영역분할법에 기반을 둔 병렬 유한요소해석 시스템

Parallel Finite Element Analysis System Based on Domain Decomposition MethodBridges

이 준 성† 塩谷 隆二* 이 은 철** 이 양 창***
Lee, Joon-Seong Shioya, Ryuji Lee, Eun-Chul Lee, Yang-Chang

(논문접수일 : 2008년 9월 23일 ; 심사종료일 : 2009년 1월 15일)

요 지

본 논문에서는 대규모 3차원 구조해석에 필요한 병렬 유한요소해석을 위한 영역분할법의 적용에 대해 묘사하였다. 영역분할법을 사용한 병렬 유한요소법 시스템을 개발하였다. 절점 생성시, 절점들간의 거리가 특정절점에서의 공간함수와 같아지면 절점이 생성되어 진다. 이 절점공간함수는 퍼지지식처리에 의해 조절되어 진다. 기본적인 요소생성은 데로우니 삼각화 기법을 적용하였다. 자동요소생성 시스템을 이용한 영역분할법은 3차원 해석에 큰 도움이 된다. 공간함수와 유사하게 절점들간의 유한요소해석을 위한 병렬 수치 알고리즘으로서 영역분할법을 전체의 해석영역을 완전히 여러 개의 작은 영역으로 겹치지 않게 나누는 공역구배인 반복적 솔버와 결합시켰다. 개발된 시스템의 효용성에 대한 성능을 몇 가지 예를 통해 제시하였다.

핵심용어 : 데로우니삼각화법, 자동요소생성, 영역분할법, 병렬계산, 유한요소해석, 공역구배

Abstract

This paper describes an application of domain decomposition method for parallel finite element analysis which is required to large scale 3D structural analysis. A parallel finite element method system which adopts a domain decomposition method is developed. Node is generated if its distance from existing node points is similar to the node spacing function at the point. The node spacing function is well controlled by the fuzzy knowledge processing. The Delaunay triangulation method is introduced as a basic tool for element generation. Domain decomposition method using automatic mesh generation system holds great benefits for 3D analyses. Aa parallel numerical algorithm for the finite element analyses, domain decomposition method was combined with an iterative solver, i.e. the conjugate gradient(CG) method where a whole analysis domain is fictitiously divided into a number of subdomains without overlapping. Practical performance of the present system are demonstrated through several examples.

Keywords : *delaunay triangulation method, automatic mesh generation, domain decomposition method, parallel computing, finite element analysis, conjugate gradient*

1. Introduction

Loads for pre-processing and post-processing are increasing rapidly in accordance with an increase of scale and complexity of analysis models to be solved. Particularly, the mesh generation process, which influences computational accuracy as well as efficiency and whose fully automation is very difficult in three-dimensional(3D) cases, has become the most critical issue in a whole process of the finite element(FE) analyses. In this respect, various researches(Anglada et al., 1999; Watson, 1991; Lee, 2002: Shephard et

† 책임저자, 정회원・경기대학교 기계시스템공학과 교수
Tel: 031-249-9813 ; Fax: 031-244-6300
E-mail: jslee1@kyonggi.ac.kr
* 일본 Toyo University 계산공학과 교수
** 경기대학교 대학원 석사과정
*** 대림대학

• 이 논문에 대한 토론을 2009년 4월 30일까지 본 학회에 보내주시면 2009년 6월호에 그 결과를 게재하겠습니다.

al., 1991; Nguyen et al., 1991; Berzins, 1999; Taniguchi et al. 1991) have been performed on the development of automatic mesh generation techniques.

In domain decomposition method(DDM), a whole domain to be analyzed is fictitiously divided into a number of subdomains and each subdomain is subdivided into continuous elements without overlapping to be analyzed by FEM. Therefore, for the DDM system, another mesh generator which decomposes mesh into subdomains, is needed in addition to the normal mesh generator for the FEM system

The present authors have developed an automatic FE mesh generation technique, which is based on the fuzzy theory(Zadeh, 1983; Lee, 1995) and computational geometry technique, is incorporated into the system, together with one of commercial solid modelers (Lee, 2004; Choi et al., 2006). In the present study, to support the DDM analysis system which require such special mesh, an automatic mesh generation and domain decomposition system combined with the automatic mesh generation system. Also, the above technique is applied to DDM combined with an iterative solver. The FE analyses of each subdomain is performed in parallel. And the system using DDM algorithms is successfully applied to incremental formulations of 3D structural problem.

In the next version, the present system will be applied to 3D elasto-plastic structural problem of over one million DOF.

2. Domain Decomposition Method with CG Algorithm

In the present domain decomposition method, a whole domain to be solved is first divided into a number of subdomains with out overlapping, and a solution is obtained iteratively in each subdomain considering the force balance and the compatibility condition is satisfied exactly, while the force balance condition is satisfied in a weak manner.

