AIAA JOURNAL
Vol. 32, No. 5, May 1994

Parallel Computing Using Semianalytical Finite Element Method

Jeong Rae Kim, Wie Dae Kim,* and Seung Jo Kim†
Seoul National University, Seoul 151-742, Republic of Korea

In this paper, a parallel computing technique using a semianalytical finite element method and its implementation method are studied, and the speedup in the elapsed computing time is discussed. The structural analysis of an axisymmetric body with nonaxisymmetric boundary conditions is considered by using a semianalytical finite element method. The displacements and the boundary conditions are expressed by a Fourier series. By the orthogonality properties of the Fourier series, the equilibrium equations are fully decoupled for each mode. Therefore, these decoupled equations can be well adapted to the individual processors of a parallel computer. Two example problems are considered for the verification of the developed code and for the application of the parallel processing to practical problems. The computing time by parallel processing is compared with that by sequential processing, and a large amount of improvement in the elapsed computing time is achieved by parallel processing. From the computational experiments, a speedup of 4.6 times in elapsed time is observed with six processors of Alliant FX2812 at Seoul National University.

Introduction

THE analysis of an axisymmetric structure with nonaxisymmetric boundary conditions cannot be performed by conventional techniques of axisymmetric analysis, which means that the analysis should be done by tedious and time-consuming three- dimensional computations. By taking advantage of the axisymmetry of the structures, the displacements and boundary conditions are expressed by a Fourier series. Then the dimension of the structures can be reduced into two dimensions, and the nonaxisymmetric boundary conditions can be fully considered by using a semianalytical finite element method.[1]

The semianalytical finite element method was first studied by Wilson,[2] and he investigated the infinite cylinder problem and the thermal analysis of a rocket nozzle. Pardon et al.[3] studied the stress analysis of an axisymmetric body with orthotropic properties. Spilker and Daugirda[4] applied the semianalytical method to the ring analysis. This method was applied in the analysis of an elastomeric composite structure.[5] The nonlinear semianalytical finite element analysis was achieved by Sedaghat and Herrmann.[6] Parallel computing and its algorithms were studied by Farhat and Wilson,[7] Farhat et al.,[8] Farhat and Roux,[9] and El-Sayed and Hsiung.[10] A review of parallel processing in finite element analysis was given by Noor.[11] There are, however, few research works concerned with parallel processing by using the semianalytical method.

In this paper, we focus on two aspects: 1) the application of the semianalytical finite element method to parallel processing and 2) the investigation of the computing time efficiency by parallel processing. By expressing the displacements and the boundary conditions by a Fourier series, the equilibrium equations are fully decoupled for each mode, and each mode equation is assigned to an individual processor of a parallel computer. The computing time of a sequential processing is compared with that of a parallel processing by changing the number of processors. The first example is an axisymmetric beam bending problem with nonsymmetrical end load, and the solution is compared with the three-dimensional results by ABAQUS, a general purpose code, for verification of the developed program. The second example is a pressure vessel problem with nonaxisymmetric side load. The maximum number of processors used is six, and the computing time efficiency is represented according to the number of processors and to the bandwidth of the finite element model.

In the next section, the semianalytical finite element method is briefly explained to understand the application of this method to parallel computation.

Received July 29, 1993; revision received Dec. 3, 1993; accepted for publication Dec. 6, 1993.

*Research Assistant, Department of Aerospace Engineering.
†Postdoctoral Researcher, Department of Aerospace Engineering. Associate Member AIAA.
‡Associate Professor, Department of Aerospace Engineering. Member AIAA.

Semianalytical Finite Element Method

The advantage of the semianalytical technique lies in the ability to express the behavior of all quantities in terms of a Fourier series in the coordinate direction in which geometry and material properties do not vary. In the case of axisymmetric structures, all quantities are expressed in a Fourier series in terms of the circumferential θ coordinate (see Fig. 1). Then, the displacements $\boldsymbol{u}$ are written in the form[4]

$$\boldsymbol{u} = \sum_l \left(\boldsymbol{C}_l^s \boldsymbol{u}_l^s + \boldsymbol{C}_l^a \boldsymbol{u}_l^a \right) \tag{1}$$

where

$$\boldsymbol{C}_l^s = \begin{bmatrix} \cos l\theta & 0 & 0 \\ 0 & \cos l\theta & 0 \\ 0 & 0 & \sin l\theta \end{bmatrix} \tag{2}$$

