} = TOL_{NR} = 10^{-12}$. Figure 11 shows the convergence of the CG and the Newton-Raphson methods. The quadratic convergence of the Newton-Raphson method is attained except for the last step, in which $\varepsilon_{NR-INB}/\varepsilon_{CG0}$ does not reach $TOL_{CG} = 10^{-12}$. On the other hand, $\varepsilon_{NR-INN}/\varepsilon_{CG0}$ fully converges. Total computation time until the Newton-Raphson method converges is 7053 seconds.

Figure 12 shows a distribution of the equivalent stresses and deformation. Due to the high resolution of the analysis, stress distribution is precisely observed. Figure 13 shows the plastic region. Figures 12 and 13 are rendered using a parallel visualization system developed in our project [Yoshimura et al., 1998].

Table 7 Nuclear structure model with 1.3 M DOFs

Total DOFs	1,312,917
Number of elements (8-node hexahedral element)	407,040
Number of nodes	437,639
Number of subdomains	2,400
Number of parents	8
DOFs in the interface nodes	597,459
Young's modulus	193 GPa
Poisson's ratio	0.275
Strain hardening parameter	9.08 GPa

Initial yield stress

544 MPa

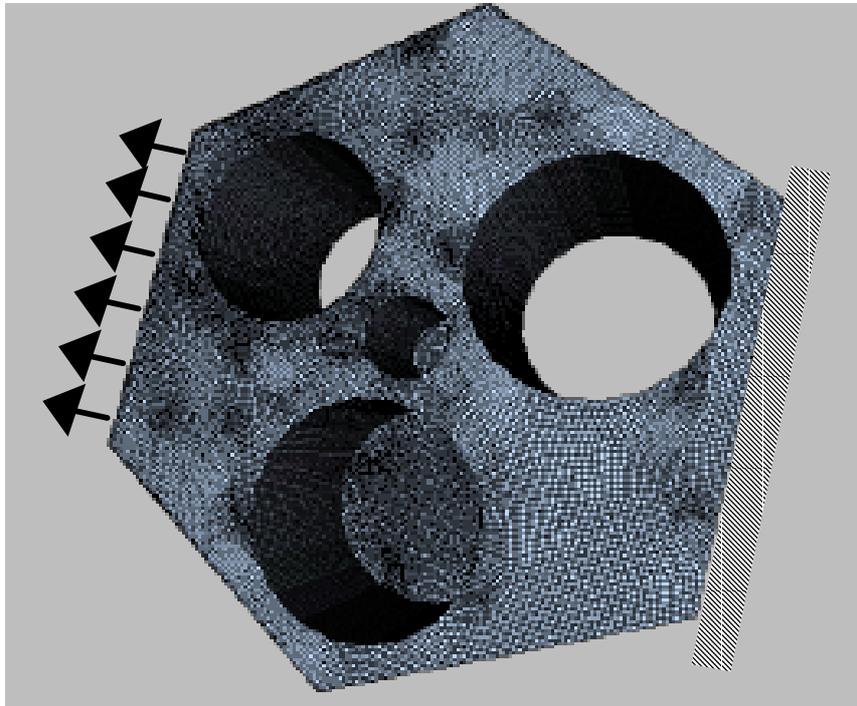


Fig. 7 Hexahedral mesh with 1.3 M DOFs

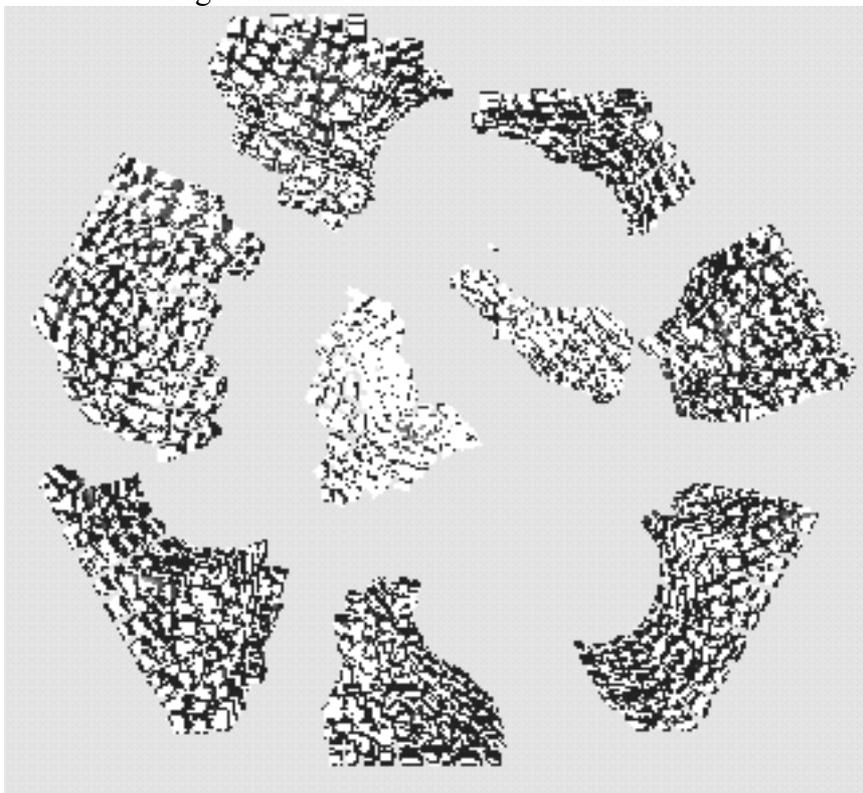


Fig. 8 Hierarchical domain decomposition; Parts: 8; subdomains: 2400

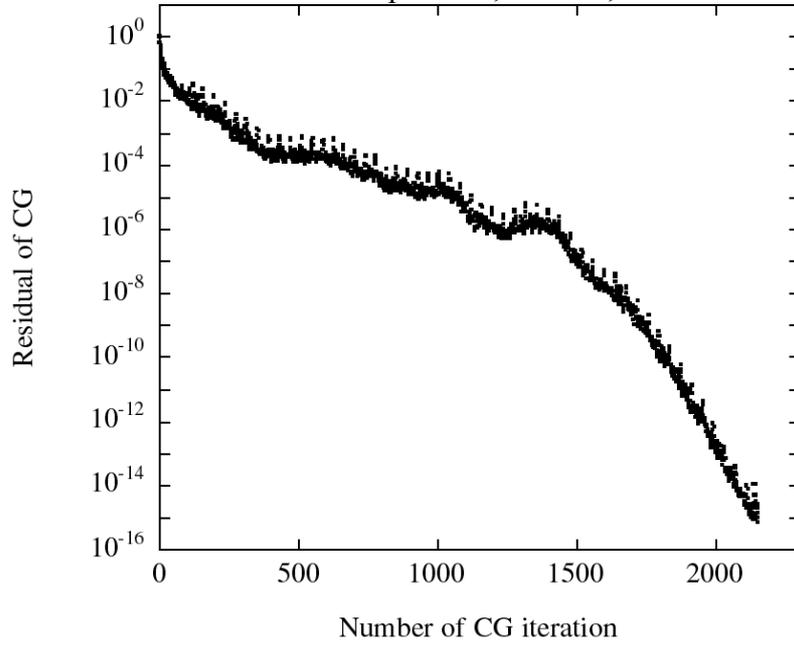


Fig. 9 Residual of CG in elastic analysis of the nuclear structure model with 1.3M DOFs

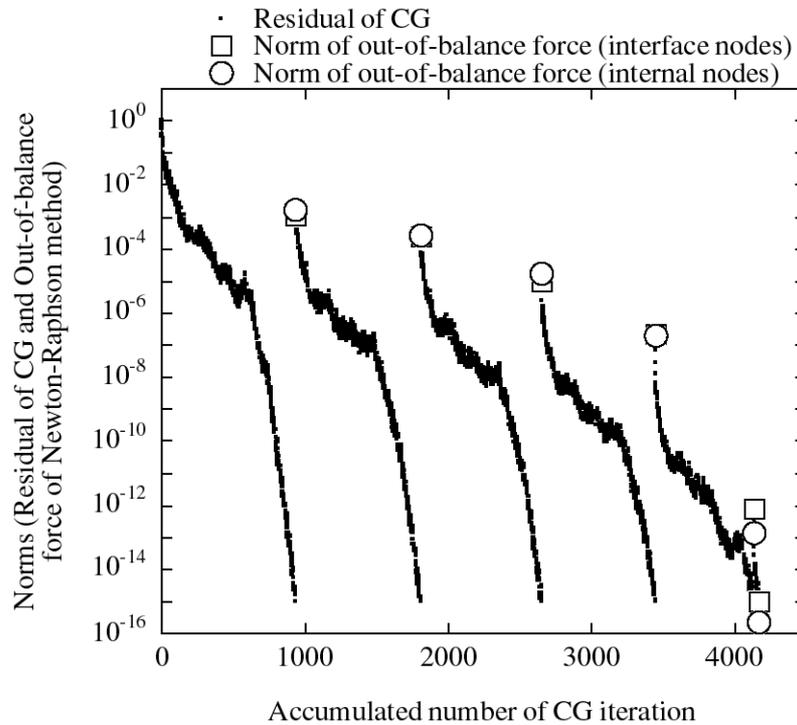


Fig. 10 Residual of CG and out-of-balance force in Newton-Raphson method; elastic-plastic analysis of the nuclear structure model with 0.175M DOFs