2.1 Fundamental Equations

The present DDM is summarized in the following. To explain its theory, let us consider an elastic problem concerning a domain Ω, as shown in Fig. 1. Here, $\overline{\mathrm{T}}$ is the traction force applied on the boundary Γ_{T}, $\overline{\mathrm{B}}$ the body force applied in the domain Ω, and $\overline{\mathrm{u}}$ the prescribed displacement on the boundary Γ_{u}.

Fundamental equations of this elastic problem are summarized in an infinitesimal displacement mode as follows:

$$\epsilon_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i}) \text{ in } \Omega \tag{1}$$

$$\sigma_{ij} = C_{ijmn}\epsilon_{mn} \text{ in } \Omega \tag{2}$$

$$\sigma_{ij,j} + \overline{B_i} = 0 \text{ in } \Omega \tag{3}$$

$$\sigma_{ij,j}\nu_j - \overline{T_i} = 0 \text{ on } \Gamma_{\mathrm{T}} \tag{4}$$

$$u_i = \overline{u_i} \text{ on } \Gamma_{\mathrm{u}} \tag{5}$$

where i, j take the value 1 to 3, u_i is a displacement vector, ϵ_{ij} a strain tensor, σ_{ij} stress tensor, C_{ijmn} a coefficient tensor of the Hooke's law and ν_j an outer normal vector on the boundary Γ, respectively. $(\ )_{,j}$ denotes the first order derivative with respect to the coordinate x_j.

The above variational from is equivalent to the following minimization problem which finds the displacement function u which is a stationary point of the energy functional :

$$J(v) = \frac{1}{2}\int_{\Omega}\sigma_{ij}\epsilon_{ij}d\Omega - \int_{\Omega}\overline{B_i}v_i d\Omega - \int_{\Gamma}\overline{T_i}v_i d\Gamma \tag{6}$$

As shown in Fig. 2, after dividing domain Ω into N_d subdomains, $(\Omega^{(d)})_{1\le d\le N_d}$ with γ_{pq} being the interface between $\Omega^{(p)}$ and $\Omega^{(q)}$, solving the above problem is equivalent to finding the displacement functions $u^{(d)}$ which are stationary points of the energy functional:

$$\begin{aligned} J'(v^{(1)},\dots,v^{(N_d)}) = &\ J^{(1)}(v^{(1)}) \\ &+ J^{(2)}(v^{(2)}) + \dots + J^{(N_d)}(v^{(N_d)}) \end{aligned} \tag{7}$$

with additional conditions on the interface boundary

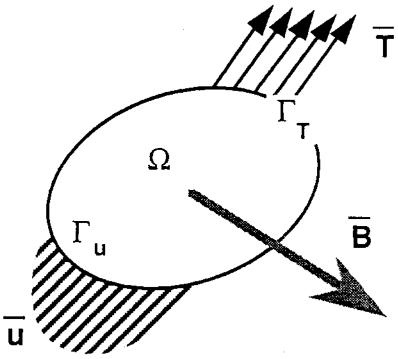

Fig. 1 Analysis domain

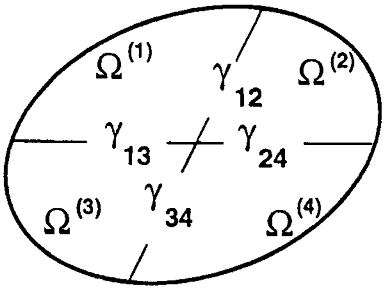

Fig. 2 Analysis domain split into subdomain

γ_{pq} :

$$u^{(p)} = u^{(q)} \text{ on } \gamma_{pq} \tag{8}$$

$$\sigma_{ij}^{(p)}\nu_j^{(p)} + \sigma_{ij}^{(q)}\nu_j^{(q)} = 0 \text{ on } \gamma_{pq} \tag{9}$$

where the superscripts( )$^{(d)}$ designate variable defined in the subdomains $\Omega^{(d)}$.

Depending on the treatment of additional interface boundary conditions of equations (8) and (9), the following two approaches are available : In the first approach, equation (8) is satisfied exactly, while equation (9) is approximately satisfied, and vice versa in the second approach.

Although both the approaches are valid in principle, the first formulation involves fully Neumann tye calculations, which often generates floating domains that to not have enough prescribed displacement to eliminate the local rigid body modes. The second approach is thus thought to be more appropriate.