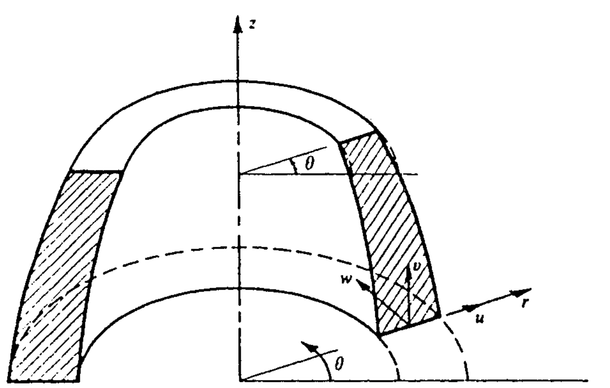

Fig. 1 Coordinates and displacements in an axisymmetric body.[1]

$$C_l^a = \begin{bmatrix} \sin l\theta & 0 & 0 \\ 0 & \sin l\theta & 0 \\ 0 & 0 & -\cos l\theta \end{bmatrix} \tag{3}$$

In Eq. (1), the summation extends over the total number of harmonics, and $(\ )_l^s$ and $(\ )_l^a$ refer to the symmetric and antisymmetric contribution of ( ) to the lth harmonic. The corresponding displacement and load components are shown in Fig. 2. The displacements u_l^s and u_l^a are functions only of r and z and are interpolated in an element in terms of nodal displacements a_l^s and a_l^a in the form

$$\begin{aligned} u_l^s &= N a_l^s \\ u_l^a &= N a_l^a \end{aligned} \tag{4}$$

where N is the interpolation function matrix that satisfies the C^0 continuity family of shape functions.

By the application of the minimum principle of the potential energy and the orthogonalities of $\sin l\theta$ and $\cos l\theta$, the following decoupled equations for the symmetric and the antisymmetric parts are obtained:

$$\begin{aligned} K_l^s a_l^s &= Q_l^s \\ K_l^a a_l^a &= Q_l^a \end{aligned} \tag{5}$$

For the detailed formulation of the semianalytical finite element method, Refs. 1–11 can be consulted.

The element stiffness matrix and the force vectors are assembled in each mode, and each mode's equations are decoupled by the characteristics of independency in each mode. Therefore, it is possible to apply the semianalytical method to parallel computation. These decoupled equations are assigned to individual processors of a parallel computer. It is noted that the efficiency in computing time is achieved in two stages: first by the semianalytical method and second by parallel computation. The method for parallel processing and program modification is explained in the next sections.

Method for Parallel Computing

In the conventional semianalytical finite element method, the solving procedure of the equations is sequential. That is, after the equation of one mode is solved, the other is solved sequentially. But, since the equations are decoupled in each mode, the semi analytical finite element method can be well adapted to parallel processing. In other words, each mode can be assigned to the individual processors of a parallel computer[12,13] by modifying the processing procedure, and parallel computing becomes possible.

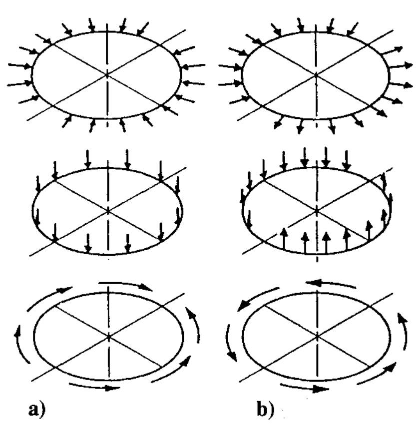

Fig. 2 a) Symmetric and b) antisymmetric load and displacement components in an axisymmetric body.[1]

