681.3:621

Paper No. 149-14

Direct Simulation of Engineering Problems with a Fast Array Computer*

(2nd Report, Parallel Computation Algorithms using SOR Method)

By Masaaki YOKOYAMA** and Kunikazu HAYASHI***

The finite difference or finite element method, which has been applied successfully to the numerical analyses of many engineering problems, consists of synthesizing linear equations for nodal points of mesh of continuous solids and solving these simultaneous linear equations by a proper algorithm using a conventional computer, which has one or a few arithmetic units (AU).

If an array computer is designed, which has hundreds of AUs connected in a network fashion, the time-consuming finite difference or finite element computations can be executed very fast, provided each AU is attached to each nodal point and parallel computations are performed at each AU according to common timing diagrams.

This report deals with the development of parallel computation algorithms using SOR method and discusses the computation efficiencies, aiming at the development of an array computer which can be used profitably to solve many engineering problems.

1. Introduction

Recently engineering designers in various industrial branches have been faced with large sized numerical computations, since the problems of structural design and of strength design to be solved have become complex and large, and the solutions of them with high accuracy have come to be needed. Usually the time consumed for these computations is very long and the reduction of it is increasing in importance.

To shorten the consumption time the authors intend to use a new type computer system, i.e. an array computer, which has hundreds of arithmetic units connected in a two- or three-dimensional network fashion. The first report⁽¹⁾ has dealt with the basic system of the array computer, the validity of parallel computation methods using Jacobi and modified Jacobi methods, and their computational efficiencies. The results have shown that such parallel computation methods are feasible if a relaxation factor is chosen appropriate, and they can compute engineering problems efficiently. But as is well known, the speed of convergence of Jacobi and Jacobi-like methods is very slow compared with that of SOR method^{(2),(3)}. So from the standpoint of practical usage, SOR method is preferred to as parallel computation method to shorten the consumption time. This report deals with parallel computation methods using SOR method, and discusses the validity of the methods and computational efficiencies by simulating the procedure with a conventional computer.

* Received 31st March, 1975.

** Associate.

*** Professor, Graduate School of Science and Engineering, Tokyo Institute of Technology, 4259 Nagatsuda Midori-ku, Yokohama

2. Basic system of array computer

Basic concept of the system of this array computer is shown in Fig. 1. The system consists of three kinds of arithmetic units, i.e. AU, AU and AU. All arithmetic units are controlled by CU (control unit). AU is a small sized unit, attached to each nodal point for performing basic arithmetic operations. AU computes the sum of AUs' data in column direction. AU is of medium size, and controls the total system and computes the sum of AUs' data in row direction. RAM is a random access memory which stores data such as values of physical quantities and of coefficients. RAM of AU is given enough capacity to store computation programs. SR is a shift register, which enables data transmission between neighboring AUs. ROM is a read only memory and I/O is an input/output apparatus. All AUs basically execute same operations and

Fig. 1 Basic system of array computer

computations simultaneously according to a common timing diagram. But in a few cases some AUs must halt their operations. In Table 1 functions of three arithmetic units necessary for parallel computations are listed.

3. Parallel computation methods using SOR method

Direct simulation with an array computer consists in attaching an AU to each nodal point and performing parallel computations at all AUs. Parallel methods using Jacobi and modified Jacobi methods described in the first report carry out this basic concept faithfully. But from the standpoint of practical usage, SOR method is preferable to shorten the computation time. In parallel computation method using SOR method (called the parallel SOR method henceforward), an AU shares necessarily several nodal points (at least two) and performs simultaneous iterations en bloc sequentially.

For example, in solving a problem using a mesh pattern as shown in Fig. 2(a) by parallel SOR method in which an AU shares 4 nodal points, each AU performs iterative computations of SOR method as follows. Assuming that the number of unknown physical quantities for each nodal point is one, iterative equations for the 22th AU, for example, are given as

1st nodal point

$$\begin{aligned}\tilde{x}_{33}^{(v+1)} &= [b_{33} - (a_{22}x_{22}^{(v)} + a_{23}x_{23}^{(v)} + a_{32}x_{32}^{(v)} \\ &\quad + a_{34}x_{34}^{(v)} + a_{43}x_{43}^{(v)} + a_{44}x_{44}^{(v)})] / a_{33} \\ x_{33}^{(v+1)} &= x_{33}^{(v)} + (\tilde{x}_{33}^{(v+1)} - x_{33}^{(v)})\end{aligned}$$