With the use of a Lagrange multiplier method, solving the equation (7) with the subsidiary condition (9) is equivalent to finding the saddle-point of the Lagrangian functional :

$$\mathcal{L}(v^{(1)},\dots,v^{(N_d)},\mu^{(1)},\dots,\mu^{(N_i)}) = \sum_d^{N_d} J^{(d)}(v^{(d)}) + \sum_{p,q}^{N_d}\int_{\gamma_{pq}} \mu^T C(v^{(p)},v^{(q)})d\gamma \tag{10}$$

where

$$C(v^{(p)},v^{(q)}) = \sigma_{ij}^{(p)}v_j^{(p)} + \sigma_{ij}^{(q)}v_j^{(q)} \tag{11}$$

and N_i is total DOF on interface γ_{pq}. The above problem can be equivalently converted to finding the displacement functions $u^{(d)}$ and the interface Lagrange multipliers $\lambda^{(i)}$ that satisfy :

$$\begin{aligned} &\mathcal{L}(v^{(1)},\dots,v^{(N_d)},\mu^{(1)},\dots,\mu^{(N_i)}) \\ &\quad \le \mathcal{L}(u^{(1)},\dots,u^{(N_d)},\lambda^{(1)},\dots,\lambda^{(N_i)}) \\ &\quad \le \mathcal{L}(v^{(1)},\dots,v^{(N_d)},\lambda^{(1)},\dots,\lambda^{(N_i)}) \end{aligned} \tag{12}$$

for any admissible $(v^{(d)})_{1\le d\le N_d}$ and $(\mu^{(i)})_{1\le i\le N_i}$.

We can solve this saddle-point problem (12) by a saddle-point solver such as Uzawa′s algorithm or its CG variants(Shioya et al., 2003). Let us describe the CG method to solve the equation (12) in next section.

2.2 CG Method for DDM

Defining the positive definite and symmetric operator A:

$$A_\mu^{(i)} = C(\mu^{(p)}(\mu),\mu^{(q)}(\mu)) \tag{13}$$

where

$$u^{(p)}(\mu) = u^{(q)}(\mu) = u^{(i)} \text{ on } \gamma_{pq} \tag{14}$$

the CG algorithm for solving equation (12) is summarized as follows:

Step 0: Initialization

$$\mu^{(i)^0} \text{ : arbitrarily given} \tag{15}$$

$$g^{(i)^0} = A\mu^{(i)^0} \tag{16}$$

$$w^{(i)^0} = g^{(i)^0} \tag{17}$$

The $g^{(i)^0}$ of equation (16) is obtained from the traction forces on γ_{pq} which are calculated by solving equations (1)~(5) in each subdomains with following constraint:

$$u^{(p)} = u^{(q)} = u^{(i)^0} \text{ on } \gamma_{pq} \tag{18}$$

Step 1: Steepest descent

$$\mu^{(i)^{n+1}} = \mu^{(i)^n} - \rho^n \omega^{(i)^n} \tag{19}$$

where

$$\rho^n = \frac{\sum_i^{N_i} g^{(i)^n} g^{(i)^n}}{\sum_i^{N_i} \omega^{(i)^n} A\omega^{(i)^n}} \tag{20}$$

Step 2: Calculation of the new descent direc-tion

$$g^{(i)n+1} = g^{(i)n} - \rho^n A\omega^{(i)^n} \tag{21}$$

$$\omega^{(i)^{n+1}} = g^{(i)^{n+1}} - \kappa^n \omega^{(i)^n} \tag{22}$$

where

$$\kappa = \frac{\sum_i^{N_i} g^{(i)^{n+1}} g^{(i)^{n+1}}}{\sum_i^{N_i} g^{(i)^n} g^{(i)^n}} \tag{23}$$

The $A\omega^{(i)^n}$ of equation (20) and (21) is obtained from the traction forces on γ_{pq} which are calculated by solving the following equations:

$$\overset{(d)}{\sigma}_{ij,j} = 0 \text{ in } \Omega(\mathrm{d}) \tag{24}$$

$$\sigma_{ij}^{(d)} \nu_j^{(d)} = 0 \text{ on } \Gamma_{\mathrm{T}}^{(\mathrm{d})} \tag{25}$$

$$u_i^{(d)} = 0 \text{ on } \Gamma_{\mathrm{u}}^{(\mathrm{d})} \tag{26}$$

$$u_i^{(d)} = \omega^n \text{ on } \gamma_{pq} \tag{27}$$

Step 3: Judgment of convergence

If $\mu^{(i)^n}$ has not converged yet, return to step 1 by setting n to be $n+1$. Here the convergence criterion is defined as:

$$\frac{\max\left|g^{(i)^n}\right|}{\max\left|g^{(i)^0}\right|} < \mathrm{Err} \tag{28}$$

in which the maximum component of force imbalance along the interface boundary, i.e. residual value, is monitored.

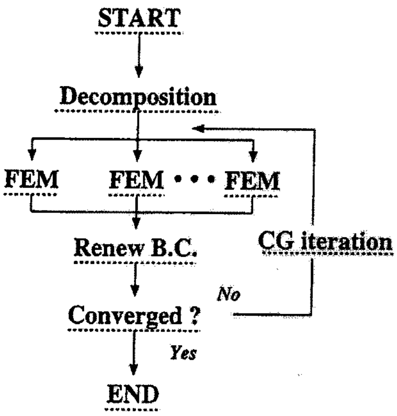

Fig. 3 Flow chart of DDM

The flow chart of the present DDM algorithm is illustrated in Fig. 3. It should be noted here that the FE analysis of each subdomain can be performed without any data communication among subdomains. In other words, the FE calculations of subdomains can be performed independently, i.e. in parallel, once the amounts of displacements on the boundaries between subdomains are known. Since the workload for each FE calculation is much larger than those of other tasks including data transfer and modification of boundary values, the so-called overhead due to communication is estimated to be very small. In addition, owing to the decomposition of a large scale FE system into a number of smaller sub-systems, only small computation storage is needed for each FE calculation. Also the calculation time depends much on the number of CG iteration. Therefore, preconditioning for CG is efficient to speed up convergence(Shioya et al., 2003).