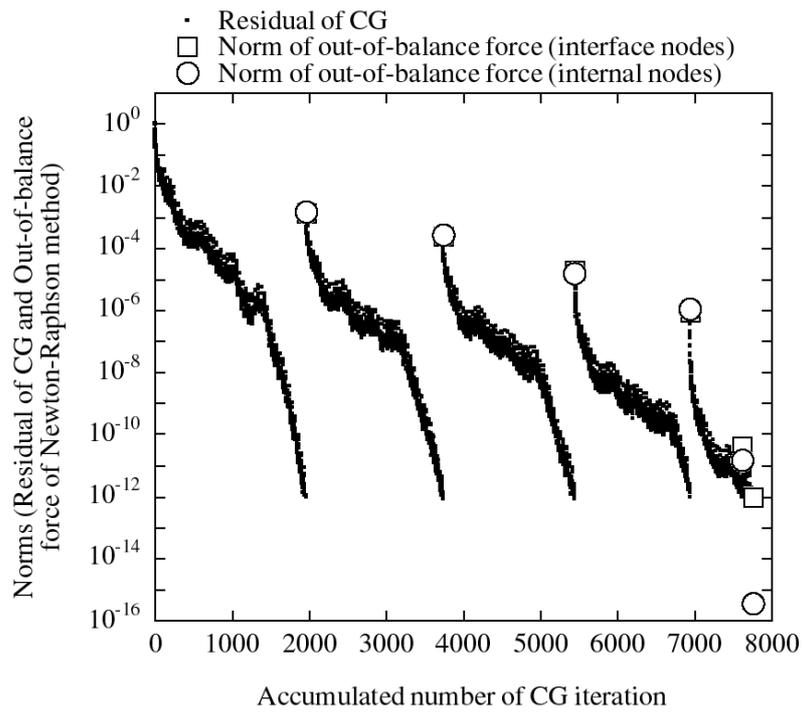


Fig. 11 Residual of CG and out-of-balance force in Newton-Raphson method; elastic-plastic analysis of the nuclear structure model with 1.3M DOFs

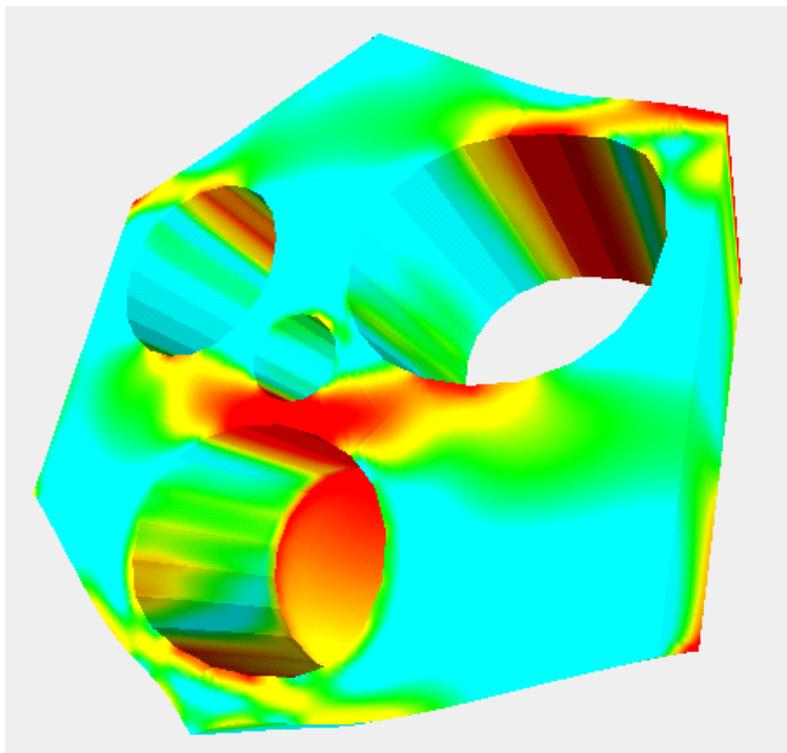


Fig. 12 Distribution of equivalent stress and deformed configuration of 1.3M DOFs model; displacements are magnified by 40 times.

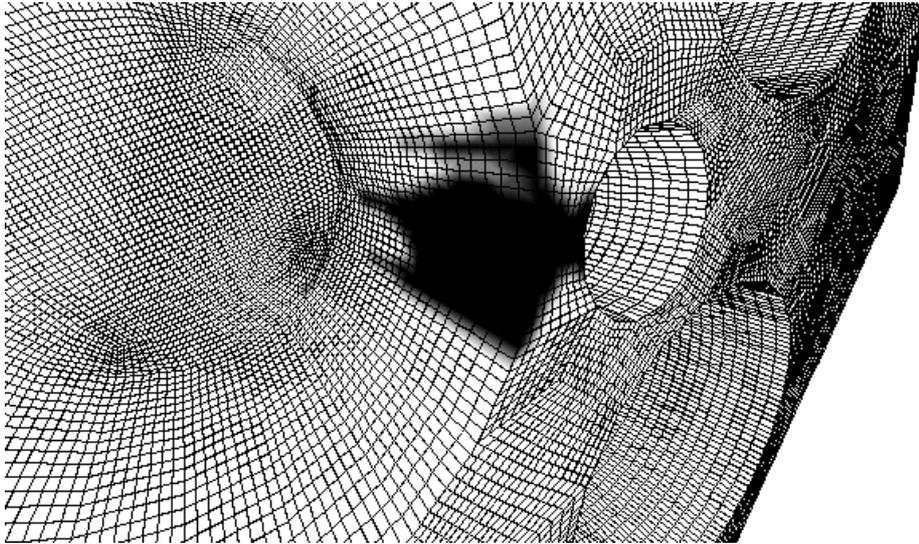


Fig. 13 Distribution of plastic region of 1.3M DOFs model

6. 2 Pressure vessel with 3 M DOFs

Figure 14 shows the geometry model (surface patches) of a pressure vessel of a nuclear power plant. An analysis model shown in Fig. 15 is generated from this geometry model by using the volume pixel (voxel) method [Ishii and Nakazato 1998][Kikuchi et al., 1999]. The voxel method is often used in these days because the mesh is easily generated from the surface patches or the image data by simply eliminating unnecessary voxels. Some techniques to quickly generate the voxel meshes are presented in Ref [Miyamura, Wada et al., 1999]. Total DOFs of this voxel mesh is about 3 M. Material is steel. Boundary conditions are shown in Fig. 15. The elastic-plastic analysis of this model is carried out by using 256 PEs of SR2201. In this case, 40 incremental steps are computed by restarting the computation. Total number of the batch jobs is 33. Sum of computation times of the first job and all the restarting jobs is about 42 hours.

Figure 16 shows the convergence of the CG and the Newton-Raphson method. Only $\epsilon_{CG} / \epsilon_{CG0}$ and $\epsilon_{NR-INB} / \epsilon_{CG0}$ in Section 3.3 are shown in this figure. The convergence criteria are relaxed in this analysis, i.e., $TOL_{CG} = TOL_{NR} = 10^{-6}$, to reduce computation time. In incremental steps, a solution vector of the previous incremental step is used for the initial vector for the CG method (see also Ref. [Soneda et al. 1991]). The Newton-Raphson method successfully converges, although the convergence criterion of the CG is relaxed. As the number of the incremental step becomes large, the number of the CG iterative steps becomes small. This is because the difference between the incremental displacement vector in the current incremental step and that in the previous incremental step, i.e., the initial vector for the CG method, becomes small as the analysis proceeds. Distributions of the equivalent stresses and deformations are shown in Fig. 17. Distributions of the plastic regions are also shown. As the deformation becomes large, areas of the plastic regions become large. Since the surface of the voxel mesh is not smooth, the plastic regions spread along the indentations in the mesh. These figures are also depicted by using the parallel visualization system.