2nd nodal point

$$\begin{aligned}\tilde{x}_{34}^{(v+1)} &= [b_{34} - (a_{23}x_{23}^{(v)} + a_{24}x_{24}^{(v)} + a_{33}x_{33}^{(v+1)} \\ &\quad + a_{35}x_{35}^{(v+1)} + a_{44}x_{44}^{(v)} + a_{45}x_{45}^{(v)})] / a_{34} \\ x_{34}^{(v+1)} &= x_{34}^{(v)} + (\tilde{x}_{34}^{(v+1)} - x_{34}^{(v)})\end{aligned}$$

3rd nodal point

$$\begin{aligned}\tilde{x}_{43}^{(v+1)} &= [b_{43} - (a_{32}x_{32}^{(v+1)} + a_{33}x_{33}^{(v+1)} + a_{42} \\ &\quad x_{42}^{(v)} + a_{44}x_{44}^{(v)} + a_{53}x_{53}^{(v+1)} + a_{54}x_{54}^{(v+1)})] / a_{43} \\ x_{43}^{(v+1)} &= x_{43}^{(v)} + (\tilde{x}_{43}^{(v+1)} - x_{43}^{(v)})\end{aligned}$$

4th nodal point

$$\begin{aligned}\tilde{x}_{44}^{(v+1)} &= [b_{44} - (a_{33}x_{33}^{(v+1)} + a_{34}x_{34}^{(v+1)} \\ &\quad + a_{43}x_{43}^{(v+1)} + a_{45}x_{45}^{(v+1)} + a_{54}x_{54}^{(v+1)} + a_{55} \\ &\quad x_{55}^{(v+1)})] / a_{44} \\ x_{44}^{(v+1)} &= x_{44}^{(v)} + (\tilde{x}_{44}^{(v+1)} - x_{44}^{(v)})\end{aligned}$$

.....(1)

a_{ij} are the elements of coefficients matrix, b_{ij} are external forces, ω is relaxation factor, and v indicates the number of iterations. In eq. (1) the sequence of computations is 1-2-3-4, but this sequence can be changed wherever necessary.

In parallel SOR method an AU shares several nodal points, and the number affects the feasibility of the method, i.e. the number of iterations before reaching convergence and/or the computational efficiency. So at first, optimum number of nodal points shared by an AU must be decided by numerical experiments. Fig. 3 shows the problems used for the numerical experiments of the present paper. Problem (A) is a cantilever with a load at its top, and problem (B) is a block of infinite length symmetrically loaded at its central points. The 13-points finite difference method applied to bi-harmonic equation is relevant to the problem

Table 1 Functions of arithmetic units necessary for parallel computation

arithmetic unit	necessary functions
AU	arithmetic operations, absolute evaluation, comparison, conditional halt
$\bar{A}U$	addition, subtraction
$\bar{A}U$	arithmetic operations, absolute evaluation, comparison, square rooting

(a)

(b)

Fig. 2 Parallel SOR method sharing each AU 4 nodal points

(A) and finite element method to problem (B). Boundary conditions for (A) are such that at clamped edge both the deflection and inclination are zero. In computation $\epsilon < 10^{-4}$ is adopted as a condition for convergence criterion. Where ϵ stands for the maximum relative error defined as

$$\epsilon = \max \left| \frac{\mathbf{x}^{(v+1)} - \mathbf{x}^{(v)}}{\mathbf{x}^{(v+1)}} \right|, \text{ where } \mathbf{x}^{(v+1)} \neq 0$$

.....(2)

Computations were carried out using single precision on HITAC 8700. Fig. 4 shows the sequence of computations for five cases with different numbers of nodal points shared by an AU. The directions axes refer to those shown in Fig. 3. Fig. 5 shows the relation between the number of iterations v before reaching convergence in relation to the number of nodal points shared by an AU. Vertical arrows in the figure indicate that if the relaxation factor ω exceeds the values at those points the parallel SOR fails. The optimum number of nodal points shared by an AU can be decided from these data. The computation time for convergence is proportional to

(number of iterations v before convergence) $\times$ (number of nodal points shared by an AU)

Fig. 6 shows the relation between the number of nodal points shared by an AU and the above mentioned proportionality factor. From Fig. 6 it can be deduced that the most efficient method is the parallel SOR method in which an AU shares 4 nodal points.

