ISE-TR-90-86

On the Efficiency of an SOR-like Method Suited to Vector Processor

by

Masaaki Sugihara, Yoshio Oyanagi, Masatake Mori and Seiji Fujino

October 4, 1990

INSTITUTE
OF
INFORMATION SCIENCES AND ELECTRONICS
UNIVERSITY OF TSUKUBA

On the Efficiency of an SOR-like Method Suited to Vector Processor *

Masaaki Sugihara

Department of Economics, Hitotsubashi University

Kunitachi, Tokyo, 186 Japan

Yoshio Oyanagi

Institute of Information Sciences, University of Tsukuba

Tsukuba, Ibaraki, 305 Japan

Masatake Mori

Faculty of Engineering, University of Tokyo

Bunkyo-ku, Tokyo, 113 Japan

Seiji Fujino

Institute of Computational Fluid Dynamics

Haramachi 2-1-4, Meguro-ku, Tokyo, 152 Japan

October 1990

Abstract

A simple implementation on vector processors of the SOR method for solving linear system of equations arising from the discretization of partial differential equations makes the SOR method into another, which, though, looks like the SOR method. In this paper, the efficiency of this SOR-like method (pseudo-SOR method) is investigated.

*Invited talk presented by Oyanagi at *International Congress on Computational and Applied Mathematics* held at Katholieke Universiteit Leuven, Belgium, on July 23–28, 1990.

For the Poisson equation in a rectangular region, in both five-point and nine-point discretization, we prove analytically that the optimal acceleration parameter is smaller and the optimal convergence rate is lower in the pseudo-SOR method. Comparison on several vector computers was made between the SOR method vectorized with the hyperplane technique and the pseudo-SOR method. It turned out that the hyperplane SOR is superior to the pseudo-SOR method although the latter is easily vectorizable on vector processors. We also tested an example of Poisson equation descretized in a curved coordinates in a region bounded by two eccentric circles and found numerically that the optimal acceleration parameter becomes small and the convergence becomes slow in the pseudo-SOR method, while the optimal acceleration parameter in the SOR method does not change appreciably. The genuin SOR method is superior to the pseudo-SOR method.

1 Introduction

Various vectorization techniques have been proposed to improve the computational speed on vector processors. The essential ingredients are to remove recursion in loops and to make the vector length as long as possible. We will consider the vectorization of the SOR(Successive OverRelaxation) method for two-dimentional partial differential equations.

The following discussion holds for general linear second-order elliptic partial differential equations. For simplicity, however, we will start with the Poisson equation in a rectangular region $\mathcal{D} = \{(x, y) \mid 0 < x < 1, 0 < y < 1\}$,

$$\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0 \text{ on } \mathcal{D},\ u(x,y) = \psi(x,y) \text{ on } \partial\mathcal{D}. \tag{1}$$

Discretization of the second derivatives of (1) by five-point or nine-point finite differences on an $(N+1) \times (N+1)$-grid with grid spacing $h = 1/N$ leads to a set of linear equations:

$$U_{i-1,j} + U_{i+1,j} + U_{i,j-1} + U_{i,j+1} - 4U_{i,j} = 0 \tag{2}$$

or

$$4(U_{i-1,j} + U_{i+1,j} + U_{i,j-1} + U_{i,j+1}) + U_{i+1,j-1} + U_{i+1,j+1} + U_{i-1,j-1} + U_{i-1,j+1} - 20U_{i,j} = 0 \tag{3}$$

for $i, j = 1, 2, \ldots, N-1$, which form a system of linear equations for the solution vector U.

Eqs.(2) and (3) can be solved with the polular SOR method. We will show here a program for the five-point case:

```
   DO 10 J=1, N-1
     DO 20 I=1, N-1
       U(I,J)=U(I,J)+OMEGA*(0.25*(U(I,J-1)+U(I,J+1)+U(I-1,J)+U(I+1,J))-U(I,J))
20   CONTINUE
10 CONTINUE
```

Here, OMEGA(ω) is the acceleration parameter. Running with i and j from 1 to $N-1$, two neighboring grid points of (i,j), $(i-1,j)$ and $(i,j-1)$ have already been updated, so that $U_{i,j}$ is computed with these new values and old values of $U_{i+1,j}$ and $U_{i,j+1}$. The SOR method, therefore, is recursive in both i- and j-directions. This data dependency makes this loop hard to be vectorized.

There are some well-known techniques to vectorize the SOR method [1]. We will consider here a so-called hyperplane method [2], which essentially reorders the numbering of the lattice points without changing the algorithm. The program will be as follows:

```
   DO 10 K=2, 2*(N-1)
*VOPTION NODEP(U)
     DO 20 J=MAX(1,K+1-N), MIN(N-1,K-1)
       I=K-J
       U(I,J)=U(I,J)+OMEGA*(0.25*(U(I,J-1)+U(I,J+1)+U(I-1,J)+U(I+1,J))-U(I,J))
20   CONTINUE
10 CONTINUE
```

This program is methematically equivalent to the SOR method. The inner loop has no recursion and the access to the array U is with only constant strides. However the hyperplane SOR method involves short average vector length $(N-1)^2/(2N-3) \approx N/2$ and ordinary vectorizing compilers cannot recognize that there are no data dependencies, so that we have to insert a directive, *VOPTION in this example, to the compiler. For some vector computers, access with stride more than one may cause a decrease in the data transfer rate from memory to vector registers and shorter vector length may increase relative start-up time. Nine-point discretization can also be processed by a hyperplane method, but with average vector length $\approx N/3$.

A more naive implementation of this loop on a vector processor is to modify the program as

```
   DO 10 J=1, N-1
     DO 20 I=1, N-1
       V(I)=U(I,J) + OMEGA*(0.25*(U(I,J-1)+U(I,J+1)+U(I-1,J)+U(I+1,J))-U(I,J))
20   CONTINUE
     DO 30 I=1, N-1
30     U(I,J)=V(I)
10 CONTINUE
```

In this program the data dependency in the SOR method is removed so that the loops can be successfully vectorized. The program involves only stride one access to the memory. Obviously this is a different algorithm, since $U_{i-1,j}$ on the right hand side comes from the old iteration level in contrast to the SOR method. We will call it the pseudo-SOR method. The pseudo-SOR method is sometimes used as a simple vectorization technique in computational fluid dynamics.

In the next section, the optimal acceleration parameter ω and the rate of converngence of the two methods for five point discretization is discussed. Similar arguments for nine-point discretization are given in section 3. Numerical results on various vector processors are shown in section 4, including one for curvilinear coordinates. In section 5, we will conclude that the hyperplane SOR is superior to the pseudo-SOR method.

2 Comparison of SOR and pseudo-SOR Methods for Five-Point Discretization

2.1 Matrix Formulation

Equation(2) forms a system of liner equations with matrix A_5 for the solution vector u. If we order the elements in the solution vector lexicographically as

$$\left(u_{1,1}, u_{1,2}, \ldots, u_{1,N-1}, u_{2,1}, u_{2,2}, \ldots, u_{2,N-1}, \ldots, u_{N-1,N-1}\right),$$

the coefficient matrix A_5 is,

$$A_5 = \begin{pmatrix} B_5 & C_5 & & & & \mathbf{0} \\ C_5 & B_5 & C_5 & & & \\ & \cdots & \cdots & & & \\ & & \cdots & \cdots & & \\ & & & C_5 & B_5 & C_5 \\ \mathbf{0} & & & & C_5 & B_5 \end{pmatrix}, \tag{4}$$

with

$$B_5 = \begin{pmatrix} 1 & -\frac{1}{4} & & & \mathbf{0} \\ -\frac{1}{4} & 1 & -\frac{1}{4} & & \\ & \cdots & \cdots & & \\ & & \cdots & \cdots & \\ \mathbf{0} & & & -\frac{1}{4} & 1 \end{pmatrix}, C_5 = \begin{pmatrix} -\frac{1}{4} & & & & \mathbf{0} \\ & -\frac{1}{4} & & & \\ & & -\frac{1}{4} & & \\ & & & \cdots & \\ \mathbf{0} & & & & -\frac{1}{4} \end{pmatrix}. \tag{5}$$