3. Mesh Generation for DDM

In DDM system, another mesh generator which decomposes mesh into subdomains, is needed in addition to the normal mesh generator for the FEM system. Thus, to support the DDM analysis system which require such special mesh, an automatic mesh generation and domain decomposition system combined with the automatic mesh generation system for FEM system(Lee, 2004), is developed

3.1 Mesh Generation for FEM

Performances of automatic mesh generation methods based on node generation algorithms depend on how to control node spacing functions or node density distributions and how to generate nodes. The basic concept of the present mesh generation algorithm is originated from the imitation of mesh generation processes by human experts on FE analyses.

The present system stores several local node patterns such as the pattern suitable to well capture stress concentration, the pattern to subdivide a finite domain uniformly, and the pattern to subdivide a whole domain uniformly. A user selects some of those local node patterns, and designates where to locate them. Then a global distribution of the node density over the whole analysis domain is automatically calculated through their superposition using fuzzy knowledge processing(Zadeh, 1983; Lee, 1995; Lee, 2004).

Node generation is one of time consuming processes in automatic mesh generation. Here, the bucketing method(Asano, 1985, Lee, 1995) is adopted to generate nodes which satisfy the distribution of node density over the whole analysis domain. The base node pattern and several special ones are placed in the domain, and all the node patterns are smoothly connected.

The Delaunay triangulation method(Watson, 1991; Taniguchi et al., 1991) is utilized to generate tetrahedral elements from numerous nodes given in a geometry.

3.2 Domain Decomposer for DDM

In DDM, a whole domain to be analyzed is fictitiously divided into a number of subdomains without overlapping and each subdomain is subdivided into continuous elements without overlapping to be analyzed by FEM. To generate such a mesh, it is generate elements in a whole domain first and then divide the whole domain into subdomains considering the boundary of elements as shown in Fig. 4.

3.3 Decomposition Algorithm

The automation of the process of DDM is the key in the present domain subdivision system. The flow of this process is as follows as shown in Fig. 5.

(1) A brick which involves the whole domain is set.

(2) The domain is divided into two subdomains against the longest side of the brick, if the domain has the DOF more than the max-imum value of DOF chosen.

(3) If the divided two subdomains also have the DOF more than the maximum value, the procedures (2) and (3) are repeated recursively.

(4) If one of the two divided subdomains has the DOF less than the half of the chosen max-imum value of DOF, the procedure of (2) is re-executed with the slightly shifted cutting plane.

(5) If both of the two subdomains have the DOF less

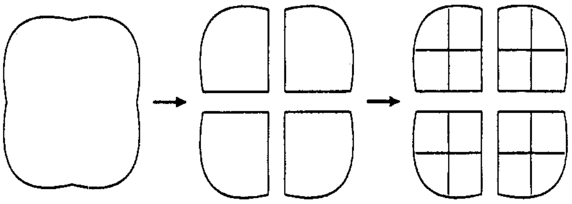

Fig. 4 Subdomain element divide approach

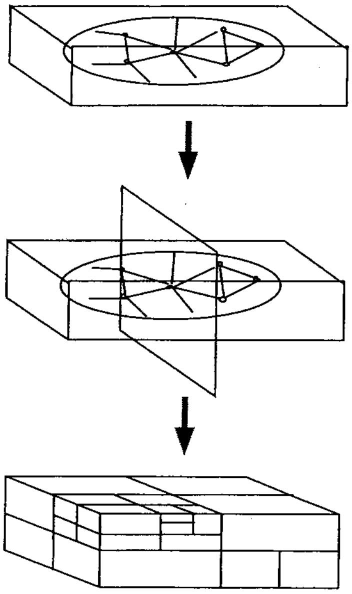

Fig. 5 Flow of domain subdivision

than the maximum and more than the half, the subdivision of the part is complete.

With this binary-tree-like algorithm, the whole domain can be divided into subdomains which have nearly and equal DOF, in case that node density of nodes changes widely.

Each FE analysis of subdomain is performed concurrently in the present DDM system.

4. Examples and Discussions

4.1 Quality of DDM Mesh

In order to evaluate the mesh generated by the present system, part of a pressure vessel with a nozzle model is defined and divided into subdomains and parts.

The geometry of this model is defined using solid modeler and node and elements are generated as shown in Fig. 6, which consists of 8,426 nodes and 38,853 tetrahedral elements. Around the lower part of the joint of the nozzle where stress concentration is supposed to occur, the node density controlled well.