If the relaxation factor is chosen appropriate the parallel SOR methods will be valid regardless of the number of nodal points shared by an AU. However the case of sharing 4 nodal points by an AU is probably the most efficient. So in the following chapter this case will be studied in more detail.

4. Parallel SOR computation using an AU sharing 4 nodal points

Comparative studies of parallel SOR using an AU sharing 4 nodal points and normal SOR on the problems (A) and (B) described in the preceding chapter and the problem (C), finite element deflection analysis of rectangular plate supported at its 4 edges rigidly and loaded at its center, were done. Element used for the problem (C) is a square one characterised by a displacement function of 4th-order polynomial with 12 parameters⁽⁴⁾. The degree of freedom per each nodal point is 1 for problem (A), 2 for (B) and 3 for (C). Sequence of com-

putations is the same as that shown in Fig. 4. Convergence criterion is $\epsilon < 10^{-4}$, and computations were executed using single precision on HITAC 8700. Typical results are shown in Figs. 7 and 8, which give the relation between the number of iterations v before reaching convergence and relaxation factor ω for problems (A) and (B). The lines through solid circles show the results of parallel SOR in the present study. These are compared with those of normal SOR method. From these figures the superiority of the present method is appreciated. Principal results obtained for three problems studied are summarized in Table 2. Fig. 9 shows the effect of the value of ϵ of convergence criterion on v . From the figure it is seen that way of convergence of iterati-

Fig. 3 Problems submitted to numerical experiments

Fig. 4 Relation between sequence of computations and number of nodal points

Fig. 5 Relation between the number of iterations v before convergence and number of nodal points shared by an AU

ons is affected a little by ϵ . For cases of $\epsilon=10^{-5}$ and 10^{-6} in Fig. 9, the computations were executed using double precision.

From the numerical experiments described above, it is concluded that the parallel SOR method adopting an AU sharing 4 nodal points is workable if a relaxation factor is chosen appropriate.

In the next place, the computational efficiency will be discussed. The computation by the present SOR method using the array computer is ideally $N/4$ (the number of AUs used) times faster than that by a conventional computer, but in reality this ratio diminishes to a

value which is less than $N/4$ due to the difference of convergence property of computational procedures, which can be accommodated in both computers. This ratio is defined by

$$\frac{\text{comp. speed of conve.}}{\text{comp. speed of array computer}} = \frac{\text{number of iterations by conve. comp. } v_N}{\text{number of iterations by array comp. } v_A}$$

x number of AUs used $N/4$
 v_N and v_A are defined as the values at optimum relaxation factors. N is the total number of nodal points. In Table 2 are shown the results of this ratio found for three problems studied, where $\epsilon < 10^{-4}$ is adopted. Comparison of the ratio from the table indicates that the efficiency of the array computer using parallel SOR method using an AU sharing 4 nodal points, almost amounts to the ideal value.

5. Convergence of iterations and the choice of relaxation factor

5.1 Convergence of iterations Assuming that the degree of freedom, i.e. the number of relevant physical quantities for each nodal point is one, iterative equations, corresponding to eq. (1), for all AUs are written as

$$\begin{aligned} \text{1st group of nodal points} \\ \tilde{x}_1^{(v+1)} = [b_1 - (A_{12}x_2^{(v)} + A_{13}x_3^{(v)} + A_{14}x_4^{(v)})] / a_{11} \end{aligned}$$

Fig. 6 Relation between the number of nodal points shared by an AU and the number of iterations v before convergence x number of nodal points shared by an AU

Fig. 7 Relation between the number of iterations v before convergence and relaxation factor ω (problem (A))