The iteration in the SOR method is written as usual,

$$u^{(m+1)} = \mathcal{L}_5^{SOR} u^{(m)} + b' = (I - \omega L_5^{SOR})^{-1}((1-\omega)I + \omega U_5^{SOR})u^{(m)} + b' \tag{6}$$

where

$$L_5^{SOR} = \begin{pmatrix} -B_5^L & & & & \mathbf{0} \\ -C_5 & -B_5^L & & & \\ & \cdots & \cdots & & \\ & & \cdots & \cdots & \\ \mathbf{0} & & & -C_5 & -B_5^L \end{pmatrix}, B_5^L = \begin{pmatrix} 0 & & & & \mathbf{0} \\ -\frac{1}{4} & 0 & & & \\ & \cdots & \cdots & & \\ & & \cdots & \cdots & \\ \mathbf{0} & & & -\frac{1}{4} & 0 \end{pmatrix}, \tag{7}$$

and

$$U_5^{SOR} = (L_5^{SOR})^T. \tag{8}$$

On the other hand, the iteration in the pseudo-SOR method is formulated

$$u^{(m+1)} = \mathcal{L}_5^{p-SOR} u^{(m)} + b' = (I - \omega L_5^{p-SOR})^{-1}((1-\omega)I + \omega U_5^{p-SOR})u^{(m)} + b'' \tag{9}$$

where

$$L_5^{p-SOR} = \begin{pmatrix} 0 & & & & \mathbf{0} \\ -C_5 & 0 & & & \\ & \cdots & \cdots & & \\ & & \cdots & \cdots & \\ \mathbf{0} & & & -C_5 & 0 \end{pmatrix}, \tag{10}$$

and

$$U_5^{p-SOR} = \begin{pmatrix} (-B_5^L - B_5^U) & -C_5 & & & \mathbf{0} \\ & (-B_5^L - B_5^U) & -C_5 & & \\ & \cdots & \cdots & & \\ & & & (-B_5^L - B_5^U) & -C_5 \\ \mathbf{0} & & & & (-B_5^L - B_5^U) \end{pmatrix}, \quad (11)$$

with $B_5^U = (B_5^L)^T$.

2.2 Acceleration Parameter ω and the Spectral Radius

The order of convergence of Eq.(6) and Eq.(9) is determined by the spectral radius of $\mathcal{L}_5^{SOR}$ and $\mathcal{L}_5^{p-SOR}$, respectively. As an illustration, $\rho(\mathcal{L}_5^{SOR})$ and $\rho(\mathcal{L}_5^{p-SOR})$ are numerically estimated for $N = 6$ and shown in Fig. 1. The result shows:

1. The optimal acceleration parameter ω_{opt} for the pseudo-SOR method is substantially smaller than that for the SOR method.

2. The rate of convergence of the pseudo-SOR method using the optimal acceleration parameter is worse than that of the SOR method.

3. The region of the acceleration parameter which ensures the convergence of the iteration is narrower for the pseudo-SOR method.

In the next subsection, we will generalize the results to arbitrary N.

2.3 Analytic Determination of the Optimal Acceleration Parameter of the SOR and the pseudo-SOR Methods

The optimal acceleration parameter of the SOR method for our case and the corresponding spectral radius is given by Varga [3]:

$$\omega_{opt} = \frac{2}{1+\sin(\frac{\pi}{N})}, \lambda_{opt} = \rho(\mathcal{L}_5^{SOR}(\omega_{opt})) = \omega_{opt} - 1. \quad (12)$$

Asymptotically,

$$\omega_{opt} = 2 - \frac{2\pi}{N} + \mathcal{O}(\frac{1}{N^2}), \lambda_{opt} = 1 - \frac{2\pi}{N} + \mathcal{O}(\frac{1}{N^2}). \quad (13)$$

On the other hand, we will later show that the optimal acceleration parameter for the pseudo-SOR method ω_{opt} and the corresponding spectral radius λ_{opt} are given as the solution of the following equation with respect to ω and λ:

$$\begin{cases} 1-(1-\frac{1}{2}\cos(\frac{\pi}{N}))\omega-\lambda=-\frac{1}{2}\cos(\frac{\pi}{N})\omega\sqrt{\lambda} \\ \lambda=(1+\frac{1}{2}\cos(\frac{\pi}{N}))\omega-1. \end{cases} \tag{14}$$

The asymptotic estimates are:

$$\omega_{opt}=\frac{4}{3}-\frac{4}{9}(\frac{\pi}{N})^2+\mathcal{O}(\frac{1}{N^4}),\ \lambda_{opt}=1-(\frac{\pi}{N})^2+\mathcal{O}(\frac{1}{N^4}). \tag{15}$$

We note that the optimal spectral radius λ_{opt} approaches unity in $\mathcal{O}(N^{-1})$ in the SOR method, while in $\mathcal{O}(N^{-2})$ in the pseudo-SOR method. The observation for $N=6$ given in the previous subsection also applies to large N.

In order to derive Eq.(14), we will find all the eigenvalues of $\mathcal{L}_5^{p-SOR}$ by solving the secular equation. We note the following equality holds:

$$\begin{aligned} &\det(\mathcal{L}_5^{p-SOR}(\omega)-\lambda I) \\ &=\det((I-\omega L_5^{p-SOR})^{-1}((1-\omega)I+\omega U_5^{p-SOR})-\lambda I) \\ &=(\det(I-\omega L_5^{p-SOR}))^{-1}\det((1-\omega-\lambda)I+\lambda\omega L_5^{p-SOR}+\omega U_5^{p-SOR}). \end{aligned} \tag{16}$$

We have only to solve the equation

$$\det((1-\omega-\lambda)I+\lambda\omega L_5^{p-SOR}+\omega U_5^{p-SOR})=0. \tag{17}$$

The matrix in the parenthesis is a block triangular matrix:

$$\left(\underbrace{\begin{matrix} E & F & & & \mathbf{0} \\ \lambda F & E & F & & \\ & \cdots & \cdots & & \\ & & \cdots & \cdots & \\ \mathbf{0} & & & \lambda F & E \end{matrix}}_{N-1\text{ blocks}}\right),$$

with

$$E = \left(\underbrace{\begin{matrix} 1-\omega-\lambda & \frac{\omega}{4} & & & \mathbf{0} \\ \frac{\omega}{4} & 1-\omega-\lambda & \frac{\omega}{4} & & \\ & \cdots & \cdots & & \\ & & \frac{\omega}{4} & 1-\omega-\lambda & \frac{\omega}{4} \\ \mathbf{0} & & & \frac{\omega}{4} & 1-\omega-\lambda \end{matrix}}_{N-1 \text{ elements}} \right) \text{ and } F = \frac{\omega}{4} I.$$

We will show the following lemma.

Lemma 1 *Let B and C be square matrices of order n, the following identity holds.*

$$\det \left(\underbrace{\begin{matrix} B & C & & & \mathbf{0} \\ \lambda C & B & C & & \\ & \cdots & \cdots & & \\ & & \lambda C & B & C \\ \mathbf{0} & & & \lambda C & B \end{matrix}}_{m \text{ blocks}} \right) = \det \begin{pmatrix} B & \sqrt{\lambda}C & & & \mathbf{0} \\ \sqrt{\lambda}C & B & \sqrt{\lambda}C & & \\ & \cdots & \cdots & & \\ & & \sqrt{\lambda}C & B & \sqrt{\lambda}C \\ \mathbf{0} & & & \sqrt{\lambda}C & B \end{pmatrix}$$

$$= \prod_{j=1}^{m} \det(B + \delta_j \sqrt{\lambda} C), with \delta_j = 2\cos(\frac{\pi j}{m+1}). \quad (18)$$

Proof Since the first equality is self-evident, we will prove the second one. The following matrix of order m with null diagonal elements is diagonalized by an orthogonal matrix P.