Table 8 Voxel mesh of pressure vessel

Total degrees of freedom	3,046,311
Number of nodes	1,015,437
Number of elements	895,030
Number of subdomains	8,000
Number of Parents	10
DOFs in the interface nodes	1,499,430
Young's modulus	193 GPa
Poisson's ratio	0.275
Strain hardening parameter	9.08 GPa
Initial yield stress	544 MPa

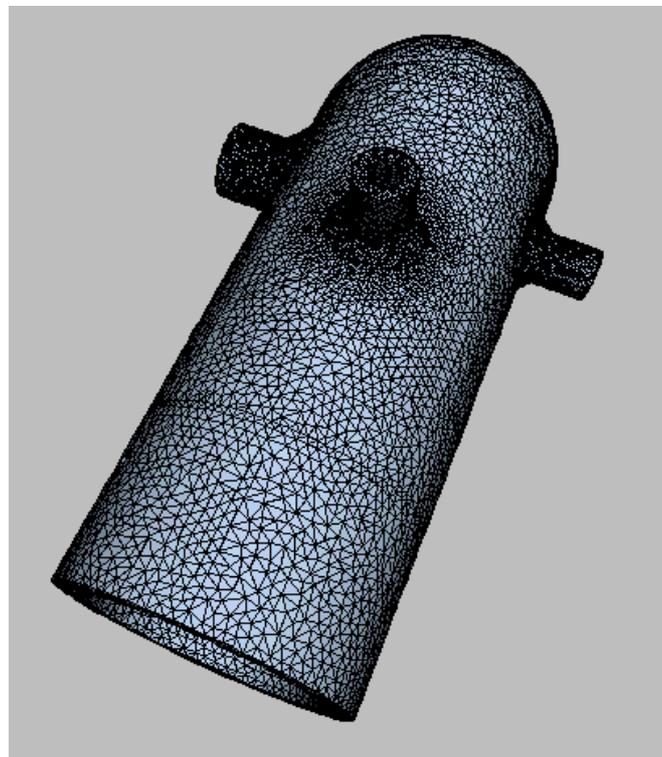


Figure 14 Surface patches of pressure vessel

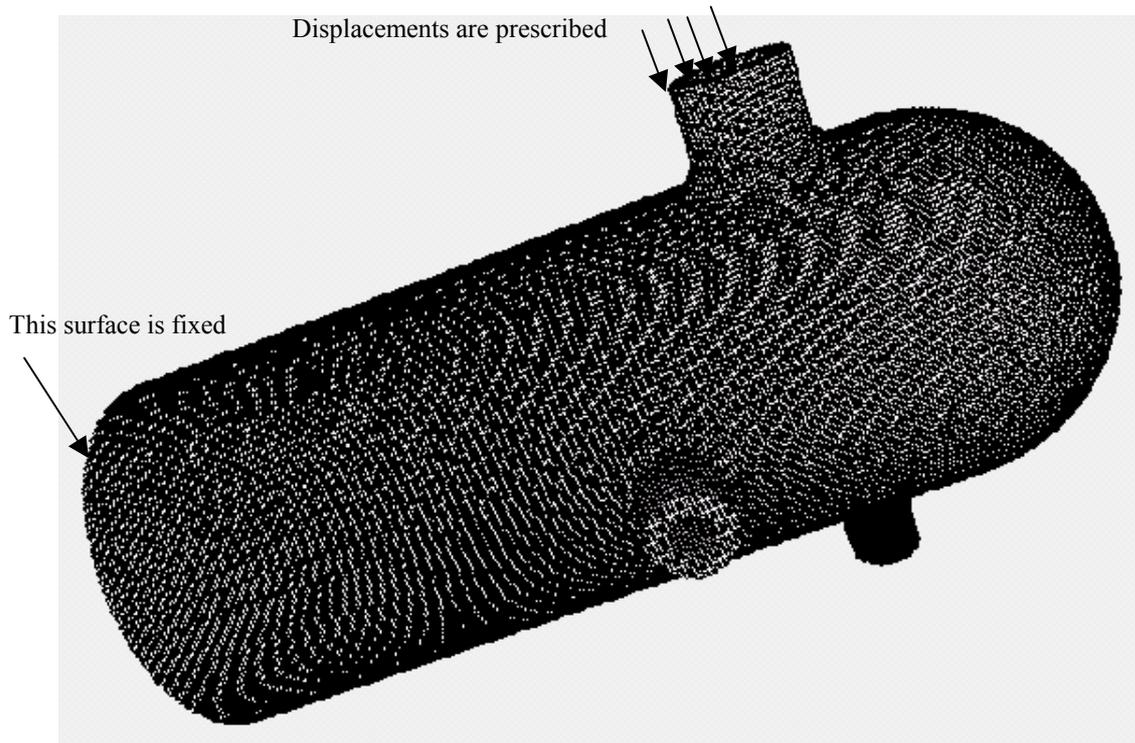


Figure 15 Voxel mesh of pressure vessel

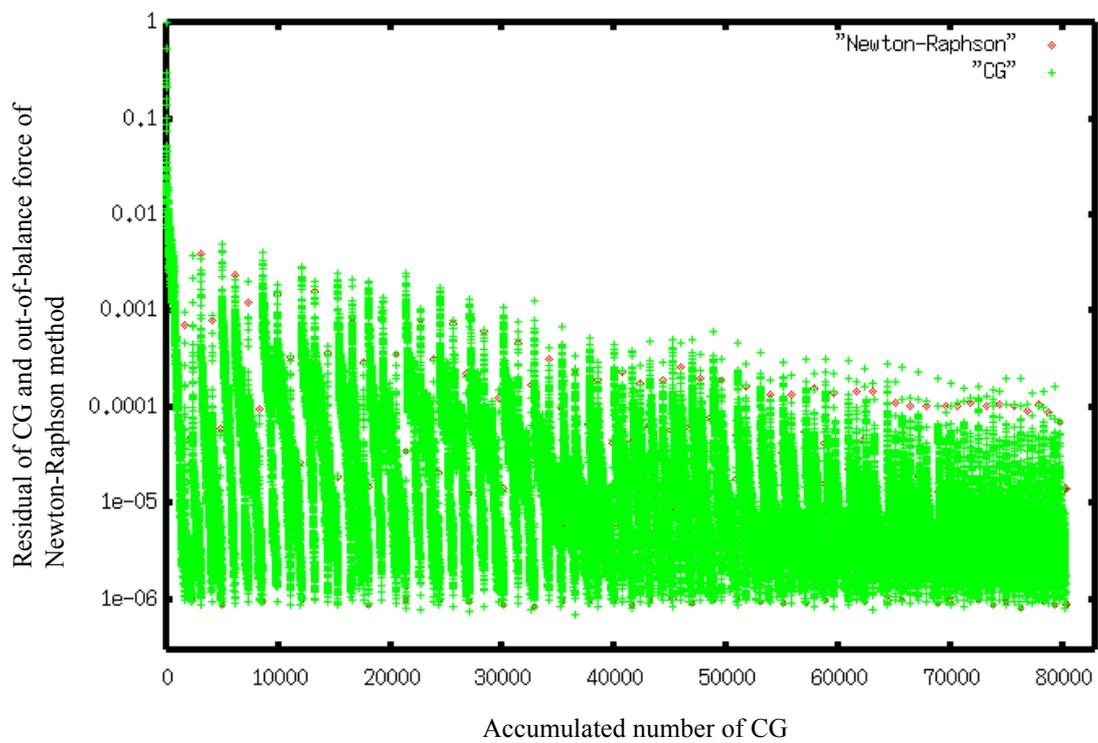


Figure 16 Residual of CG and out-of-balance force in Newton-Raphson method

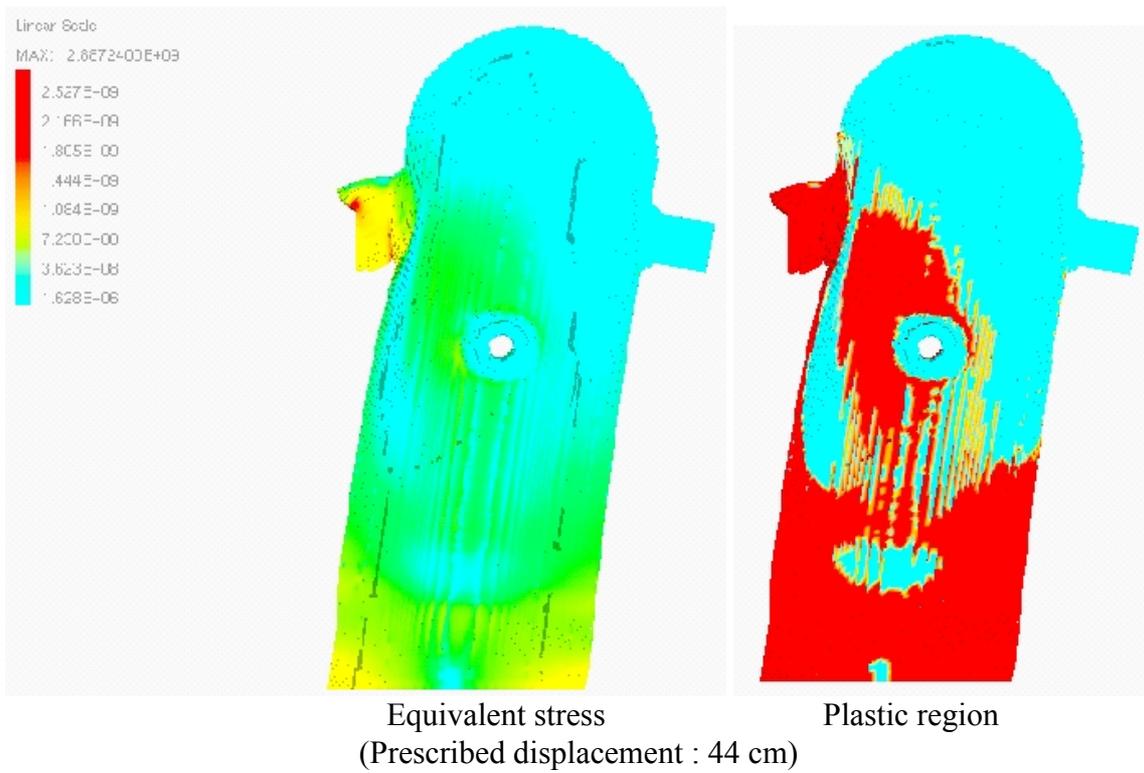
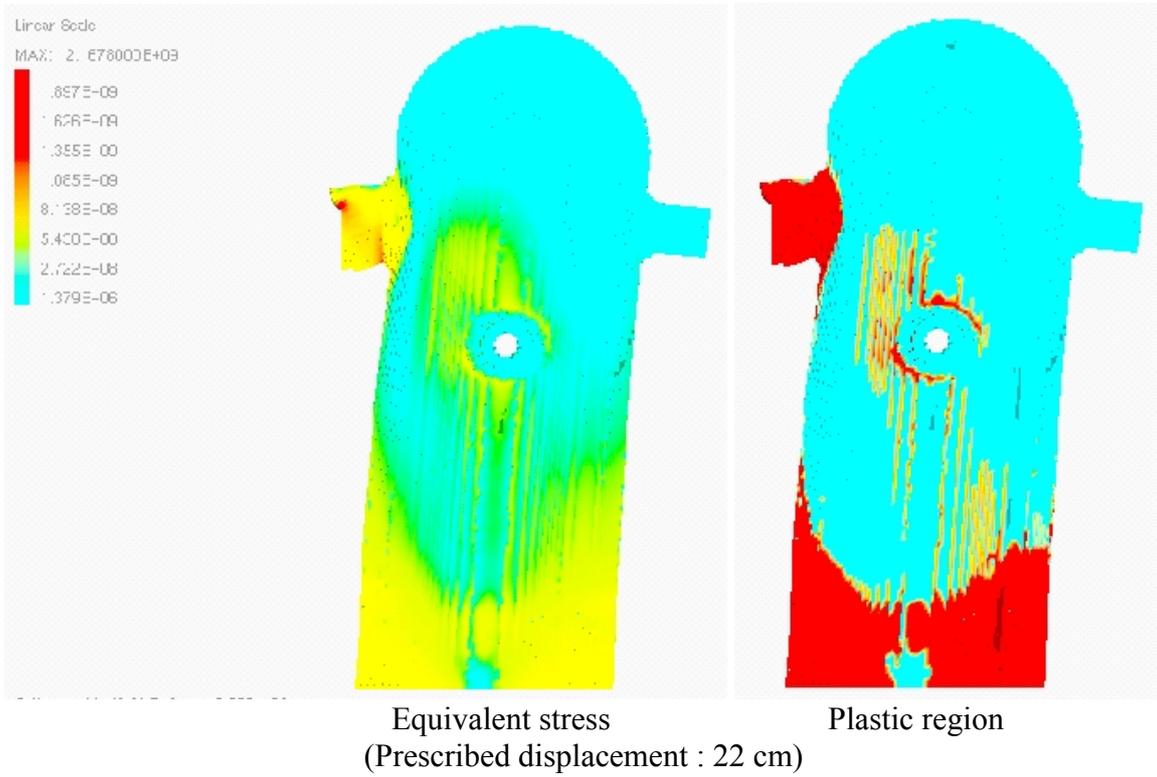


Figure 17 Distributions of equivalent stresses and plastic region, and deformation (displacements are magnified by 5 times)

6.3 Elastic-plastic analysis of 10 M cube

The elastic-plastic analysis of the 10 M cube shown in Table 2 is performed using 1024 PEs of HITACHI SR2201. The bottom of the cube is fixed, and uniform displacements in the upper direction are prescribed on the top of the cube. The number of subdomains is 10000. In an elastic analysis, the residual of CG converges to 10^{-10} within one hour, which is the maximum permitted elapsed time of a batch job. To