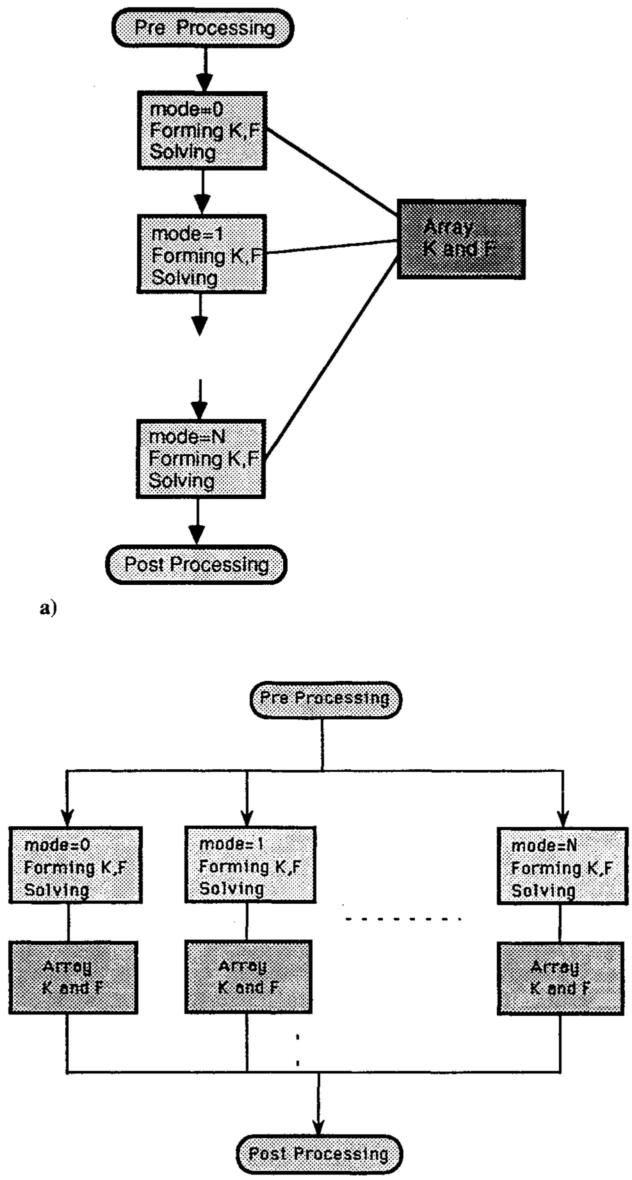

Fig. 3 Comparison of a) sequential processing and b) parallel processing.

This algorithm is especially appropriate for the multiple instructions and multiple data streams (MIMD) parallel computing environments.

The computational experiments by parallel computing are performed on an Alliant FX2812, which has 12 processors of the Intel i860 running on 40 MHz. The programming language is FX/Fortran[14] developed for parallel processing, which has some special parallel command on concurrent or vector processing. The number of processors can be changed in the executing program, and the maximum number of processors used in these experiments is six. In assigning each mode equation to each processor, the program is modified to have independent arrays in each mode, and the FX/Fortran command is used for the program to be executed in the parallel mode. The processing procedures in sequential processing and in parallel processing are shown in Figs. 3a and 3b, respectively. In sequential processing, each mode accesses the same arrays, but in parallel processing, each mode has independent arrays.

The original program for sequential processing consists of three parts: preprocessing, main processing, and postprocessing. Among them, main processing contains stiffness and force matrix calculation and equation solving routines. The Gauss elimination method is used for equation solving. The routine, which we call Solver,

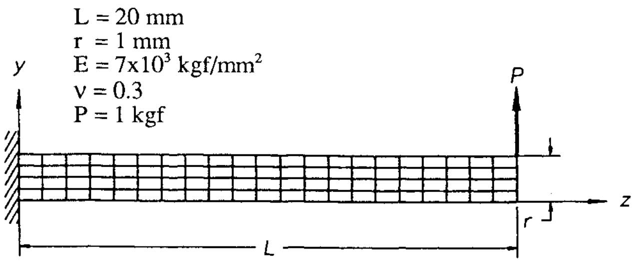

Fig. 4 Semianalytical finite element model of beam bending.

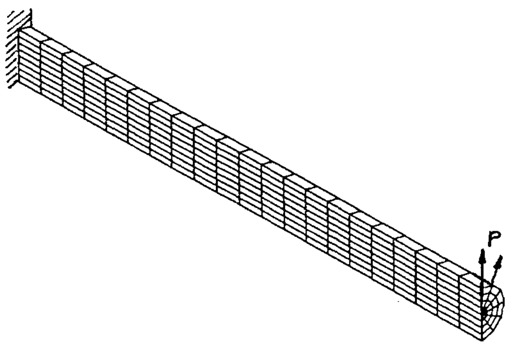

Fig. 5 Three-dimensional finite element model used in analysis by ABAQUS.