$$A = \begin{pmatrix} 0 & 1 & & & \mathbf{0} \\ 1 & 0 & 1 & & \\ & \cdots & \cdots & & \\ & & 1 & 0 & 1 \\ \mathbf{0} & & & 1 & 0 \end{pmatrix} = P \begin{pmatrix} \delta_1 & & & & \mathbf{0} \\ & \delta_2 & & & \\ & & \cdots & & \\ & & & \delta_{m-1} & \\ \mathbf{0} & & & & \delta_m \end{pmatrix} P^{-1}. \quad (19)$$

Replacing each element a_{ij} with a block $a_{ij}\sqrt{\lambda}C$ and adding B to the diagonal blocks, we have the following relation between blocked matrices of order nm.

$$\begin{pmatrix} B & \sqrt{\lambda}C & & & 0 \\ \sqrt{\lambda}C & B & \sqrt{\lambda}C & & \\ & \cdots & \cdots & & \\ & & \cdots & \cdots & \\ 0 & & & \sqrt{\lambda}C & B \end{pmatrix} = \tilde{P} \begin{pmatrix} B+\delta_1\sqrt{\lambda}C & & & 0 \\ & B+\delta_2\sqrt{\lambda}C & & \\ & & \cdots & \\ & & & \cdots \\ 0 & & & B+\delta_m\sqrt{\lambda}C \end{pmatrix} \tilde{P}^{-1}, \tag{20}$$

where $\tilde{P}$ is a matrix of order nm extended from P, where p_{ij} is replaced with a block, $p_{ij}I_n$. Estimating the determinants on both sides, we obtain Eq.(18). Q.E.D.

We apply this lemma to Eq.(17) and find that the equation is equivalent to the system of equations:

$$\det(E + \delta_j\sqrt{\lambda}F) = 0,\ \delta_j = 2\cos(\frac{\pi j}{N})$$

for $j = 1, \ldots, N-1$. Since $E + \delta_j\sqrt{\lambda}F$ is again a tridiagonal matrix, applying this lemma again, we find that Eq.(17) is equivalent to the sysmtem of equations for λ:

$$1 - (1 + \frac{\delta_k}{4})\omega - \lambda = \frac{\delta_j}{4}\sqrt{\lambda}\omega \qquad (j = 1, \ldots, N-1;\ k = 1, \ldots, N-1) \tag{21}$$

The following proposition holds:

Proposition 1 *The absolute values of the solutions of the equation (21) is smaller than or equal to*

- *The largest absolute value of the solutions for* $1 - (1 - \frac{1}{2}\cos(\frac{\pi}{N}))\omega - \lambda = -\frac{1}{2}\cos(\frac{\pi}{N})\sqrt{\lambda}\omega$ *or*
- $\left|1 + \frac{1}{2}\cos(\frac{\pi}{N}))\omega - 1\right|$.

If this proposition is proved, it is straightforward to show that ω_{opt} is given as the solution of Eq.(15).

Proof Eq.(21) can be expressed as the family of quadratic equations for λ,

$$\frac{1 - \alpha\omega - \lambda}{\omega} = \pm\beta\sqrt{\lambda}, \text{with} \alpha > \beta \geq 0. \tag{22}$$

We will first discuss the case for $\beta > 0$. Figure 2 shows that Eq.(22) has two real solutions for $0 < \omega \leq \omega_1$ and two complex solutions for $\omega_1 < \omega$ with the absolute value $\alpha\omega - 1$, where

$\omega_1 = 2/(\alpha + \sqrt{\alpha^2 - \beta^2})$. For real solutions, the larger one monotonically decreases as ω increases and the absolute value of complex solutions increases as ω increases.

It can easily be understood that the absolute value of the complex solution of Eq.(22) is smaller than or equal to $(1 + \frac{1}{4}2\cos(\frac{\pi}{N}))\omega - 1$, where $2\cos(\frac{\pi}{N})$ is the maximum value of δ_k.

Let us consider the real solutions. Figure 3 shows the curves for two sets of equations with the same α and different β, $\beta_2 > \beta_1$. If the equation with β_1 has real solutions, the equation with β_2 has also real solutions whose larger value is greater than that for β_1. Figure 4 shows the curves for two sets of equations with the same β and different α, $\alpha_2 > \alpha_1$. It can easily be seen that, if the equation with α_2 has real solutions, the equation with α_1 has also real solutions whose larger value is greater than that for α_2.

From these observations the absolute value of the real solution for Eq.(22) is always smaller than or equal to the larger solution for Eq.(22) with the smallest α and the largest β. The real solution of Eq.(21) is no greater than the larger solution of

$$1 - (1 - \frac{1}{2}\cos(\frac{\pi}{N}))\omega - \lambda = -\frac{1}{2}\cos(\frac{\pi}{N}\sqrt{\lambda}\omega).$$

This completes the proof for $\beta > 0$.

If $\beta = 0$, Eq.(22) is a linear equation. The same arguments hold if real and complex solutions are understood as positive and negative solutions, respectively. Q.E.D.

The numerical results of ω_{opt} and λ_{opt} are shown in Table. 1.

3 Comparison of SOR and pseudo-SOR Methods for Nine-Point Discretization

In this section we will analyse the SOR and pseudo-SOR method for the nine-point discretization of Poisson equation, Eq.(3). In the SOR method, $U_{i,j}$ is computed with four new values $U_{i-1,j}$, $U_{i-1,j-1}$, $U_{i,j-1}$ and $U_{i,j+1}$, while in the pseudo-SOR method three new values $U_{i-1,j-1}$, $U_{i,j-1}$ and $U_{i,j+1}$ are used, so that we expect the degradation in using pseudo-SOR method may not be so large as in the five-point discretization.

3.1 Matrix Formulation

For the nine-point discretization, the linear equation is described in the matrix form,

$$A_9 u = b, A_9 = \begin{pmatrix} B_9 & C_9 & & & \mathbf{0} \\ C_9 & B_9 & C_9 & & \\ & \cdots & \cdots & & \\ & & C_9 & B_9 & C_9 \\ \mathbf{0} & & & C_9 & B_9 \end{pmatrix}, \tag{23}$$

with

$$B_9 = \begin{pmatrix} 1 & -\frac{1}{5} & & & \mathbf{0} \\ -\frac{1}{5} & 1 & -\frac{1}{5} & & \\ & \cdots & \cdots & & \\ & & \cdots & \cdots & \\ \mathbf{0} & & & -\frac{1}{5} & 1 \end{pmatrix}, C_9 = \begin{pmatrix} -\frac{1}{4} & -\frac{1}{20} & & & \mathbf{0} \\ -\frac{1}{20} & -\frac{1}{5} & -\frac{1}{20} & & \\ & \cdots & \cdots & & \\ & & -\frac{1}{20} & -\frac{1}{5} & -\frac{1}{20} \\ \mathbf{0} & & & -\frac{1}{20} & -\frac{1}{5} \end{pmatrix}. \tag{24}$$

We note that C_9 is not a diagonal matrix in contrast to C_5. The iteration in the SOR method is,

$$u^{(m+1)} = \mathcal{L}_9^{SOR} u^{(m)} + b' = (I - \omega L_9^{SOR})^{-1}((1-\omega)I + \omega U_9^{SOR})u^{(m)} + b' \tag{25}$$

where

$$L_9^{SOR} = \begin{pmatrix} -B_9^L & & & & \mathbf{0} \\ -C_9 & -B_9^L & & & \\ & \cdots & \cdots & & \\ & & -C_9 & -B_9^L & \\ \mathbf{0} & & & -C_9 & -B_9^L \end{pmatrix}, B_9^L = \begin{pmatrix} 0 & & & & \mathbf{0} \\ -\frac{1}{5} & 0 & & & \\ & \cdots & \cdots & & \\ & & \cdots & \cdots & \\ \mathbf{0} & & & -\frac{1}{5} & 0 \end{pmatrix}, \tag{26}$$

and

$$U_9^{SOR} = (L_9^{SOR})^T. \tag{27}$$

On the other hand, the iteration of the pseudo-SOR method for nine-point discretization is formulated in the same way as for five-point discretization,

$$u^{(m+1)} = \mathcal{L}_9^{p-SOR} u^{(m)} + b' = (I - \omega L_9^{p-SOR})^{-1}((1-\omega)I + \omega U_9^{p-SOR})u^{(m)} + b'' \tag{28}$$

where

$$L_9^{p-SOR} = \begin{pmatrix} 0 & & & & \mathbf{0} \\ -C_9 & 0 & & & \\ & \cdots & \cdots & & \\ & & -C_9 & 0 & \\ \mathbf{0} & & & -C_9 & 0 \end{pmatrix}, \tag{29}$$

and

$$U_9^{p-SOR} = \begin{pmatrix} (-B_9^L - B_9^U) & -C_9 & & & \mathbf{0} \\ & (-B_9^L - B_9^U) & -C_9 & & \\ & \cdots & \cdots & & \\ & & & (-B_9^L - B_9^U) & -C_9 \\ \mathbf{0} & & & & (-B_9^L - B_9^U) \end{pmatrix}. \tag{30}$$