$$\begin{aligned}
 \mathbf{x}_1^{(v+1)} &= \mathbf{x}_1^{(v)} + \omega(\tilde{\mathbf{x}}_1^{(v+1)} - \mathbf{x}_1^{(v)}) \\
 \text{2nd group of nodal points} \\
 \tilde{\mathbf{x}}_2^{(v+1)} &= [\mathbf{b}_2 - (\mathbf{A}_{21}\mathbf{x}_1^{(v+1)} + \mathbf{A}_{23}\mathbf{x}_3^{(v)} + \mathbf{A}_{24}\mathbf{x}_4^{(v)})] / \mathbf{a}_{22} \\
 \mathbf{x}_2^{(v+1)} &= \mathbf{x}_2^{(v)} + \omega(\tilde{\mathbf{x}}_2^{(v+1)} - \mathbf{x}_2^{(v)}) \\
 \text{3rd group of nodal points} \\
 \tilde{\mathbf{x}}_3^{(v+1)} &= [\mathbf{b}_3 - (\mathbf{A}_{31}\mathbf{x}_1^{(v+1)} + \mathbf{A}_{32}\mathbf{x}_2^{(v+1)} + \mathbf{A}_{34}\mathbf{x}_4^{(v)})] / \mathbf{a}_{33} \\
 \mathbf{x}_3^{(v+1)} &= \mathbf{x}_3^{(v)} + \omega(\tilde{\mathbf{x}}_3^{(v+1)} - \mathbf{x}_3^{(v)}) \\
 \text{4th group of nodal points} \\
 \tilde{\mathbf{x}}_4^{(v+1)} &= [\mathbf{b}_4 - (\mathbf{A}_{41}\mathbf{x}_1^{(v+1)} + \mathbf{A}_{42}\mathbf{x}_2^{(v+1)} + \mathbf{A}_{43}\mathbf{x}_3^{(v+1)})] / \mathbf{a}_{44} \\
 \mathbf{x}_4^{(v+1)} &= \mathbf{x}_4^{(v)} + \omega(\tilde{\mathbf{x}}_4^{(v+1)} - \mathbf{x}_4^{(v)})
 \end{aligned}
 \tag{3}$$

$\mathbf{A}_{ij}$, $\mathbf{x}_i$, $\mathbf{b}_i$ are the elements of the matrix equation shown in Fig. 10, which is a partitioned form of

$$\mathbf{A} \mathbf{x} = \mathbf{b} \tag{4}$$

$\mathbf{a}_{ii}$ are diagonal elements of $\mathbf{A}_{ii}$. From now, let $\mathbf{a}_{ii}$ represent a square matrix, whose diagonal elements are $\mathbf{a}_{ii}$ and off-diagonal elements are zero. Then, for the finite element mesh of Fig. 2(a)

$$\mathbf{I} / \mathbf{a}_{ii} = \mathbf{A}_{ii}^{-1} \tag{5}$$

is satisfied. Substituting eq. (5) into eq. (3) yields the iterative equations which coincide with the well known SBOR method⁽⁵⁾. This means that the present parallel SOR method, when executed on a conventional computer, is the same as SBOR method. Let the matrix $\mathbf{D}$ be defined as in Fig. 11. If $\mathbf{D}$ is positive definite, the SBOR method is convergent for all $\mathbf{x}^{(0)}$ if and only if $0 < \omega < 2$ and $\mathbf{A}$ is positive definite⁽⁶⁾. Coefficients matrix $\mathbf{A}$ derived from finite element method is real and symmetric, and hence Hermitian as well as positive definite. Further, if

Table 2 Results of numerical experiments

No.	problem	number of nodal points N	number of iterations in parallel SOR method v_A	number of iterations in normal SOR method v_N	$\frac{v_N}{v_A} \times \frac{N}{4}$	number of used AUs in Parallel SOR method $N/4$	ω_{opt}	ω_{cal}	number of calculated iterations by $\frac{\omega_{cal}}{\omega_{opt}}$	$\frac{v_N}{v_A}$
1	(A)	4x9	103	143	12.5	9	1.600	1.50	438	4.25
2		6x12	204	403	35.6	18	1.625	1.50	954	4.67
3		8x18	2221	3121	50.6	36	1.575	1.60	4171	1.87
4		12x24	5377	9753	130.6	72	1.625	1.60	6940	1.29
5	(B)	4x12	78	80	12.3	12	1.450	1.40	107	1.37
6		8x24	242	260	52.8	48	1.650	1.60	450	1.85
7		16x48	889	889	192.0	192	1.600	1.70	1124	1.26
8		24x72	2369	2129	388.2	432	1.625	1.10	3664	1.54
9*		6x12	1665	1849	20.0	18	1.600	1.30	2914	1.74
10*		6x26	2897	3078	41.4	39	1.625	1.40	3947	1.36
11	(C)	8x16	622	723	37.1	32	1.550	1.35	918	1.47
12		16x32	2623	2984	145.6	128	1.600	1.40	3541	1.35

* finite element method, unequal mesh spacing

Fig. 8 Relation between the number of iterations v before convergence and relaxation factor ω (problem (B))

the matrix A is Hermitian and positive definite, any partitioning of A directly guarantees the positive definite Hermitian character of D . Thus the parallel SOR method using an AU sharing 4 nodal points is theoretically convergent if a relaxation factor is chosen as $0 < \omega < 2$. If the number of unknown quantities is two, the matrix equation is partitioned into 8 blocks.