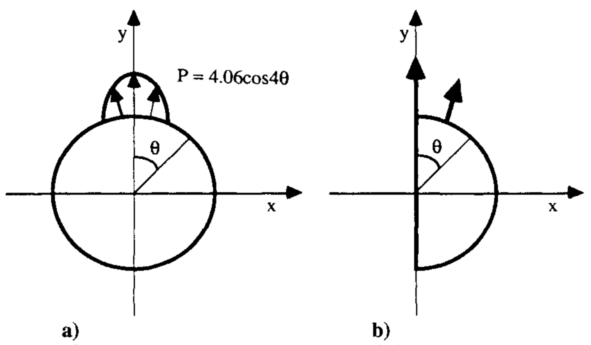

Fig. 6 Loading condition: a) nonaxisymmetrical load and b) loading in ABAQUS.

consists of three routines: Gauss, Elimination, and Substitution. The Gauss is a main subroutine and calls the subroutine Elimination for triangularization and the subroutine Substitution for backward substitution. In solving the simultaneous equations, most of the CPU time is consumed in the Elimination routine. The larger the degree of freedom is, the higher the ratio of CPU time in Elimination is. For this reason, modifying the Elimination subroutine is necessary for speedup in computing time, and code modification is focused on this point. However, it does not mean that modification of Elimination is sufficient for parallel computing. Independent arrays, for example, the stiffness matrix, should be assigned to each mode, and overall modification should be made for parallel computing.

Two kinds of codes, one for sequential processing and the other for parallel processing, are used in the analysis of the computing performance. The elapsed CPU time in main processing (Solver) is mainly considered because the time consumption in preprocessing or postprocessing depends on data generation or outputs. The elapsed CPU time in each subroutine is checked by averaging the CPU time of processors used in every execution.

Numerical Results

Cantilever Beam

As the first example, a circular cantilever beam with a nonsymmetrical end load is examined. The finite element model is shown in Fig. 4, and the number of nodes and elements used are 110 and 84, respectively.

For the verification of the solution, three-dimensional analysis is carried out with ABAQUS whose model is represented in Fig. 5. The loading condition is $\boldsymbol{P} = 4.06 \cos 4\theta$ for $-\pi/8 \leq \theta \leq \pi/8$ and $\boldsymbol{P} = 0$ otherwise, which is shown in Fig. 6a, and the magnitude of the load summation of the half-plane in the y direction is 1. This load is expanded by six terms of Fourier series, and this is given in Fig. 7. In the Abaqus model, equivalent point loads are applied as in Fig. 6b. The normal stress distribution at midspan is compared in Fig. 8. The error may be caused from the slight difference in loading condition between the present code and Abaqus.

Although the beam is discretized by the same mesh, the bandwidth of the stiffness matrix can be changed by the different assignment of node numbers to examine the relationship between the bandwidth and the elapsed time. The relationship between the

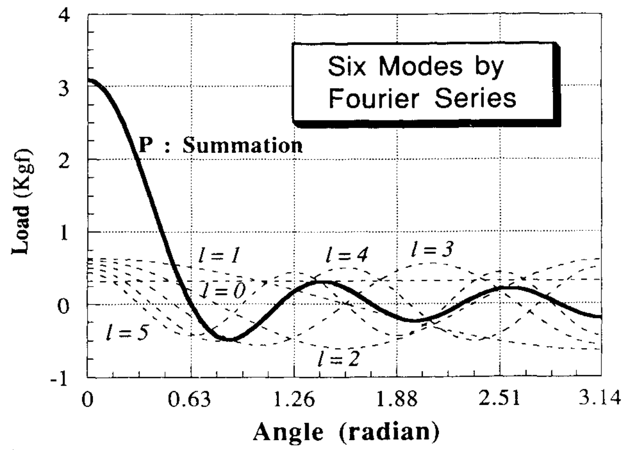

Fig. 7 Fourier series expansion of loading condition.