3.2 Acceleration Parameter ω and the Spectral Radius

First we estimated numerically the spectral radia of $\mathcal{L}_9^{SOR}(\omega)$ and $\mathcal{L}_9^{p-SOR}(\omega)$ for $N = 9$. The results are shown in Fig. 5. The same statements holds as in Section 2.2. Although three new values are used to update $U_{i,j}$, the pseudo-SOR method is considerably inefficient. We will discuss the convergence of these two methods for general N.

3.3 Analytic Determination of the Optimal Acceleration Parameter of SOR and pseudo-SOR Methods

In a similar manner to the five-point case, we have to solve $\det((1-\omega-\lambda)I + \lambda\omega L_9^{SOR} + \omega U_9^{SOR}) = 0$, to obtain the eigenvalues of $\mathcal{L}_9^{SOR}$. Using Lemma 1 twice, the problem is reduced to a set of equations,

$$1 - \omega - \lambda - \delta_j(\omega/5)\sqrt{\lambda} - \delta_k(\omega/5)\sqrt{(\lambda - (\delta_j/4)\sqrt{\lambda})(1 - (\delta_j/4)\sqrt{\lambda})} = 0, \tag{31}$$

for $j = 1, \ldots, N-1, k = 1, \ldots, N-1$. Based on our numerical experience, we conjecture that the optimal acceleration parameter ω_{opt} is given by the ω for which the largest real solution coincides with the absolute value of the complex solution of the quartic equation with respect to λ,

$$(1-\omega-\lambda)^4 - 2\delta^2(8+\delta^2)(\frac{\omega}{5})^2\lambda(1-\omega-\lambda)^2 + \delta^8(\frac{\omega}{5})^4 +$$

$$16\delta^4(\frac{\omega}{5})^3\lambda(\lambda+1)(1-\omega-\lambda)-4\delta^6(\frac{\omega}{5}\lambda(\lambda+1)^2=0. \tag{32}$$

Assuming this conjecture, we calculated ω_{opt} and λ_{opt} numerically for $N \leq 1000$ and found an approximate formula:

$$\omega_{opt} \approx 2-\frac{6.63}{N},\ \lambda_{opt} \approx 1-\frac{5.61}{N}. \tag{33}$$

On the other hand, in a similar manner to the five-point case, the optimal acceleration parameter and the corresponding spectral radius for the pseudo-SOR method are determined in terms of the set of equation:

$$\begin{aligned} &1-(1-\frac{2}{5}\cos(\frac{\pi}{N}))\omega-\lambda=-\frac{\omega}{5}\cos(\frac{\pi}{N})(2+\cos(\frac{\pi}{N}))\sqrt{\lambda} \\ &\lambda=(1-\frac{2}{5}\cos(\frac{\pi}{N}))\omega-1. \end{aligned} \tag{34}$$

Asymptotically,

$$\omega_{opt}=\frac{10}{7}-\frac{85}{98}(\frac{\pi}{N})^2+\mathcal{O}(\frac{1}{N^4}),\ \lambda_{opt}=1-\frac{3}{2}(\frac{\pi}{N})^2+\mathcal{O}(\frac{1}{N^4}). \tag{35}$$

Table 2 shows the numerical values for ω_{opt} and λ_{opt}. We can see that the optimal acceleration parameter for the pseudo-SOR method is substantially smaller than that for the SOR method and the rate of convergence of the pseudo-SOR method using the optimal acceleration parameter is worse than that of the SOR method.

4 Numerical Results on Vector Processors

We have seen that the pseudo-SOR method is inferior to the original SOR method in terms of the convergence ratio for both five-point and nine-point discretization, so that the former method requires more number of iterations than the latter. On the other hand, the pseudo-SOR method is very suited to vector supercomputers, since it involves only stride one access and has longer vector length than the hyperplane SOR. We will implement those method on various supercomputers and compare the CPU time for solving the problem.

4.1 Dirichlet Problem on a Rectangular Mesh

As an example, we will consider a Dirichlet problem on the unit square with the following boundary conditions:

$$
\begin{aligned}
&U(0,y) = 0, \quad U(1,y) = 0 && (0 < y < 1) \\
&U(x,0) = 0, \quad U(x,1) = 0.5 - |x - 0.5|\,. && (0 < x < 1)
\end{aligned}
$$

4.1.1 Numerical Verification of ω_{opt} for $N = 6$

As a numerical test, we discretized the square by six in one direction ($N = 6$) and solved the equation. We show in Figs.6–9 the behavior of convergence for SOR method and the pseudo-SOR method for both five-point and nine-point discretizations in terms of the L_1-norm of the residual vector. The initial vector $u^{(0)}$ is a zero vector. The calculations were performed in the double precision. Figures (a) are for $\omega < \omega_{opt}$ and (b) for $\omega > \omega_{opt}$. The theoretical optimal acceleration parameters ω_{opt} for $N = 6$ are given in Table 1. Those figures show that the predicted ω_{opt} is actually optimal.

4.1.2 Performance on Vector Computers

In order to make a realistic evaluation, we have tested the same problem with more grip points and measured the performance on various vector supercomputers. We only applyed the five-point discretization. We tested ten cases with $N = 45, 63, 77, 89, 99, 109, 119, 127, 135, 141$. The acceleration parameter is fixed on the theoretical optimal value. The initial vector is null and the computations were done in the double precision. The convergence criterion is that the L_2-norm of the residual is less than 10^{-6}. The computers used are,

Vendor	Model	Max. Speed
a) Fujitsu	VP-200	570 MFLOPS
b) Fujitsu	VP-400E	1700 MFLOPS
c) Hitachi	S-820/80	2000 MFLOPS
d) NEC	SX-2	1300 MFLOPS
e) Convex	C-1 XP	20 MFLOPS
f) Ardent	TITAN	16 MFLOPS
g) Stellar	GS1000	40 MFLOPS

The CPU time spent till the convergence is shown in Fig.10 as a function of the number of grid points. The hyperplane vectorized SOR is superior to the pseudo-SOR for all the supercomputers tested here.

4.2 Dirichret Problem on a Curved Mesh

In general curved coordinate (ζ, η), the Laplacian contains a cross term $\partial^2 u/\partial\zeta\partial\eta$, which is discretized in terms of $(U_{i+1,j+1} + U_{i-1,j-1} - U_{i+1,j-1} - U_{i-1,j+1})$, so that the discretized equation is a nine-point difference formula in general [4]. We test here two cases, a) where the numbers of grid points in two dimensions are the same, and b) where the number of grid points in the horizontal direction is larger than the one in the vertical direction.

a) The domain is shown in Fig.11. The boundary values are 20 on the top, 50 on the bottom and have interpolated values on the right and left sides. The grid points are $46^2, 62^2, 78^2, 90^2, 102^2, 110^2, 118^2, 126^2, 134^2, 142^2$. The initial value is all 0 and the convergence condition is that the L_2-norm of the residual vector be less than 10^{-6}. Since the optimal acceleration parameter is not known theoretically, we searched the optimal one in step 0.02. The CPU time for the SOR and pseudo-SOR methods is shown in Fig.12. The hyperplane vectorized SOR method is much faster than the pseudo-SOR method even in the nine-point discretization.

b) In this case a horizontally long domain with curved boundaries is treated(Fig.13). We tested three types of grids, $40\times82, 40\times162, 40\times322$. Other conditions are same as in a). Table 3 shows the CPU time for VP-200 and S-820/80. Although a long vector length is favorable for the pseudo-SOR method, the result in this table indicates, the larger the number of grid points, the faster the hyperplane SOR method as compared to the pseudo-SOR method.