finish the elastic-plastic analysis within a batch job, the convergence criteria of the CG and the Newton-Raphson methods are relaxed, i. e., $TOL_{CG} = TOL_{NR} = 10^{-5}$. Figure 18 shows the convergence of the CG and the Newton-Raphson method. The Newton-Raphson method converges in one iterative step since the model shape is rather simple and the convergence criterion is relaxed. $\epsilon_{NR-INN} / \epsilon_{CG0}$ converges to the smaller value than $\epsilon_{NR-INB} / \epsilon_{CG0}$. Although this example is simple, it is shown that the elastic-plastic analyses of ten millions DOFs class problems are possible by using a currently available MPP with 0.3 T Flops peak performance.

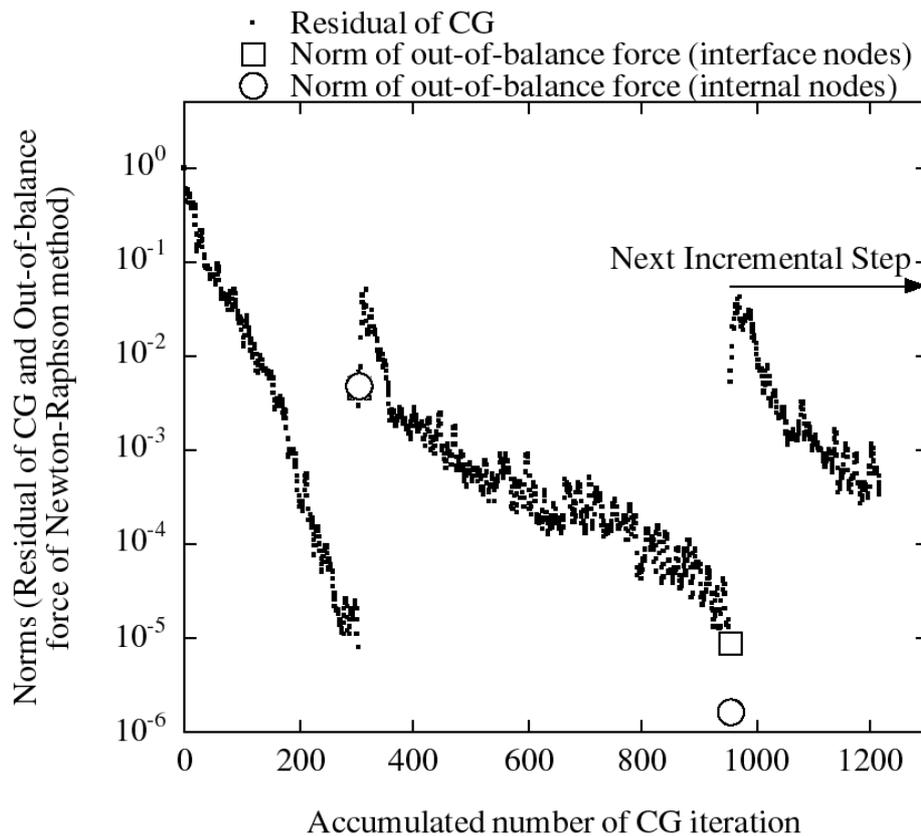


Fig. 18 Residual of CG and out-of-balance force in Newton-Raphson method; elastic-plastic analysis of the cube model

7. CONCLUDING REMARKS

In this study, a technique to improve computation speed of the hierarchical domain decomposition method (HDDM) is first presented. Then, several techniques to apply the HDDM to the elastic-plastic analysis of solids are explained: (1) implementation on parallel computers, (2) convergence criteria of the CG and Newton-Raphson methods and (3) evaluation of amounts of memory and communication.