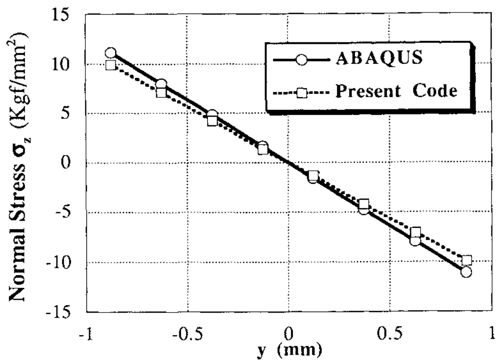

Fig. 8 Comparison of normal stress σ_z at midspan.

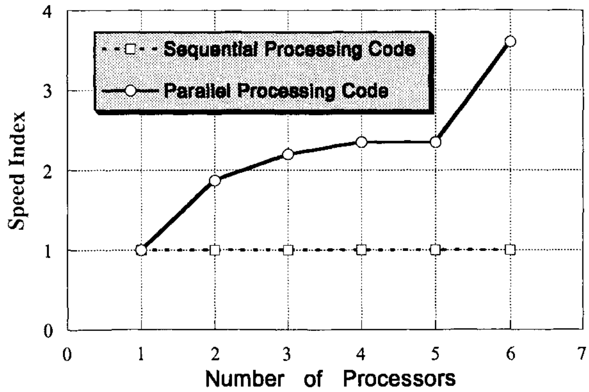

Fig. 9 Relationship between computing speed and number of processors.

computing time and the number of processors is shown in Fig. 9. The speed index is calculated by reference to the speed of sequential processing. When six processors are used, computing speed-up by the parallel processing code is 4.4 times.

It should be noted in Fig. 9 that reasonable speed-up is achieved in the case of using two and three processors, but the processing speed is almost the same in four and five processors as in the three processors. This can be explained from the fact that the idling processors that have no jobs can exist[13] in parallel processing, and total elapsed CPU time counts the averaged elapsed CPU time of the tasking processors. The comparison of calculation procedures of the parallel computer with three and four processors is presented in Fig. 10. In the case of four processors, two idling processors exist. Either with three processors or with four processors, two calculating steps are required. Therefore, it takes the same elapsed time for both cases. From these results, it can be stated that the matching of the number of processors and the number of modes is very important for the improvement of computing time in parallel processing.

The comparison of elapsed CPU time in sequential processing with parallel processing is presented in Table 1. The former takes 93 s, and the latter takes 23 s in total time. The time consumed in

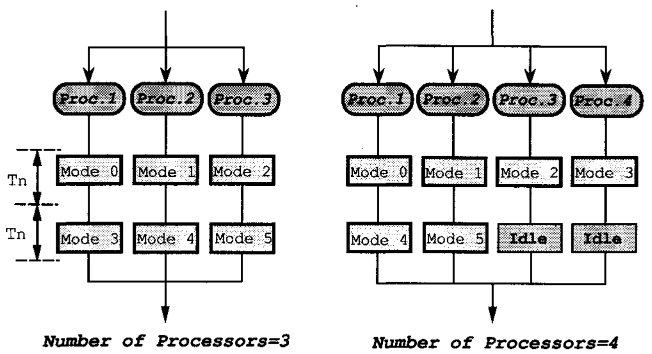

Fig. 10 Comparison of calculation procedures.

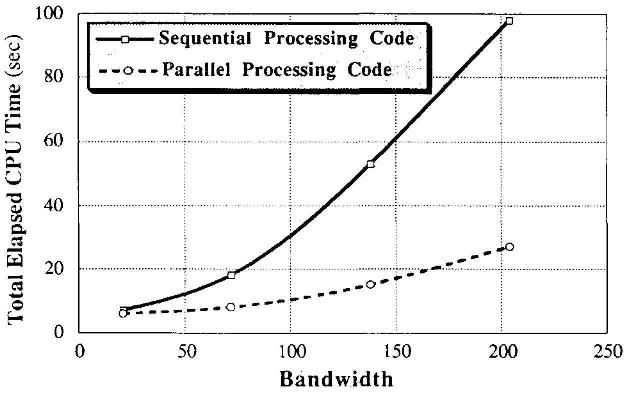

Fig. 11 Effect of bandwidth on total elapsed CPU time.