4.3 Mixed Boundary Problem on a Circular Mesh

In this subsection we will discuss Poisson equation

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = -2\sin(x)\sin(y)$$

in regions bounded by two eccentric circles, as shown in Fig.14. We tested six kinds of grid with different eccentricity and concentricity [5]. Such grids are used in calculating the stream around a circular cylinder. Two kinds of boundary conditions are assumed: a) Dirichlet contition on all boundaries, and b) Dirichlet on the inner boundary and Neumann on the outer boundary. The size of the grid is 16(r-direction) $\times$ 50(θ-direction). The analytic solution is $u = \sin(x)\sin(y)$.

The optimal acceleration parameter and the corresponding spectral radius are searched numerically in step 0.02. The results are shown in Table 4 for the six kinds of grid in Fig.14. The optimal acceleration parameters in the hyperplane SOR method are stable for both Dirichet and mixed conditions with respect to the eccentricity, while those the pseudo-SOR method changes appreciably. In the case of hyperplane SOR method, the optimal acceleration parameter for the mixed condition is larger by 0.12–0.18 than that for Dirichlet condition, while in the case of the pseudo-SOR method, there is no difference.

Moreover, the spectral radius for the optimal acceleration parameter for the mixed condition is larger than that for Dirichlet condition. Especially, the spectral radius for the pseudo-SOR method is nearly 1.0, the convergence limit. Fig.15 shows the numbers of iterations for different acceleration parameters. It is to be noted that the pseudo-SOR method is very critical, in the sense that an acceleration parameter slightly larger than the optimal value may cause divergence of the iteration. On the other hand, the convergence region for the hyperplane SOR method is large.

5 Conclusion

We have analysed both theoretically and numerically the performance of the genuine SOR and pseudo-SOR. Our observation is as follows.

1. For the rectangular region, in both five-point and nine-point discretization, the asymptotic convergence rate is $1 - C/N$ for the SOR method, while $1 - C'/N^2$ for the pseudo-SOR method. Although the latter is easily vectorizable, the overall efficiency of the former is higher.

2. Even when the aspect ratio of the grid is large, which case is favorable for the pseudo-SOR method, this method is inferior to the hyperplane SOR method.

3. In the curved mesh discussed here,

(a) As the eccentricity increases, the optimal acceleration parameter becomes small and the spectral radius approches unity in the pseudo-SOR method. On the other hand, the optimal acceleration parameter in the hyperplane SOR does not change appreciably.

(b) In the pseudo-SOR method, acceleration parameter slightly larger than the optimal value may cause divergence.

(c) When the outer boundary is with Neumann condition, the spectral radius for the pseudo-SOR method is nearly unity.

We conclude the pseudo-SOR method is less than the genuin SOR method in efficiency on vector supercomputers, although the former is easily vectorizable with unit stride. The rate of convergence should be most respected in the vectorization of an algorithm.

Acknowledgement

This work is partly supported by the Grand-in-Aid for General Scientific Research of The Ministry of Education, Science and Culture of the Japanese Government (No.01540169).

References

[1] J. M. Ortega, "Introduction to Parallel and Vector solution of Linear Systems", Plenum Press, New York and London, 1988.

[2] L. Lamport, "The Parallel Execution of `DO` Loops", Comm. of ACM, **17**, pp.83-93, 1974.

[3] R. S. Varga, "Matrix Iterative Analysis", Prentice-Hall, New York, 1962.

[4] J. F. Thompson, Z. U. A. Warsi, "Numerical Grid Generation", North Holland, 1974.

[5] T. Kawamura, H. Takami, K. Kuwahara, "Computaion of High Reynolds Number Flow around a Circular Cylinder with Surface Roughness", Fluid Dynamics Research, 1, pp.145–162, 1986.

Table 1 ω_{opt} and λ_{opt} for five-point discretization

	SOR method		pseudo-SOR method	
	ω_{opt}	λ_{opt}	ω_{opt}	λ_{opt}
$N = 6$	1.33333	0.33333	1.23431	0.76878
$N = 10$	1.52786	0.52786	1.29285	0.90764
$N = 20$	1.72945	0.72945	1.32259	0.97584
$N = 50$	1.88183	0.88183	1.33158	0.99606
$N = 100$	1.93909	0.93909	1.33289	0.99901
$N = 1000$	1.99373	0.99373	1.33332	0.99999

Table 2 ω_{opt} and λ_{opt} for nine-point discretization

	SOR method		pseudo-SOR method	
	ω_{opt}	λ_{opt}	ω_{opt}	λ_{opt}
$N = 6$	1.31393	0.37071	1.26184	0.69896
$N = 10$	1.50902	0.56335	1.35459	0.86991
$N = 20$	1.71627	0.75377	1.40799	0.96425
$N = 50$	1.87542	0.89351	1.42517	0.99411
$N = 100$	1.93567	0.94529	1.42772	0.99852
$N = 1000$	1.99337	0.99439	1.42856	0.99999

Table 3 CPU time in sec for the optimal acceleration

grid	40×82	40×162	40×322
pseudo-SOR	0.49	2.55	17.2
hyperplane SOR	0.27	0.72	3.29

(VP-200)

grid	40×82	40×162	40×322
pseudo-SOR	0.17	0.82	9.08
hyperplane SOR	0.08	0.27	1.69

(S-820/80)

Table 4 ω_{opt} and ρ_{opt} for an eccentric double circle

mesh	pseudo-SOR		hyperplane SOR	
	ω_{opt}	ρ_{opt}	ω_{opt}	ρ_{opt}
Fig.14 (a)	1.22	0.971	1.68	0.720
Fig.14 (b)	1.12	0.971	1.70	0.717
Fig.14 (c)	1.00	0.977	1.70	0.782
Fig.14 (d)	1.20	0.970	1.68	0.720
Fig.14 (e)	1.14	0.975	1.72	0.766
Fig.14 (f)	1.02	0.975	1.72	0.766

a) Dirichlet condition on both boundaries

mesh	pseudo-SOR		hyperplane SOR	
	ω_{opt}	ρ_{opt}	ω_{opt}	ρ_{opt}
Fig.14 (a)	1.26	0.991	1.84	0.840
Fig.14 (b)	1.16	0.993	1.88	0.856
Fig.14 (c)	1.02	0.994	1.82	0.953
Fig.14 (d)	1.26	0.998	1.88	0.900
Fig.14 (e)	1.16	0.998	1.88	0.913
Fig.14 (f)	1.00	0.997	1.80	0.924

b) mixed boundary conditions

Figure Captions

1. Spectral radii of $\mathcal{L}_5^{SOR}(\omega)$ and $\mathcal{L}_5^{p-SOR}(\omega)$.

2. Parabola $\pm\beta\sqrt{\lambda}$ and lines $(1-\alpha\omega-\lambda)/\omega)$ with $\omega < \omega_1$, $\omega = \omega_1$ and $\omega > \omega_1$.

3. Line $(1-\alpha\omega-\lambda)/\omega)$ and parabolas $\pm\beta\sqrt{\lambda}$ for $(\beta_1 < \beta_2)$.

4. Parabolas $\pm\beta_0\sqrt{\lambda}$ and lines $(1-\alpha\omega-\lambda)/\omega)$ for $(\alpha_1 < \alpha_2)$.

5. Spectral radii of $\mathcal{L}_9^{SOR}(\omega)$ and $\mathcal{L}_9^{p-SOR}(\omega)$.

6. Convergence behavior of the SOR method with five-point discretization.

7. Convergence behavior of the pseudo-SOR method with five-point discretization.

8. Convergence behavior of the SOR method with nine-point discretization.

9. Convergence behavior of the pseudo-SOR method with nine-point discretization.

10. Comparison of the CPU time on several vector computers (five-point discretization). The horizontal axis is the number of grid points and the vertical axis is the CPU time in seconds.