The models analyzed as illustrative examples are a simplified nuclear structure with 1.31 M DOFs and a cube with 10 M DOFs. The improved HDDM is first tested by analyzing some elastic problems. Then, the elastic-plastic analyses are performed. In the analysis of the nuclear structure, quadratic convergence of the Newton-Raphson method was observed. The Newton-Raphson method also successfully converged in the 10 M DOFs cube problem.

The techniques presented in this study can be applied not only to the elastic-plastic analysis but also to general nonlinear analyses of solids using an implicit solver. The results of the illustrative examples show that simple nonlinear static finite element analyses of 1 M to 10 M DOFs can be done by using a currently available MPP with 0.3 T Flops peak performance. Further researches are going on to solve larger and more complex problems in a few years.

ACKNOWLEDGEMENT

Pre- and post-processing tools, i. e., automatic mesh generator, BC attachment tool, parallel domain decomposer and parallel visualizer are developed by Dr. Y. Wada (Univ. of Tokyo), Mr. H. Nitta (Allied Eng.), Mr. H. Takubo (Univ. of Tokyo) and Mr. S. Shoui (Allied Eng.), respectively, who are members of the ADVENTURE project. Their supports are gratefully acknowledged.

REFERENCES

- [1] Yagawa G and Shioya R, Parallel Finite Elements on a Massively Parallel Computer with Domain Decomposition. *Comp. Sys. in Eng.* 1993; 4: 495-503
- [2] Soneda N, Yagawa G and Yoshimura S, Elastic-Plastic Finite Element Analysis using Domain Decomposition Method. *Proc. SMiRT-11, Tokyo, Japan B, 1991: 243-248*
- [3] Yagawa G and Uchida Y, Elastic-Plastic Finite Element Analysis Using EWS Network. *Trans. of Japan Soc. of Mech. Eng.* 1994, Vol. 60, No. 580, Series A: 2737-2742 (in Japanese)
- [4] Shioya R and Yagawa G, Massively Parallel Computing for Large Scale Elasto-Plastic Analyses. *Computational Plasticity (Fundamentals and Applications)*, Owen DRJ, Onate E and Hinton E (Eds.), CIMNE, Barcelona 1997; 1843-1848
- [5] Miyamura T, Noguchi H, Shioya R, Yoshimura S and Yagawa G, Massively Parallel Elastic-Plastic Finite Element Analysis Using the Hierarchical Domain Decomposition Method. *Trans. of Japan Soc. of Mech, Series A* 1999, to appear (in Japanese)
- [6] Fahmy MW and Namini AH, A survey of parallel nonlinear dynamic analysis methodologies. *Comput Struct* 1994; Vol. 53, No. 4: 1033-1043
- [7] Farhat C and Roux FX, Implicit Parallel Processing in Structural Mechanics. *Comp Mech Advances* 1994; 2: 1-124
- [8] Farhat C, Chen PS, Mandel J and Roux FX, The two-level FETI method Part II:

- Extension to shell problems, parallel implementation and performance results. *Comput Methods Appl Mech Engrg* 1998; 155: 153-179
- [9] Kacou S and Parsons ID, A parallel multigrid method for history-dependent elastoplasticity computations. *Comput Methods Appl Mech Engrg* 1993; 108: 1-21
- [10] Parsons ID, Parallel adaptive multigrid methods for elasticity, plasticity and eigenvalue problems. *Parallel solution methods in computational mechanics*, Edited by Papadrakakis M, Wiley 1997; Chapter 4: 143-180
- [11] Fish J, Pandheeradi M and Belsky V, An efficient multilevel solution scheme for large scale non-linear systems. *Int J Num Methods Engrg* 1995; 38: 1597-1610
- [12] Nikishkov GP, Kawka M, Makinouchi A, Yagawa G and Yoshimura S, Porting an industrial sheet metal forming code to a distributed memory parallel computer. *Comput Struct* 1998; 67: 439-449
- [13] Nakajima K and Okuda H, Parallel iterative Solvers with Localized ILU Preconditioning for Unstructured Grids on Workstation Cluster. 4th Japan-US Sympo on Finite Element Methods in Large-Scale Computational Fluid Dynamics (to be appeared in *Int J Comp Fluid Dynamics*)
- [14] Venkatakrishnan V, Parallel implicit unstructured grid Euler solvers. *AIAA J* 1994; Vol. 32, No. 10: 1985-1991
- [15] Iizuka M, Okuda H and Yagawa G, Large scale parallel elastic-plastic analysis by GeoFEM. *Trans. of Japan Soc for Comp Eng and Sci*, to appear (in Japanese)
- [16] Garatani K, Feasibility study of GeoFEM for solving 100 million DOF problems, *Abstracts of ACES Inaugural Workshop 1999; Brisbane and Noosa, Australia*: 227-230
- [17] Tezduyar TE, Alibadi SK, Behr M and Mittal S, Massively parallel finite element simulation of compressible and incompressible flows. *Comput Methods Appl Mech Engrg* 1994; 119: 157-177
- [18] Kennedy JG, Behr M, Kalro V and Tezduyar TE, Implementation of implicit finite element method for incompressible flows on the CM-5, *Comput Methods Appl Mech Engrg* 1994; 119: 95-111
- [19] Keyes DE and Venkatakrishnan V, Newton-Krylov-Schwarz methods: Interfacing sparse linear solvers with nonlinear applications. *Zeitschr Angew Math Mech* 1996; 76 (Suppl. 1): 147-150
- [20] Keyes DE, How scalable is domain decomposition in practice. *Proc of the 11th Intl Conf on Domain Decomposition Methods*, Lai CH et al. eds 1998
- [21] Johan Z, Mathur KK, Johnsson L and Hughes TJR, Scalability of finite element applications on distributed-memory parallel computers. *Comput Methods Appl Mech Engrg* 1994; 119: 61-72
- [22] Belytschko T, Chiang HY and Plaskacz E, High resolution two-dimensional shear band computations: imperfections and mesh dependence. *Comput Methods Appl Mech Engrg* 1994; 119: 1-15
- [23] Yagawa G and Soneda N, *Parallel Computing*. Baifu-kan 1991 (in Japanese)
- [24] Bathe KJ, *Finite Element Procedure*, Prentice Hall 1996
- [25] Hisada T and Noguchi H, *Foundations and Applications of nonlinear FEM*. Maruzen 1995 (in Japanese)
- [26] Simo JC and Taylor RL, Consistent Tangent Operators for Rate-Independent Elastoplasticity. *Comp. Meth. in Appl. Mech. Eng.* 1985; 48: 101-118

- [27] Takubo H., Yoshimura S. and Yagawa G., Domain Decomposer of Ultra-large Scale Finite Element Mesh in Parallel/Distributed Environment. Proc. of 11 th JSME CMD Meeting 1998; No. 98-2: 575-576 (in Japanese)
- [28] Karypis G and Kumar V, Multilevel k-way Partitioning Scheme for Irregular Graphs. Tech. Rep. TR 95-064, Dept. of Comp. Sci., Univ. of Minnesota 1995
- [29] Karypis G and Kumar V, A Coarse-grain Parallel Multilevel k-way Partitioning Algorithm. Proc. of 8 th SIAM Conf. on Prallel Processing for Sci. Comp. 1997
- [30] Arai T et al., Experiment and Analysis on Elastic Deformation Properties of Graphite and Carbon Materials of HTTR Core Components. Japan Atomic Energy Research Institue 1992; JAERI-M report, 92-085
- [31] Wada Y, Yoshimura S and Yagawa G, Hexahedral Mesh Generation of Nuclear Structures Using Intelligent Local Approach. 15th Int. Conf. on Structural Mechanics in Reactor Technology, Seoul, Aug. 1999
- [32] Miyamura T, Noguchi H, Shioya R, Yoshimura S and Yagawa G, Elastic-plastic analysis of nuclear structures with millions of DOFs using hierarchical domain decomposition method. 15th Int. Conf. on Structural Mechanics in Reactor Technology, Seoul, Aug. 1999
- [33] Yoshimura S, Shoui S, Yagawa G and Akiba H, User Friendly and Parallel Visualization System of Finite Element Solutions of Ten Million DOFs. Proc. of Conf. Comp. Eng. Sci., JSCES 1998; Vol. 3, No. 1: 71-72 (in Japanese)
- [34] Ishii K and Nakazato C, Development of the image based modeling and analysis system. Proceedings of the conference on computational engineering and science, JSCES 1998, Vol. 3, 385-386 (in Japanese)
- [35] Kikuchi N, Hollister S, Chen BC and Sekiguchi M, Possibility of CAE for Generating New Values. Proceedings of the conference on computational engineering and science, JSCES 1999, Vol. 4, 5-8 (in Japanese)
- [36] Miyamura T, Wada Y, Noguchi H, Yoshimura S and Yagawa G, Massively parallel elastic-plastic finite element analysis with voxel mesh. Proceedings of the conference on computational engineering and science, JSCES 1999, Vol. 4, 179-182 (in Japanese)
- [37] <http://adventure.q.t.u-tokyo.ac.jp/>