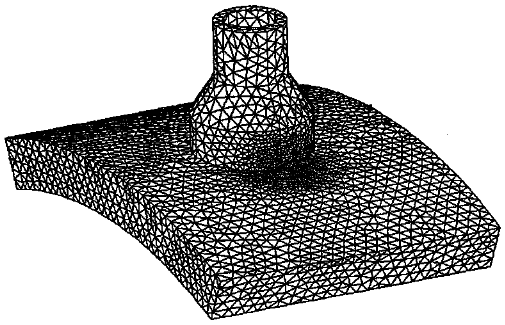

Fig. 6 Mesh of nozzle model

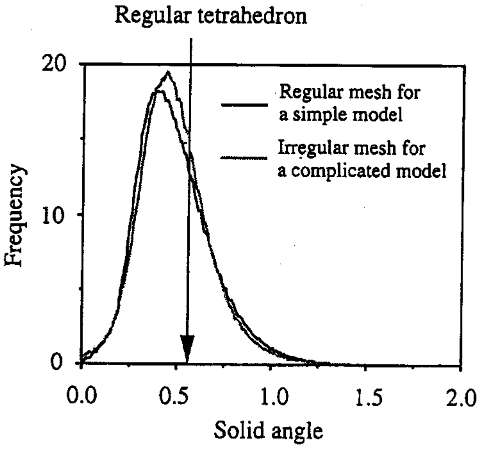

Fig. 7 Distribution of values of solid angle in the FEM mesh

To check the quality of this mesh, all solid angle of each tetrahedral element are measured and its distribution is plotted in Fig. 7. For comparison, the distribution of a regular mesh of the same size for a simple model is also plotted. Since the solid angle of the regular tetrahedron is equivalent to 0.51, it is demonstrated that this mesh generator system can control the node density with a good quality of mesh.

Next, the present decomposition system is applied to this mesh, that is, it is divided into subdomains.

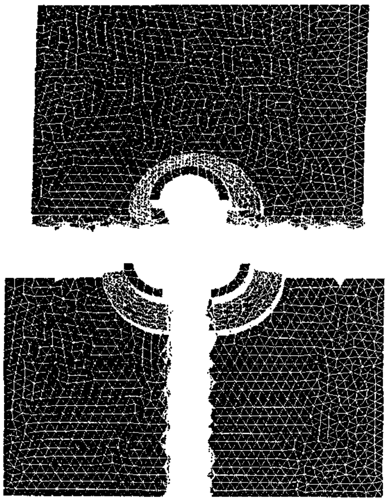

Fig. 8 Second dividing step 3 subdomains

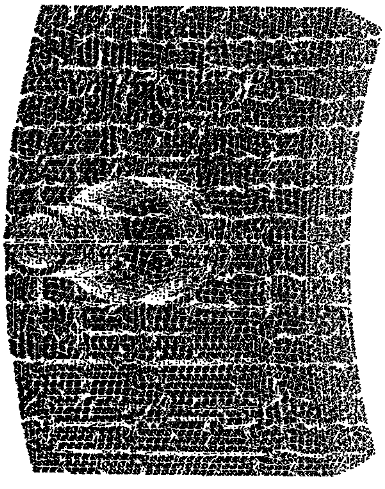

Fig. 9 Final divided model

On the first step of decomposition, this model is divided into two subdomains. And in following steps, this model is divided into three subdomains as shown in Fig. 8. Iterating such dividing procedure, finally 480 subdomains are obtained as shown in Fig. 9.

4.2 Comparison with Sequential FEM

In order to evaluate the results performed by present parallel FEM system, its physical values, displacement and stress calculated by the present system, which is abbreviated as 'DDM', were compared with those by the uusual sequential FEM system, abbreviated as 'FEM'.

'DDM' and 'FEM' are applied to the FE analysis of a cubic structure subject to uniform tension. The structure was modeled with 10,368 4-node tetrahedral elements, and the total DOF was 6,591. In the case of analysis of 'DDM', the whole structure was divided into 27 subdomains and each subdomain consists of 384 elements as shown in Fig. 10.

Table 1 shows the maximum and mean value of the errors of displacement and stress respectively given by 'DDM' relative to 'FEM'. Here, the displacement and stress, denoted $\overline{\delta}$ as and $\overline{\sigma}$, respectively, are defined as follows:

$$\overline{\delta}=\sqrt{\delta_x^2+\delta_y^2+\delta_z^2} \tag{29}$$

$$\overline{\sigma}=\sqrt{\frac{1}{2}[(\sigma_y-\sigma_z)^2+(\sigma_z-\sigma_x)^2+(\sigma_x-\sigma_y)^2]} \\ +\sqrt{3\left(\tau_{yz}^2+\tau_{zx}^2+\tau_{xy}^2\right)^2} \tag{30}$$

where $\delta_x, \delta_y, \delta_z$ are the displacement values of axis *x*, *y*, *z*, $\sigma_x, \sigma_y, \sigma_z$ are the stresses of axis *x*, *y*, *z* and $\tau_{xy}, \tau_{yz}, \tau_{zx}$ are the shear stresses.