11. The domain for analysis.

12. Comparison of the CPU time on several vector computers (nine-point discretization in curved mesh).

13. The domain for analysis.

14. The grid systems of eccentric double circles.

15. The number of iterations needed for convergence as a function of the acceleration parameter for (I) pseudo-SOR and (II) hyperplane SOR method. The boundary condition is Dirichlet and the grid systems are (a), (b) and (c) in Fig. 14.

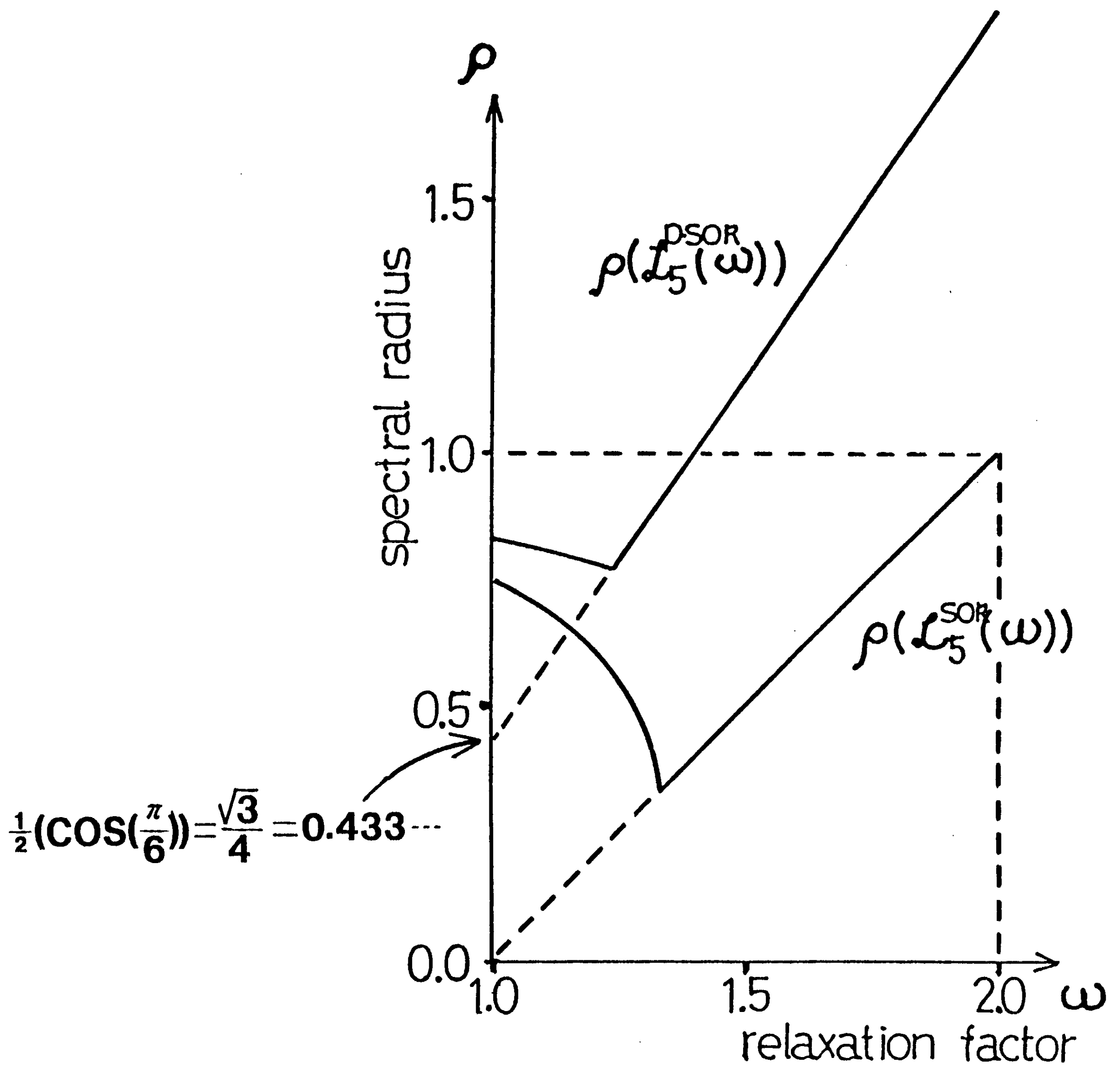
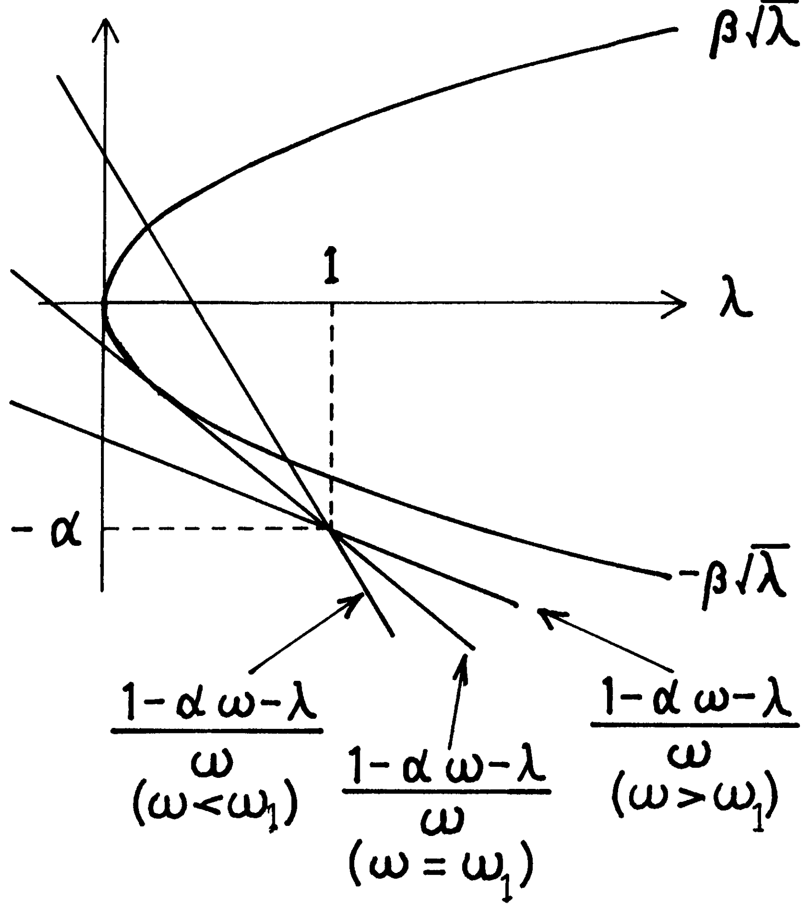
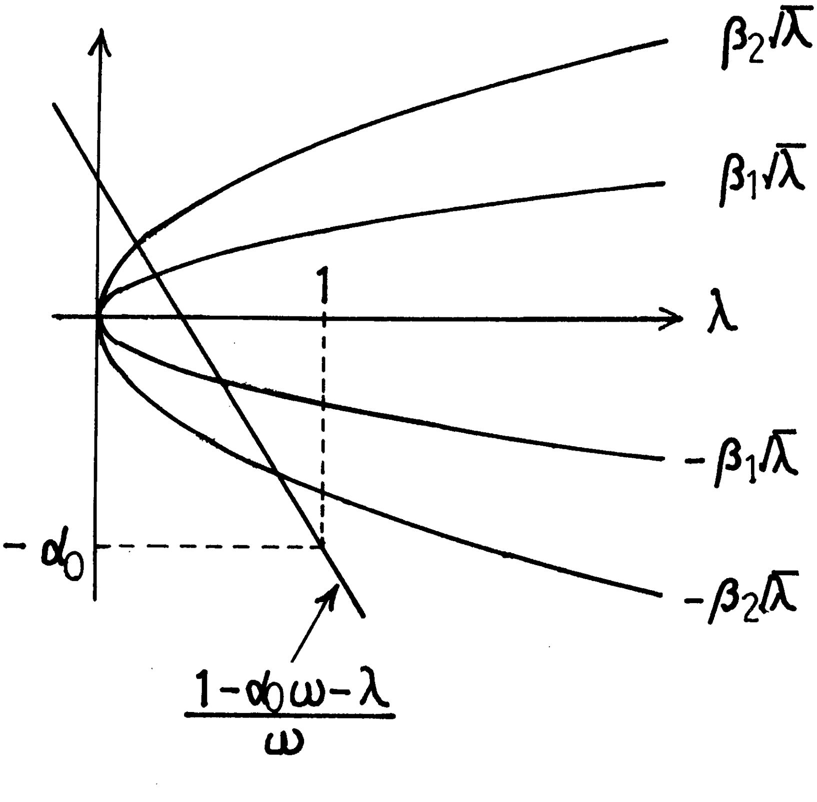