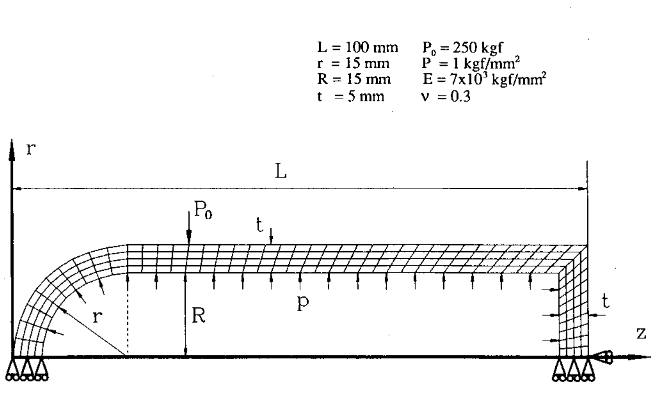

Fig. 12 Semianalytical finite element model of pressure vessel.

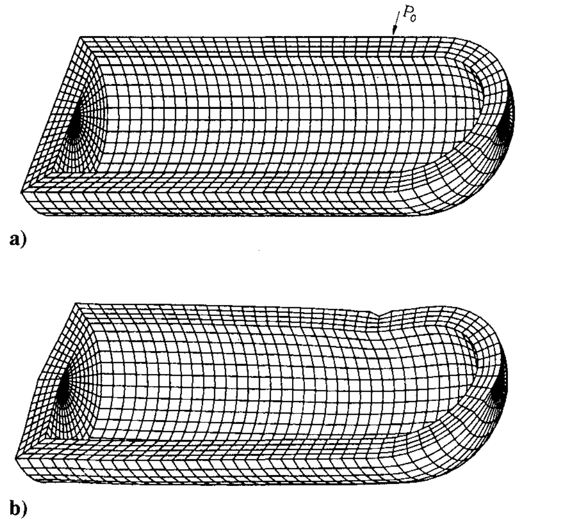

Fig. 13 In ABAQUS: a) three-dimensional finite element model and b) deformed shape magnified 50 times.

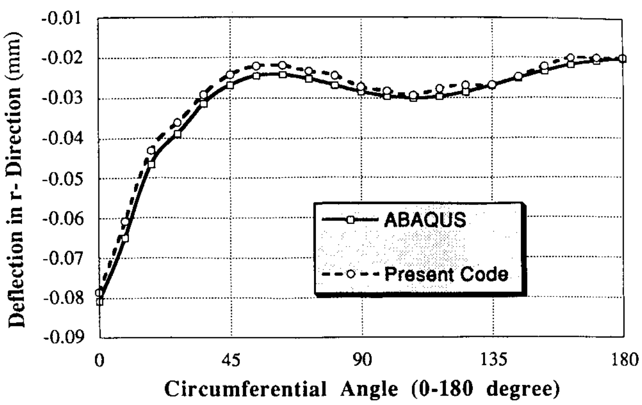

Fig. 14 Comparison of deflection in the r direction along the circumference.

the Elimination subroutine is much reduced from 87 to 16 s, in which the speed-up is 5.4 times.

The relationship between the computing time and the bandwidth of stiffness matrix is shown in Fig. 11, where six processors are used in parallel processing. The degree of freedom is fixed, and the bandwidth is changed by assigning node numbers differently. It is observed that parallel processing is more efficient in a problem with large bandwidth than with small bandwidth.

Pressure Vessel

For the application of this parallel computing technique to a practical engineering problem, a pressure vessel with internal pressure and nonsymmetrical side force is chosen as the second example. The finite element model is shown in Fig. 12 in which the number of nodes and elements used are 225 and 220, respectively. The side force P_0 has a similar pattern with the beam model and is only different in magnitude. This force is also approximated by six terms of Fourier series.

For the comparison of the solution, a three-dimensional analysis is also carried out with Abaqus whose model is represented in Fig. 13a, and equivalent point forces are applied at three points along the circumference. It has 4000 elements and 5155 nodes. The deformed shape of the three-dimensional model is shown in Fig. 13b magnified by 50 times in deformation. The r-directional deflection along the circumference at loaded point is compared in Fig. 14 and gives a good agreement.