Figs. 11 and 12 show the variations of the residual value and the error values plotted with respect to the iteration number of the CG algorithm, respectively. Now, the number of CG iteration of 'DDM' is defined when the residual value defined in equation (28) decreases by the order of 10^{-3} of the first step. It is shown here that the error values decrease rapidly with the number of iterations.

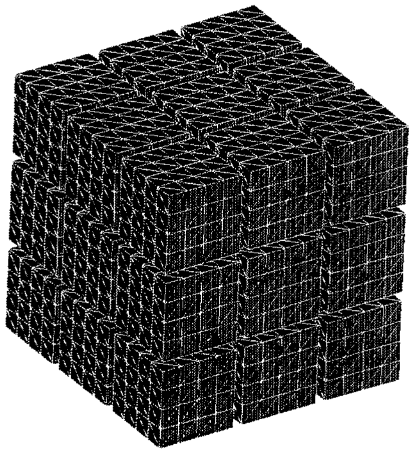

Fig. 10 Cubic model with decomposition

Table 1 Error of DDM for FEM

	Displacement	Stress
Max. Error	1.03 %	1.51 %
Mean Error	0.30 %	0.42%

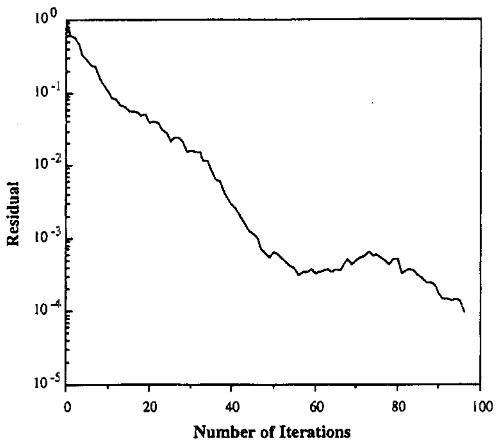

Fig. 11 Residual vs number of iterations

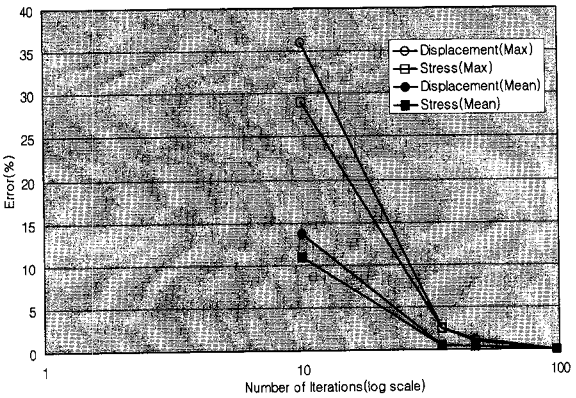

Fig. 12 Error vs number of iterations

To check the convergence of physical value such as displacement, the cubic structure identical to the above was analyzed again. Now the cubic structure was modeled with 74,088 isoparametric 8-noded elements and the whole domain was divided into 216 subdomains, each consisting of 343 elements. The total DOF of this model was 238,521 and this structure was subjected to uniform tension.

Fig. 13 shows the variations of the residual value against CG iterations and Fig. 14 shows the variations of the displacement of some positions shown in Fig. 15 against CG iterations. Comparing Fig. 13 and Fig. 14, it is estimated that about 40 CG iterations may be sufficient to attain converged solutions of displacements in a practical sense.