ρ
spectral radius
1.5
1.0
0.5
0.0
$\rho(\mathcal{L}_5^{DSOR}(\omega))$
$\rho(\mathcal{L}_5^{SOR}(\omega))$
$\frac{1}{2}(\cos(\frac{\pi}{6}))=\frac{\sqrt{3}}{4}=0.433\cdots$
1.0
1.5
2.0
ω
relaxation factor

Fig. 1



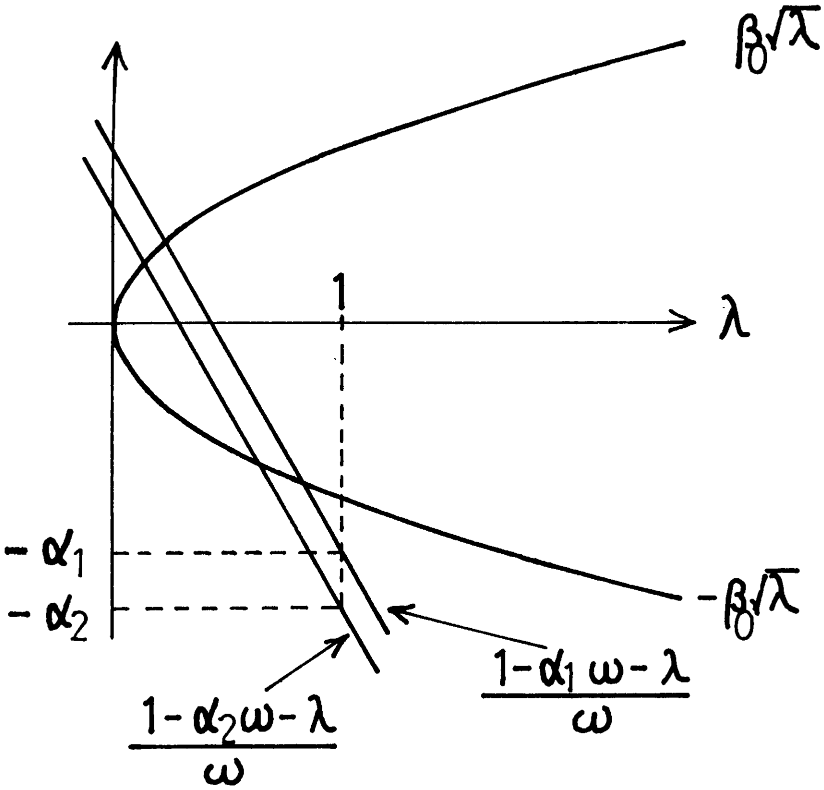
$\beta\sqrt{\lambda}$
1
λ
$-\alpha$
$-\beta\sqrt{\lambda}$
$\frac{1-\alpha\omega-\lambda}{\omega}$
$(\omega<\omega_1)$
$\frac{1-\alpha\omega-\lambda}{\omega}$
$(\omega=\omega_1)$
$\frac{1-\alpha\omega-\lambda}{\omega}$
$(\omega>\omega_1)$

F i g . 2



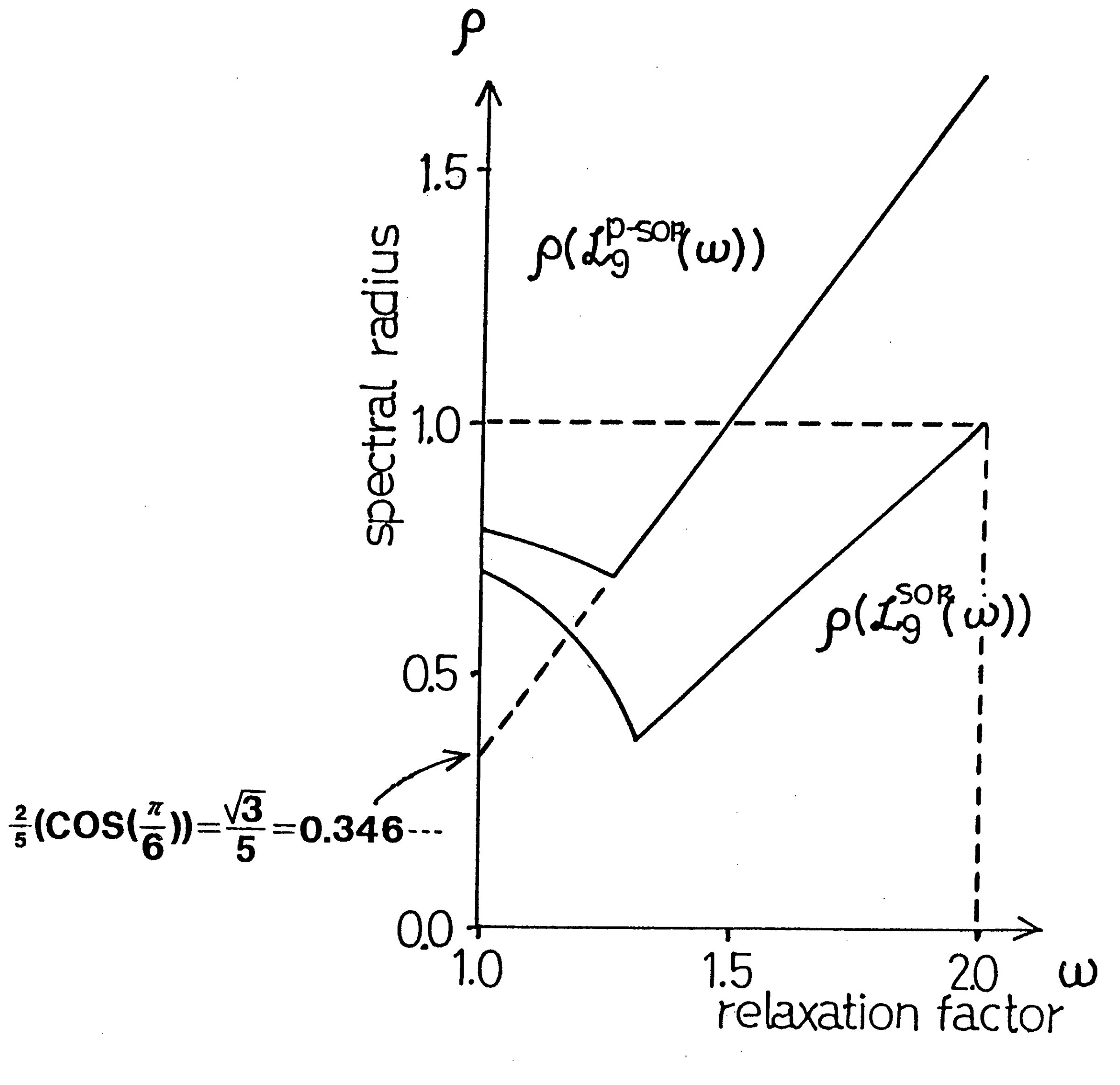
$\beta_2\sqrt{\lambda}$
$\beta_1\sqrt{\lambda}$
1
λ
$-\beta_1\sqrt{\lambda}$
$-\alpha_0$
$-\beta_2\sqrt{\lambda}$
$\frac{1-\alpha_0\omega-\lambda}{\omega}$