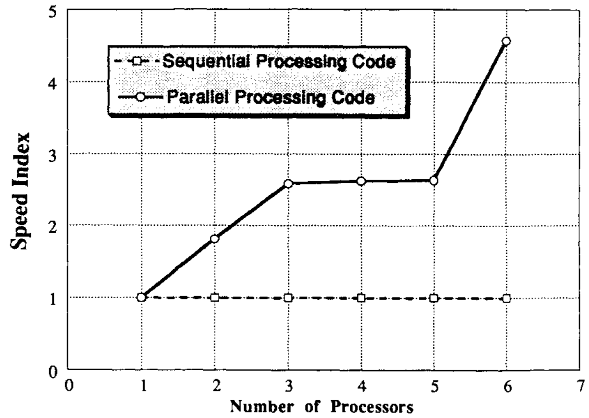

Fig. 15 Relationship between computing speed and number of processors.

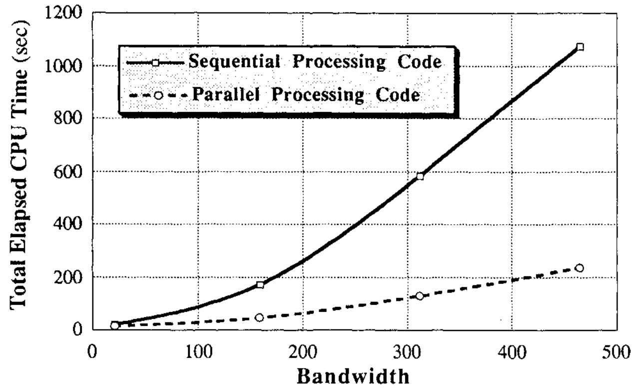

Fig. 16 Effect of bandwidth on total elapsed CPU time.

Table 1 Elapsed CPU time by sequential processing (number of processors = 1) and parallel processing (number of processors = 6)

	Sequential processing code		Parallel processing code	
Routines	Percentage	Time, s	Percentage	Time, s
Elimination	92.1	86.86	67.3	16.18
Stiffness assembly	4.5	4.24	15.4	3.71
Substitution	1.3	1.24	6.5	1.56
Element assembly	1.0	0.98	4.9	1.17
Force assembly	0.1	0.07	0.6	0.14
Gauss	0.0	0.00	0.9	0.22
Total time, s		93.39		22.98

The total computing time by Abaqus is 669 s with the IRIS workstation of Silicon Graphics Co. The computing time by the developed sequential code using the semianalytical method is 40 s with the same IRIS workstation, and this shows the merit of the semianalytical method.

The relationships between the computing time and the number of processors are shown in Fig. 15. The speed index is also calculated by reference to the elapsed time of sequential processing. When six processors are used, computing speed-up by the parallel processing code is 4.6 times. The comparison of the elapsed CPU time in sequential processing with that in parallel processing is presented in Table 2, in which case the bandwidth of the model is 465. The elapsed CPU time in sequential processing is 1074 s and 235 s in parallel processing with six processors of the Alliant FX2812 machine. The time consumed in the Elimination subroutine is much reduced from 1044 to 210 s. In this case, a speedup of 5.0 times is achieved. The relationship between computing time and the bandwidth of the stiffness matrix is presented in Fig. 16, and the results are similar to those in the beam example.

Table 2 Elapsed CPU time according to the number of processors (bandwidth = 465)

Main routines	Number of processors / Elapsed CPU time, s (%) 1	2	3	4	5	6
Elimination	1044.13 (97.50)	567.56 (95.40)	389.96 (93.69)	385.71 (94.23)	384.47 (94.20)	209.90 (89.37)
Stiffness assembly	13.06 (1.22)	12.43 (2.09)	8.94 (2.15)	8.87 (2.17)	8.93 (2.19)	9.04 (3.85)
Substitution	9.29 (0.87)	9.48 (1.59)	9.32 (2.24)	9.21 (2.25)	9.16 (2.24)	8.20 (3.49)
Element assembly	2.90 (0.27)	2.94 (0.49)	2.94 (0.71)	2.90 (0.71)	2.72 (0.67)	2.81 (1.20)
Force assembly	0.24 (0.02)	0.32 (0.05)	0.29 (0.07)	0.18 (0.04)	0.24 (0.06)	0.21 (0.09)
Gauss	1.28 (0.12)	2.20 (0.38)	4.77 (1.14)	2.45 (0.60)	2.60 (0.64)	4.71 (2.00)
Total time	1073.87	594.93	416.22	409.32	408.12	234.87

Conclusions

In this paper, a parallel computing technique using the semianalytical finite element method and its implementation method is studied, and the speed-up in the elapsed computing time is presented. Overall program modification is made to remove the data dependency and each mode is assigned to an individual processor of a parallel computer.