So far discussed is the case of finite element method. For the finite difference method, the coefficients matrix is not necessarily symmetric, but can be regarded symmetric for the large overall matrix equations related with the engineering problems under consideration. The matrix A is positive definite, hence SBOR method is convergent if ω satisfies $0 < \omega < 2$. Assuming that the number of degrees of freedom per nodal point is one, the iterative equations of parallel SOR method in case of sharing 4 nodal points per AU for the finite difference method are written as eq. (6). In these equations the matrix A_{ii}^* is $A_{ii} - a_{ii}$, and this reduced to zero in the case of the finite element method or 5-points finite difference scheme. For 13-points finite difference scheme⁽⁷⁾ applied to bi-harmonic equation, the elements of A_{ii}^* are the coefficients connected with nodal points apart 2 mesh intervals, and are 1/20 of those for the central nodal point.

Therefore, the elements of A_{ii}^* are very small, and the matrix A_{ii}^* is practically equal to I/a_{ii} . Hence the parallel SOR method is again the same as SBOR method. The SBOR method is convergent if ω satisfies $0 < \omega < 2$, and so is the parallel SOR method.

5.2 Choice of relaxation factors It has been confirmed by the discussion of the preceding section that the parallel SOR method using an AU sharing 4 nodal points is convergent if ω satisfies $0 < \omega < 2$, but the most important problem is the choice of a relaxation factor, which makes the convergence the fastest. Up to the present, the method for choosing an appropriate factor has not yet been established except for a system of linear

1st group of nodal points

$$\begin{aligned}\tilde{x}_1^{(v+1)} &= [b_1 - (A_{11}^* x_1^{(v)} + A_{12} x_2^{(v)} \\ &\quad + A_{13} x_3^{(v)} + A_{14} x_4^{(v)})] / a_{11} \\ x_1^{(v+1)} &= x_1^{(v)} + \omega(\tilde{x}_1^{(v+1)} - x_1^{(v)})\end{aligned}$$

2nd group of nodal points

$$\begin{aligned}\tilde{x}_2^{(v+1)} &= [b_2 - (A_{21} x_1^{(v+1)} + A_{22}^* x_2^{(v)} \\ &\quad + A_{23} x_3^{(v)} + A_{24} x_4^{(v)})] / a_{22} \\ x_2^{(v+1)} &= x_2^{(v)} + \omega(\tilde{x}_2^{(v+1)} - x_2^{(v)})\end{aligned}$$

3rd group of nodal points

$$\begin{aligned}\tilde{x}_3^{(v+1)} &= [b_3 - (A_{31} x_1^{(v+1)} + A_{32} x_2^{(v+1)} \\ &\quad + A_{33}^* x_3^{(v)} + A_{34} x_4^{(v)})] / a_{33} \\ x_3^{(v+1)} &= x_3^{(v)} + \omega(\tilde{x}_3^{(v+1)} - x_3^{(v)})\end{aligned}$$

4th group of nodal points

$$\begin{aligned}\tilde{x}_4^{(v+1)} &= [b_4 - (A_{41} x_1^{(v+1)} + A_{42} x_2^{(v+1)} \\ &\quad + A_{43} x_3^{(v+1)} + A_{44}^* x_4^{(v)})] / a_{44} \\ x_4^{(v+1)} &= x_4^{(v)} + \omega(\tilde{x}_4^{(v+1)} - x_4^{(v)})\end{aligned}$$

... (6)

Fig. 9 Relation between the number of iterations v before convergence of parallel SOR method and ϵ

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix}$$

Fig. 10 Partition of matrix equation into blocks

$$D = \begin{bmatrix} A_{11} & & & \\ & A_{22} & & 0 \\ & & A_{33} & \\ 0 & & & A_{44} \end{bmatrix}$$

Fig. 11 Definition of matrix D

equations, whose coefficients matrix has Young's property $A^{(8)}$. Therefore, the optimum factor should be determined by numerical experiments. In the present paper, a rather tedious procedure is adopted, by which optimum value ω_{opt} is evaluated by a series of computations using different values of ω .