APPENDIX 1 : ELASTIC ANALYSIS OF 10 M DOFS VOXEL MESH OF PANTHEON IN ROME

Pantheon is an ancient dome placed in Rome, Italy (Photo A1). Here, only the roof of Pantheon is modeled, and elastic analysis under dead load is conducted. Figure A1 shows the cross-section of the roof depicted by using 3D CAD (Micro CADAM Hellix). By rotating this cross-section with the CAD, a solid model shown in Fig A2 can be made. Surface patches are then generated by using KSWAD. Finally a voxel mesh shown in Fig. A3 and Table A1 is generated.

Bottom surfaces of this structure are fixed in the three directions, and dead load is applied. Elastic stress analysis is conducted by using the HDDM. Figure A4 shows convergence of the CG. Figure A5 shows the distribution of equivalent stresses. This figure is depicted by using the parallel visualizer developed in the ADVENTURE project.

Table A1 Voxel mesh of Pantheon with 10 M DOFs

Surrounding voxel space*	400x400x400
Total degrees of freedom	9,591,219
Number of nodes	3,197,073
Number of elements (10-node quadratic tetrahedral element)	2,904,706
Number of subdomains	20,000
Number of Parents	20
DOFs in the interface nodes	4,546,065
Young's modulus	193 GPa
Poisson's ratio	0.275

*The voxel mesh is generated by eliminating unnecessary voxels from this space.