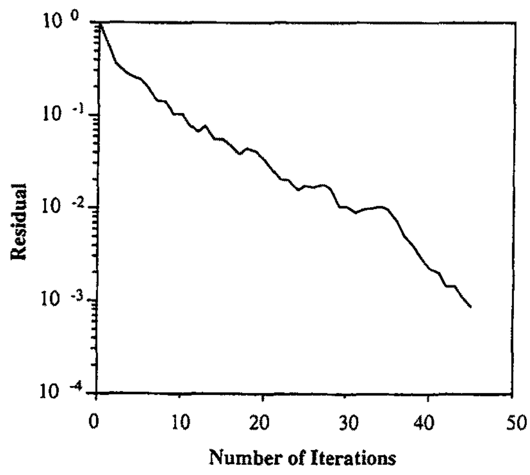

Fig. 13 Residual vs number of iterations

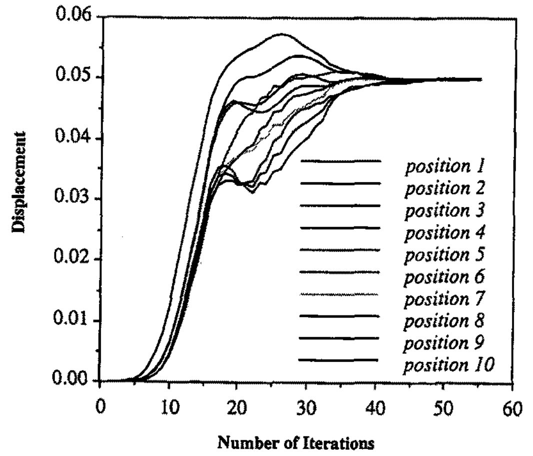

Fig. 14 Displacement vs number of iterations

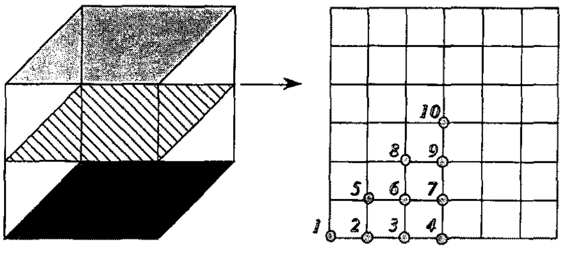

Fig. 15 Monitored positions for displacement

4.3 Division Number

As described in section 3.2 and 3.3, with this system, we can divide the model into any number of subdomains. Then how to decide the division number is important for performance.

For simplicity, a cubic model shown in Fig. 10 is used here to estimate the computational time. Each axis of the model is now divided by N with isoparametric 8-noded elements, therefore, the domain of the model is divided into N^3 elements and the total number of nodes is equivalent to $(N+1)^3$. Each axis is then divided into subdomains by d, and therefore, the total number of subdomains is equivalent to d^3 and each subdomain consists of $(N/d)^3$ elements and $(N/\mathrm{d}+1)^3$ nodes, DOF of which, $D_{\mathrm{subdomain}}$, is $(N/d+1)^3 \times 3$. The total number of interface DOF, $D_{\mathrm{interface}}$, on the other hand amounts to approximately $(N^2 \times d \times 3)$.

Now the computational time of the FEM analysis of each subdomain, T_{domain} and the total number of CG iterations $N_{\mathrm{iteration}}$ can be evaluated as follows:

$$T_{domain} = f_c(D_{domain}) = f_c((\frac{N}{2}+1)^3 \times 3) \tag{31}$$

$$N_{iteration} = f_i(D_{interface}) = f_i(N^2 \times d \times 3) \tag{32}$$

where f_c and f_i are function of DOF of each subdomain and DOF of interface, respectively.

Using f_c and f_i, ignoring communication, total computational time T_{total} can be described as follows:

$$\begin{aligned} T_{total} &= T_{domain} \times d^3 \times N_{iteration} \\ &= f_c(D_{domain}) \times d^3 \times f_i(D_{interface}) \end{aligned} \tag{33}$$

Unfortunately, those functions are unknown but it can be only estimated in a rough fashion. For f_c, with a direct solver, the main part of FEM analysis is the factorization of matrix part and forward and backward elimination part. The order of operation times of these calculation are $O(N^3)$ and $O(N^2)$, respectively, where N is DOF of problem. Here, for simplicity, f_c is defined as a quadratic function of $D_{\text{subdomain}}$.

For f_i, from the CG algorithm for matrix solution, it is known that the number of iterations is linear to the square root of DOF and its coefficient depends on the condition number of the matrix.

Using these assumptions, f_c and f_i are described as follows:

$$f_c(x) = ax^2 + bx + c \tag{34}$$

$$f_i(x) = dx^{1/2} + e \tag{35}$$

where a, b, c, d and e are all coefficients.

To find out these coefficients, the total computational time can be estimated. To determine them, some numerical experimentations are performed here. For some sets of N and d, analyses of the DDM are performed and f_c and f_i are evaluated.

The results of these numerical experimentations which are performed by only on workstation, Sun SS(50MHz), are shown in Table 2. In this assumption, since communication time is avoided, one processor is used for all execution of the DDM without communication.

First, using the results of f_c with $D_{\text{subdomain}}$, f_c is estimated as follows:

$$f_c(x) = ax^2 + bx + c$$

$$a = 3.72 \times 10^{-6}$$

$$b = 1.50 \times 10^{-4}$$

$$c = 1.92 \times 10^{-3}$$

Table 2 Computational time and CG iterations

N	d	$D_{subdomain}$	$D_{interface}$	f_i	f_c	T_{total}
8	2	375	583	0.58	20	92.8
8	4	81	1,359	0.04	25	66.8
12	3	375	2,406	0.58	26	407.2
12	4	102	3,315	0.17	29	329.7
24	4	1,092	14,367	4.10	43	11386.4
24	6	375	21,975	0.58	47	5888.2
24	8	192	28,175	0.17	53	4613.1
24	12	81	36,927	0.04	58	4009.0

Next, f_i was estimated for each N using two of d, i.e. in case of (N=24), the results of (d=4) and (d=6) are used, as follows:

$$f_i(x) = dx^{1/2} + e$$

$$d = 8.5,\ e = 7.9\ (N = 8)$$

$$d = 11.2,\ e = 6.6\ (N = 12)$$

$$d = 9.0,\ e = 25.2\ (N = 24)$$

Using these results, the total computation time T_{total} were estimated and plotted against division number d in Fig. 16 and Fig. 17 As shown in these figures, there exists one division number d which minimize the total computation time.