The iterative procedure of SBOR for the equations partitioned into blocks as shown in Fig. 10 is written in the form⁽⁹⁾,

$$x^{(v+1)} = Mx^{(v)} + g \quad \dots\dots\dots (7)$$

where

$$M = (D - \omega E)^{-1} [(1 - \omega)D + F]$$

$$g = \omega (D - \omega E)^{-1} b$$

The matrix A under consideration is decomposed as

$$A = D - E - F \quad \dots\dots\dots (8)$$

where D , E , F are diagonal, lower-left triangular, and upper-right triangular matrices respectively. Now, let e be

$$e^{(v+1)} = x^{(v+1)} - x^{(v)} \quad \dots\dots\dots (9)$$

Then from eqs. (7), (9)

$$e^{(v+1)} = Me^{(v)} \quad \dots\dots\dots (10)$$

is derived. Necessary and sufficient condition for the convergence of SBOR method is that spectral radius of M , i.e. maximum value of absolute eigenvalue, is less than 1. The smaller the value is, the faster the convergence of the method is. Eq. (10) assumes a form similar to that in power method⁽¹⁰⁾, and after sufficient number of iterations the ratios of corresponding elements in succeeding vector $e^{(v+1)}$ and $e^{(v)}$ converge to the maximum value of the absolute eigenvalue of matrix M . In ordinary way, the ratio of Euclidean norms is used in place of the elements of eigenvector. Relaxation factor, which makes this maximum value smallest, is chosen in the following way.

After carrying out N_1 iterations of parallel SOR method at $\omega=1$, and using x at that stage of iteration as initial vector $x^{(0)}$, iterative process is repeated N_2 times for the value of $\omega=1.2, 1.4, 1.6, 1.8$ respectively. Calculate eigenvalues corresponding to each relaxation factor, and choose a factor which gives the minimum value as ω^* . After N_1 iterations for ω^* , and using x at that stage as initial vector $x^{(0)}$, iteration process is repeated N_2 times for the values of $\omega=\omega^*-0.1, \omega^*-0.05, \omega^*+0.05, \omega^*+0.1$ respectively. The value of which gives the minimum eigenvalue is the one which is chosen as the optimum relaxa-

tion factor ω_{cal} . With ω_{cal} , the iterative process is further continued until the condition of convergence is fulfilled. In Table 2 the relaxation factor ω_{cal} for the case of $N_1=N_2=20$ and the total number of iterations N_w are shown. The results suggest that the method of determination of ω_{cal} is acceptable, even though ω_{cal} is smaller compared with ω_{opt} obtained by numerical experiments.

6. Conclusive remarks

To shorten the time consumed for design computations in many engineering branches, the authors intend to use a new type computer system, i.e. an array computer, which has hundreds of arithmetic units connected in a two- or three-dimensional network fashion. This report has dealt with parallel computation methods using SOR method, and discussed the validity of the methods and computational efficiencies by simulating the procedures with a conventional computer. The results of the simulation show that the parallel computation methods are feasible if a relaxation factor is chosen appropriate and they can compute engineering problems efficiently, and particularly that the parallel computation method using SOR method using an AU sharing 4 nodal points is the best method. Further a method for the choice of appropriate relaxation factors is proposed.

References

- (1) Hayashi, K. and Yokoyama, M., Bull. JSME, 20-149 (1977), 1438.
- (2) Varga, R.S., Matrix Iterative Analysis (1962), 113, Prentice-Hall.
- (3) Yamada, Y. (ed.), The Application of Matrix Method (1972), 299, The Univ. of Tokyo Press.
- (4) Zienkiewicz, O.C. and Cheng, Y.K., The Finite Element Method in Engineering Science (1971), 177, McGraw-Hill.
- (5) Togawa, H., Numerical Analysis of Matrix (1971), 86, Ohm Co.
- (6) 80 page of (2).
- (7) Timoshenko, S. and Goodier, J.N., Theory of Elasticity (1951), 489, McGraw-Hill.
- (8) 99 page of (6).
- (9) Westlake, J.R., A Handbook of Numerical Matrix Inversion and Solution of Linear Equations (1968), 82, John Wiley & Sons.
- (10) Hirano, S., et al., Computation Technique and Numerical Analysis (1971), 142, Baifukan Co..