Photo A1 Interior of Pantheon

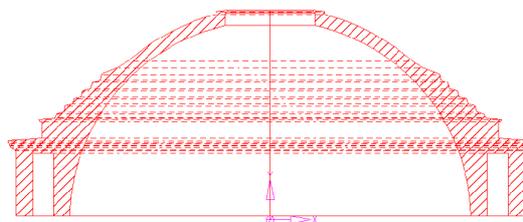


Fig. A1 Cross-section of the roof

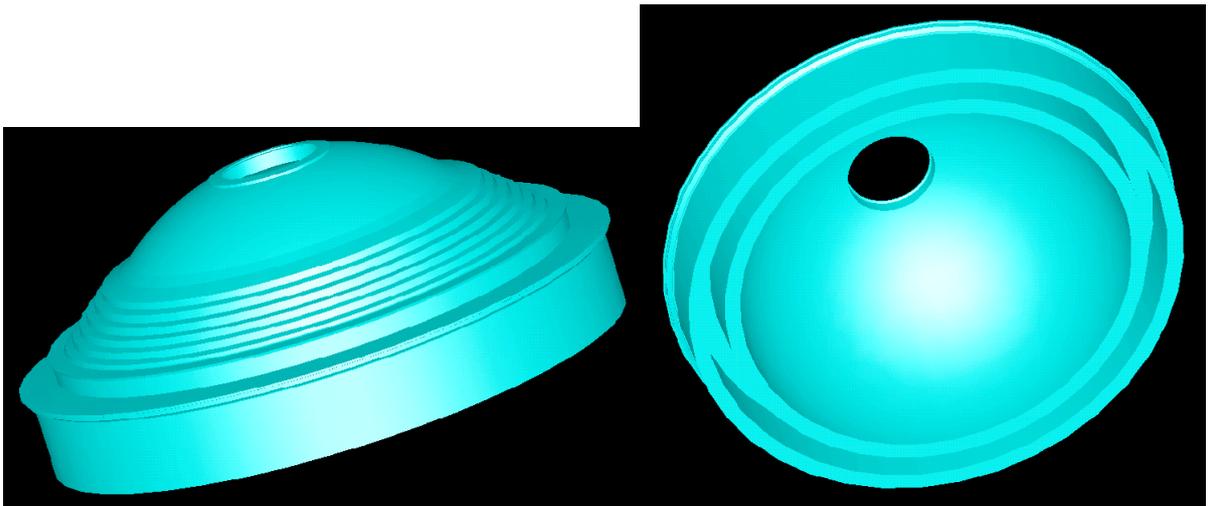


Fig. A2 Solid model made by using Micro CADAM

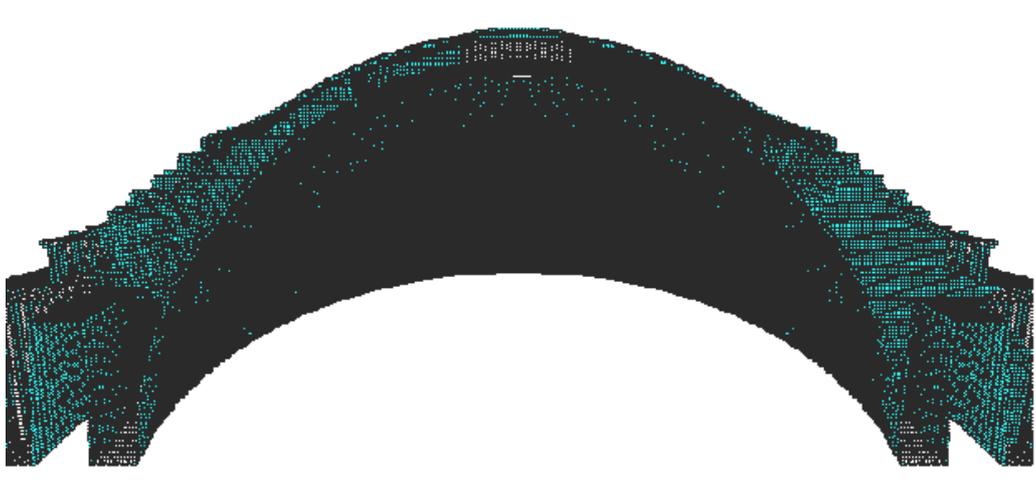


Fig A3 Voxel mesh with 10 M DOFs

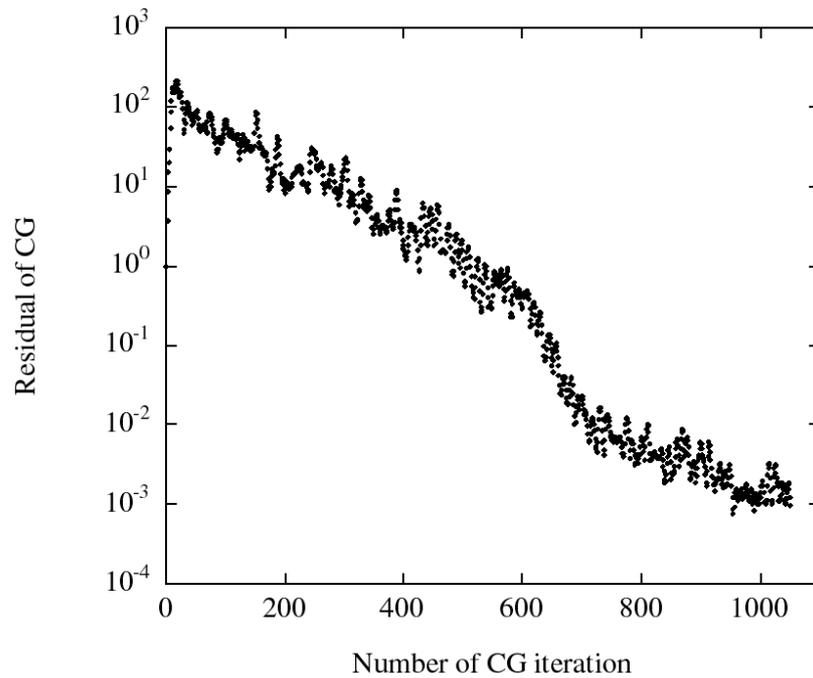


Fig A4 Convergence of CG

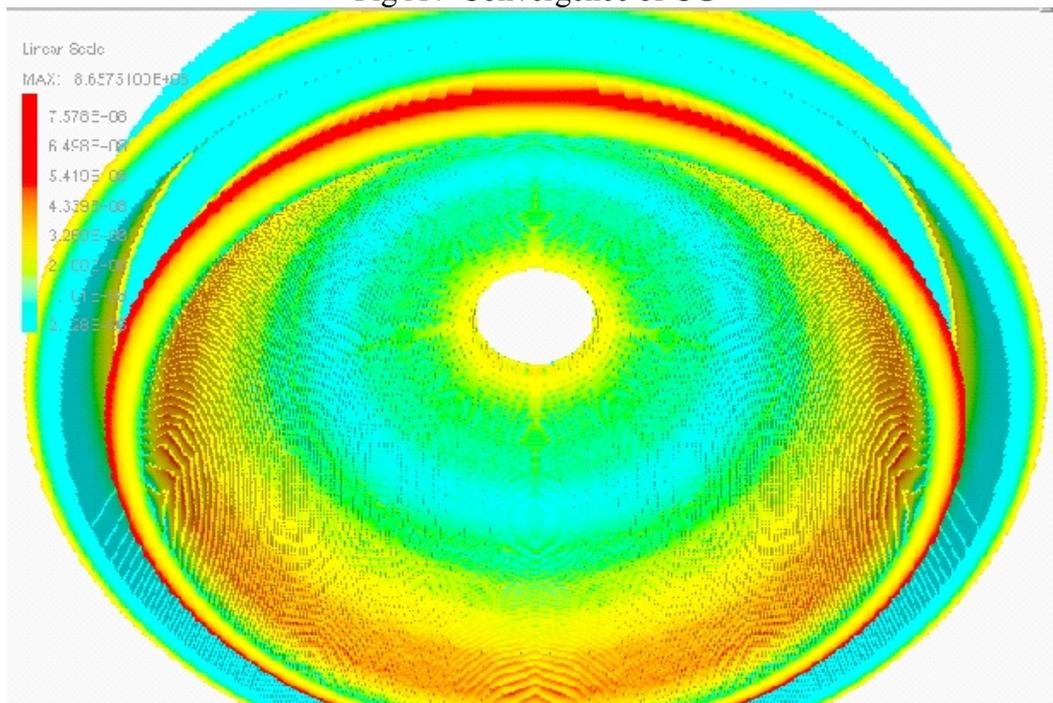


Fig A5 Distribution of equivalent stresses

APPENDIX 2 : ELASTIC ANALYSIS OF A SIMPLE MODEL WITH 47 M DOFS

A simple structure is made from a cube by eliminating a part of it as shown in Fig. A6 (deformed shape). An analysis model of this structure with 47 M DOFs is made by using

tetrahedral elements as shown in Table A2. A surface of this structure is fixed and uniform loads are prescribed on the opposite surface as shown in Fig. A6. Elastic stress analysis is conducted on the analysis model. The computation is performed by using 1024 PEs of SR2201 in the University of Tokyo. Because the maximum elapsed time is limited to 1 hour when 1024 PEs are used, 12 jobs are carried out by restarting the analysis. Figure A6 shows the domain decomposition of the model. Only Parts are shown in this figure, and subdomains in each Part have not been visualized.

Figure A7 shows convergence of the CG. Figure A8 shows the deformation and distribution of equivalent stresses. Figure A9 is a close-up view.

Table A2 A simple 47 M DOFs model

Total degrees of freedom	46,683,273
Number of nodes	15,561,091
Number of elements (10-node quadratic tetrahedral element)	11,481,750
Number of subdomains	44,000
Number of Parents	22
DOFs in the interface nodes	18,173,426
Young's modulus	193 GPa
Poisson's ratio	0.275

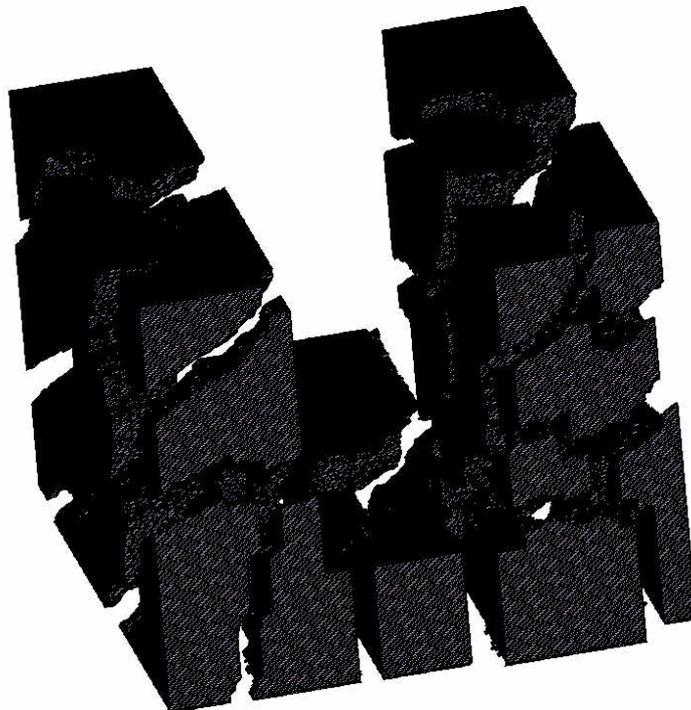


Fig A6 Hierarchical domain decomposition; only parts are shown in this figure.