For the case of (N=24), the results with (d=8) and (d=12) were not used to estimate T_{total}, but they match well with the estimated results. This means, to perform a few numerical experimentations for the model, it can be estimated the best division number d for the model.

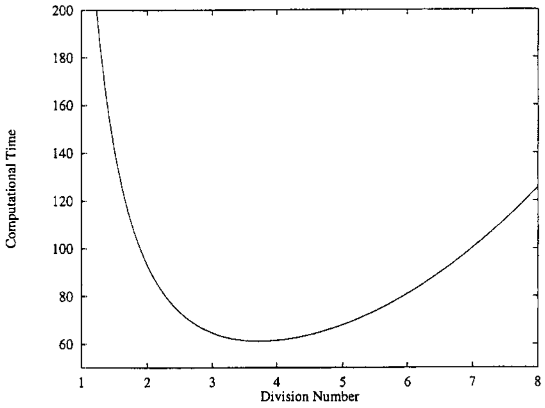

Fig. 16 Computational time vs division number(N=8)

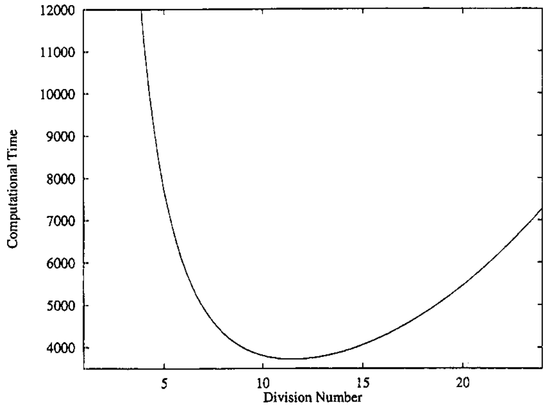

Fig. 17 Computational time vs division number (N=24)

5. Conclusions

The parallel finite element system based on the DDM and the automatic mesh generator for DDM system, were successfully developed in the present study.

The key features of the present algorithm are an easy control of complex 3D node density distribution and fast node and element generation owing to some computational geometry techniques. The effective of the present system is demonstrated through several mesh generations for 3D complex structures such as pressure vessel with nozzles model.

In the next version, the present system will be applied to 3D elasto-plastic structural problem of over one million DOF.

References

Asano, T. (1985) Practical use of bucketing techinques in computational geometry, *Computational Geometry, North-Halland*, pp.153~195.

Anglada, M.V., Garcia, N.P., Crosa, P.B. (1999) Directional Adaptive Surface Triangulation, *Computer Aided Geometric Design*, 16, pp.107~126.

Berzins, M. (1999) Mesh Quality: A Function of Geometry, Error Estimates or Both, *Engineering with Computers*, 15, pp. 236~247.

Chio, J.B., Park, Y.J., Ko, H.O., Chang, Y.S., Kim, Y.J., Lee, J.S. (2006) Parallel Process System and Its Application to Flat Display Modules Impact Analysis, *Proceedings of the 7th World Congress on Computational Mechanics*, pp.16~22.

Nguyen, D.T., Al-Nasra, M. (1981) An Algorithm for Domain Decomposition in Finite Element Analysis, *Computer and Structures*, 39(3), pp.277~289.

Lee, J.S. (1995) Automated CAE System for Three-Dimensional Complex Geometry, *Doctoral Thesis, The University of Tokyo*.

Lee, J.S. (2002) Development of the Fuzzy-Based System for Stress Intensity Factor Analysis, *Int. J. of Fuzzy Logic and Intelligent Systems*, 12(3), pp.255~260.

Lee, J.S. (2004) Development of High-Performance FEM Modeling System Based on Fuzzy Knowledge Processing, *Int. J. of Fuzzy Logic and Intelligent Systems*, 4(2), pp.193~198.

Shioya.R., Yagawa. G. (2003) Finite Elements on massively Parallel Computer with Domain Decomposition, *Computing Systems in Engineering* 4(4), pp.495~503.

Taniguchi, T., Ohta, C. (1991) Delaunay-Based Grid Generation for 3D Body with Complex Boundary Geometry, *Grid Generation Conference*, pp.55~60.

Watson, D.F. (1991) Computing the N-Dimensional Delaunay Tessellation with Application to Voronoi Polytopes, *The Computer Journal*, 24(2), pp.167~172.

Zadeh, L.A. (1983) Fuzzy Algorithms, *Information and Control, Cybernetics*, SMC-3, 12, pp.28~44.