The computing time by parallel processing is compared with that by sequential processing, and a large amount of improvement in computing time is achieved by parallel processing. A speed-up of 4.6 times in elapsed time is observed with six processors of the Alliant FX2812 machine at Seoul National University. It is important to note that the maximum speed can be reached by matching the number of processors and modes since there may exist idling processors. It can be also stated that parallel processing becomes more efficient in a large-size problem than in a small-size problem.

The analysis results of the semianalytical method agree well with those of the three-dimensional analysis by Abaqus, and this method can be applied to many practical engineering problems. This technique can also be adapted to the promising concurrent network computing since the method requires the least data communication between processors (individual nodes in the network system) during computation.

References

[1]Zienkiewicz, O. C., and Taylor, R. L., *The Finite Element Method*, 4th ed., McGraw-Hill, New York, 1989.

[2]Wilson, E. L., "Structural Analysis of Axisymmetric Solids," *AIAA Journal*, Vol. 3, No. 12, 1965, pp. 2269–2274.

[3]Pardon, G. C., Falco, A. D., and Crose, J. G., "Asymmetric Stress Analysis of Axisymmetric Solids with Rectangularly Orthotropic Properties," *AIAA Journal*, Vol. 14, No. 10, 1976, pp. 1419–1426.

[4]Spilker, R. L., and Daugirda, D. M., "Analysis of Axisymmetric Structures Under Arbitrary Loading Using the Hybrid-Stress Model," *International Journal for Numerical Methods in Engineering*, Vol. 17, No. 6, 1981, pp. 801–828.

[5]El-Hawary, M. M., and Herrmann, L. R., "Semi-Analytical Finite-Element Study for Elastomeric Composite Solids of Revolution," *AIAA Journal*, Vol. 27, No. 1, 1989, pp. 87–94.

[6]Sedaghat, M. S., and Herrmann, L. R., "A Nonlinear, Semi-Analytical Finite Element Analysis for Nearly Axisymmetric Solids," *Computers & Structures*, Vol. 17, No. 3, 1983, pp. 389–401.

[7]Farhat, C., and Wilson, E. L., "A New Finite Element Concurrent Computer Program Architecture," *International Journal for Numerical Methods in Engineering*, Vol. 24, No. 9, 1987, pp. 1771–1792.

[8]Farhat, C., Wilson, E. L., and Powell, G., "Solution of Finite Element Systems on Concurrent Processing Computers," *Engineering with Computers*, Vol. 2, 1987, pp. 157–165.

[9]Farhat, C., and Roux, F. X., "A Method of Finite Element Tearing and Interconnecting and Its Parallel Solution Algorithm," *International Journal for Numerical Methods in Engineering*, Vol. 32, No. 6, 1991, pp. 1205–1227.

[10]El-Sayed, M. E., and Hsiung, C. K., "Parallel Finite Element Computation with Separate Substructures," *Computers & Structures*, Vol. 36, No. 2, 1990, pp. 261–265.

[11]Noor, A. K., "Parallel Processing in Finite Element Structural Analysis," *Engineering with Computers*, Vol. 3, 1988, pp. 225–241.

[12]Anon., *FX/FORTRAN-2800 Programmer's Handbook*, Alliant Computer Systems Corp., June 1990.

[13]Fox, G., Johnson, M., Otto, S., Salmon, J., and Walker, D., *Solving Problems on Concurrent Processors*, Vol. I, Prentice-Hall, Englewood Cliffs, NJ, 1988.

[14]Glowinski, R., and Lichnewsky, A., *Computing Methods in Applied Science and Engineering*, Society for Industrial and Applied Mathematics, 1990.