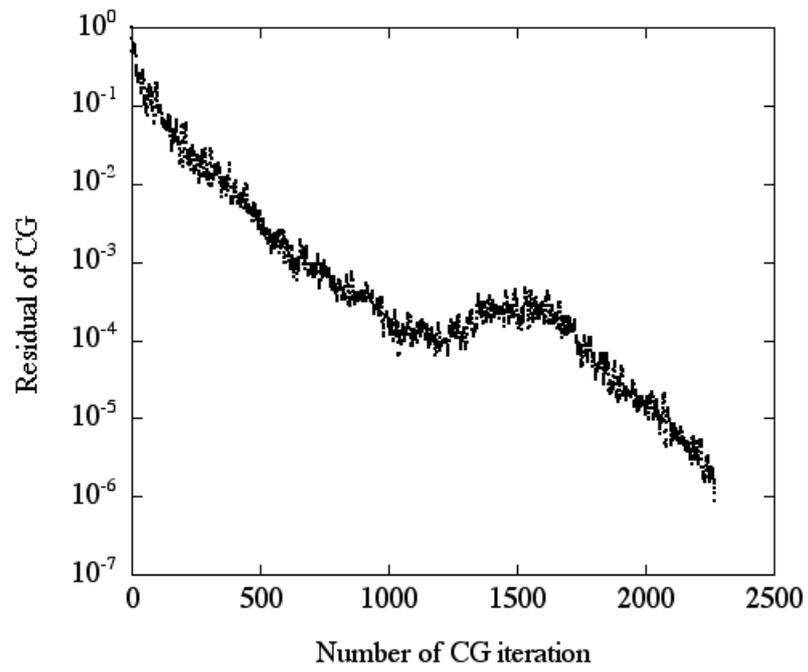


Fig A7 Convergence of CG

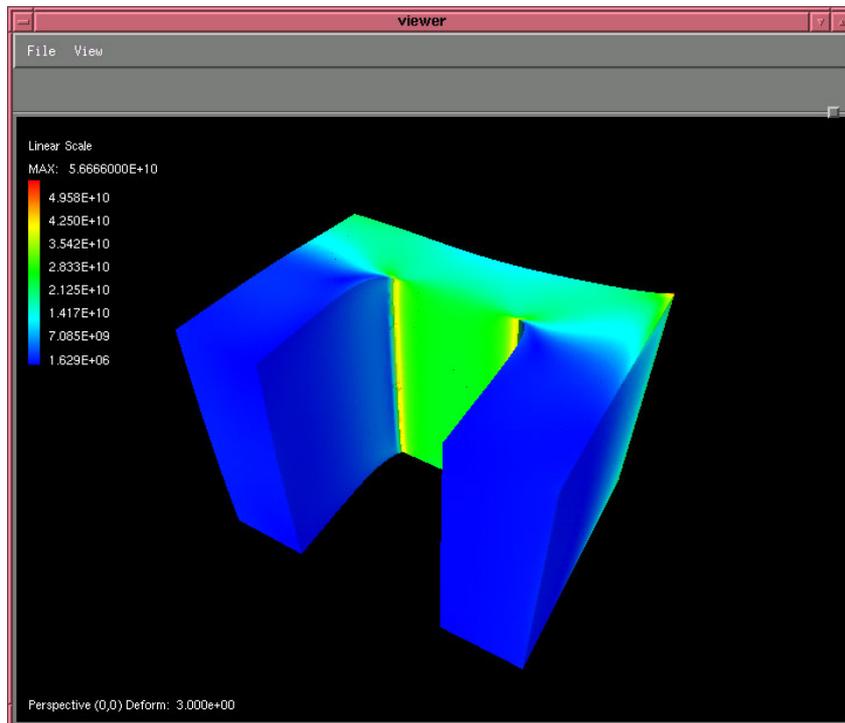
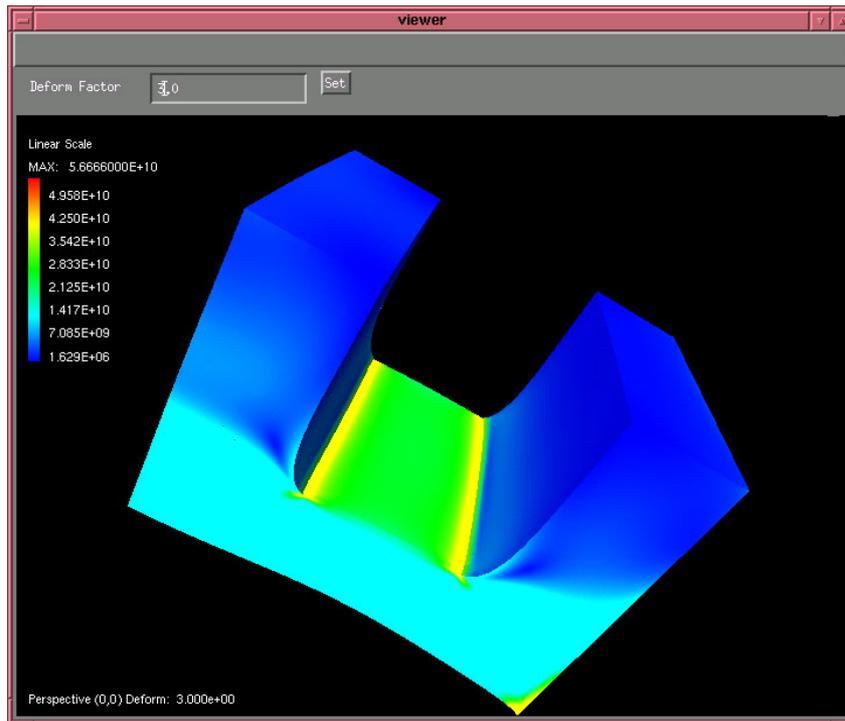


Fig. A8 Deformation and distribution of equivalent stresses

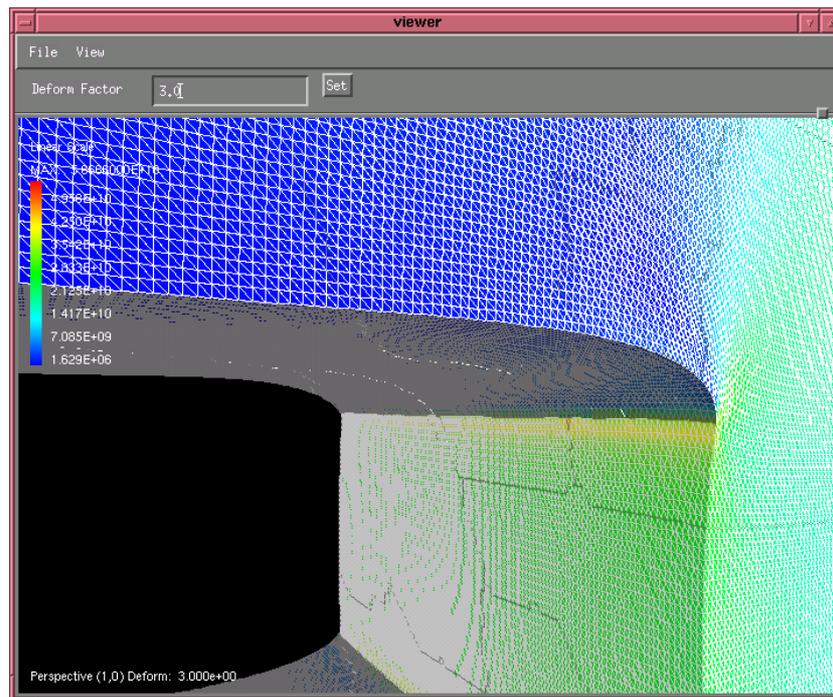


Fig. A9 Deformation and distribution of equivalent stresses (close up)

APPENDIX 3 : ELASTIC ANALYSIS OF 10 M DOFS VOXEL MESH OF NOZZLE

Elastic analysis of the voxel mesh of a nozzle shown in Table A3 is conducted by using 1024 PEs of SR2201. Displacements are prescribed along a part of the edge of the nozzle. Computation time including IO time is 3393 seconds that means the computation is finished within one hour batch job. Figure A10 shows convergence of the CG, and Fig. A11 shows the deformation and distribution of equivalent stresses.

Table A3 Voxel mesh of nozzle

Surrounding voxel space*	280x280x280
Total degrees of freedom	11,992,542
Number of nodes	3,997,514
Number of elements (8-node linear hexahedral element)	3,862,100
Number of subdomains	12,000
Number of Parents	20
DOFs in the interface nodes	4,981,572
Young's modulus	193 GPa
Poisson's ratio	0.275

*The voxel mesh is generated by eliminating unnecessary voxels from this space.

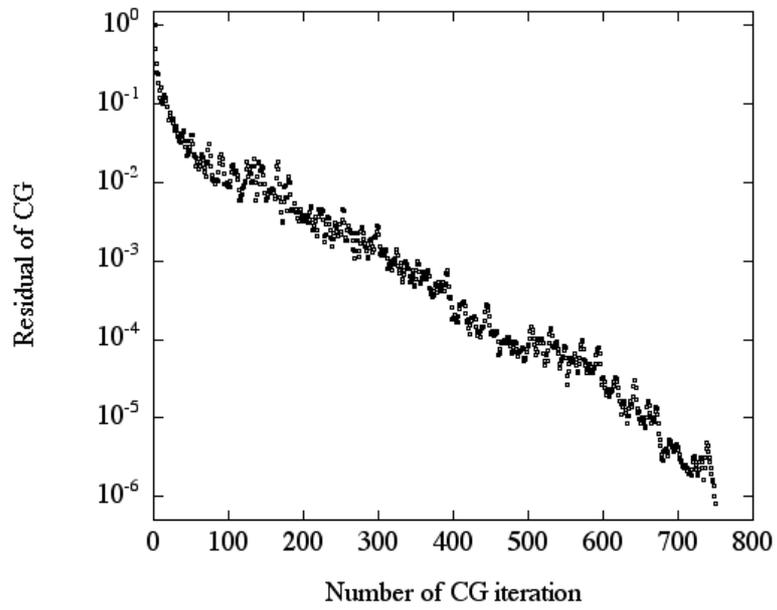


Fig A10 Convergence of CG

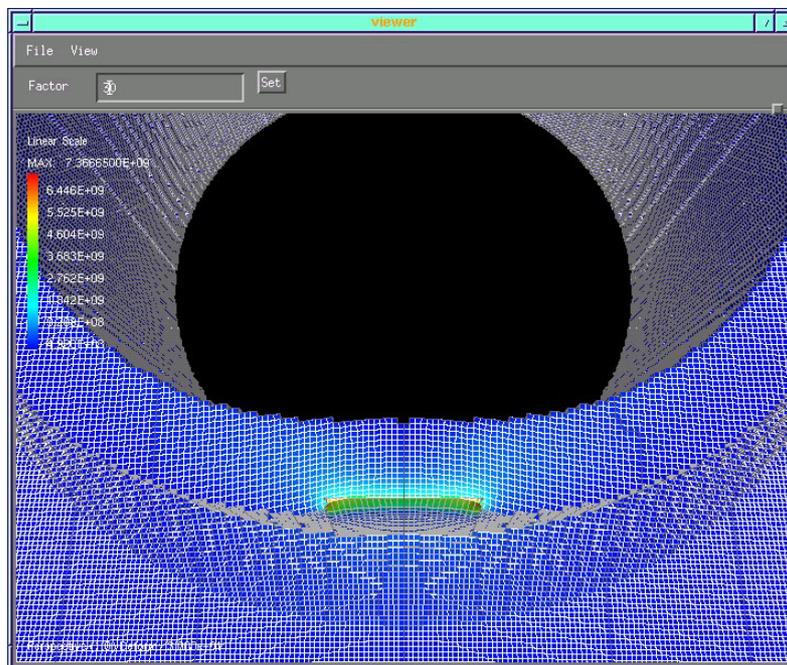


Fig. A11 Deformation and distribution of equivalent stresses