F i g . 3



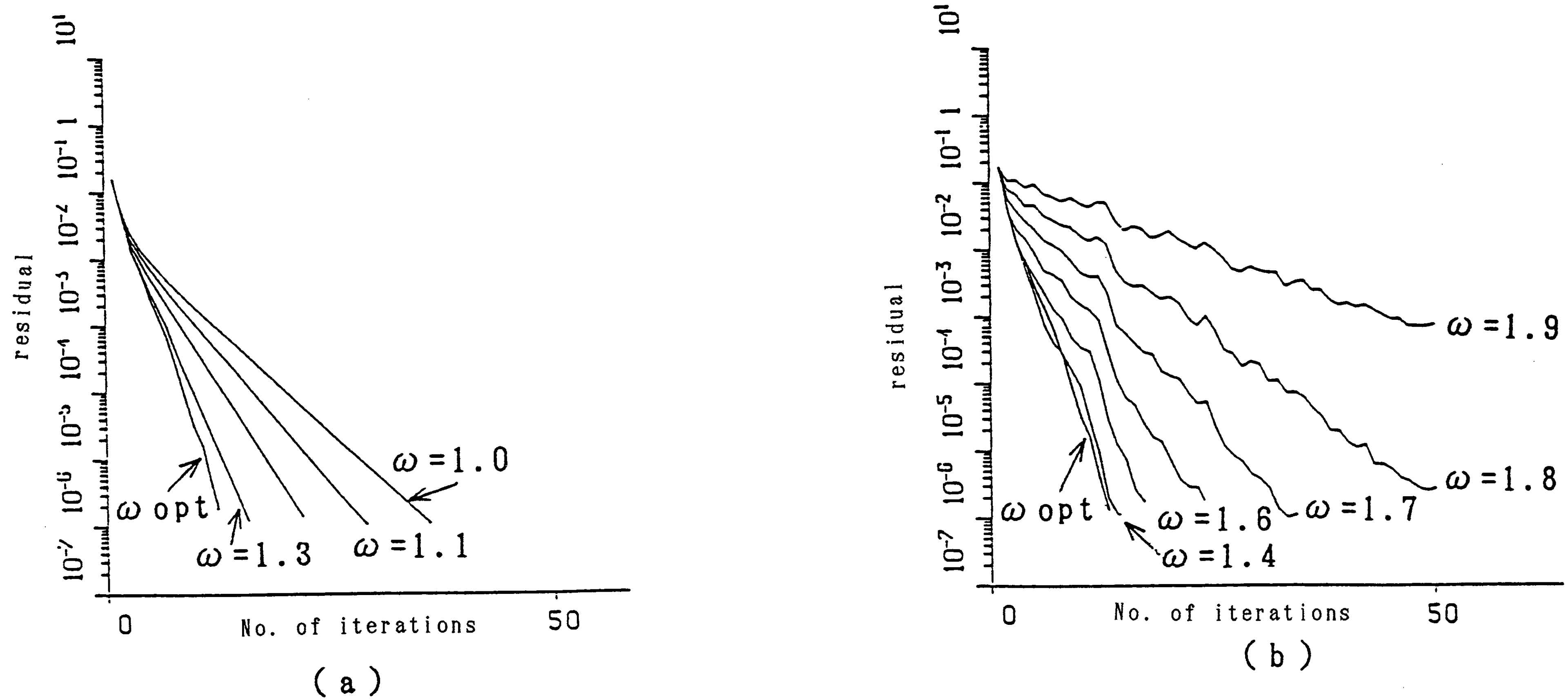
$\beta_0\sqrt{\lambda}$
1
λ
$-\alpha_1$
$-\alpha_2$
$-\beta_0\sqrt{\lambda}$
$\frac{1-\alpha_1\omega-\lambda}{\omega}$
$\frac{1-\alpha_2\omega-\lambda}{\omega}$

F i g . 4



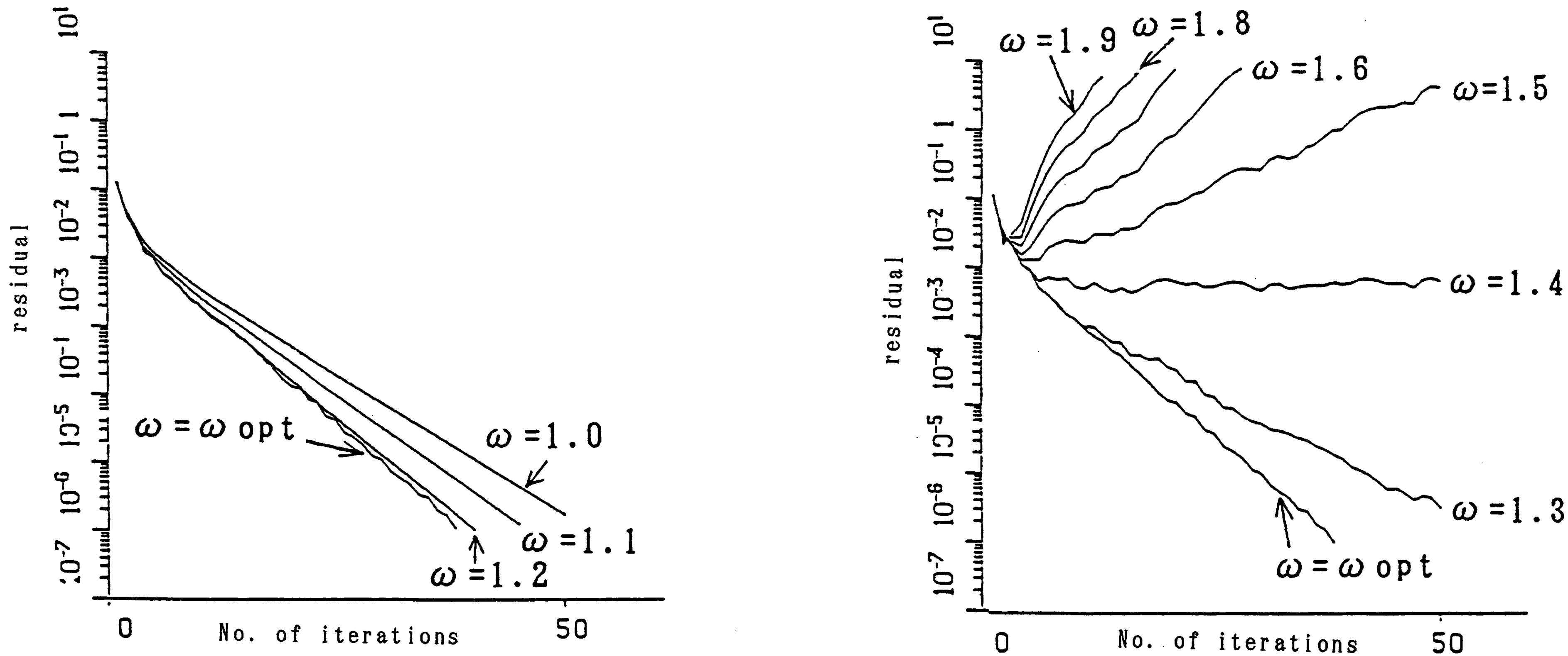
ρ
spectral radius
1.5
1.0
0.5
0.0
$\rho(\mathcal{L}_9^{\text{p-SOR}}(\omega))$
$\rho(\mathcal{L}_9^{\text{SOR}}(\omega))$
$\frac{2}{5}(\cos(\frac{\pi}{6}))=\frac{\sqrt{3}}{5}=0.346\cdots$
1.0
1.5
2.0
ω
relaxation factor

F i g . 5



residual
10^1
1
10^-1
10^-2
10^-3
10^-4
10^-5
10^-6
10^-7
No. of iterations
0
50
ω=1.0
ω opt
ω=1.3
ω=1.1
(a)
ω=1.9
ω=1.8
ω=1.7
ω=1.6
ω=1.4
(b)

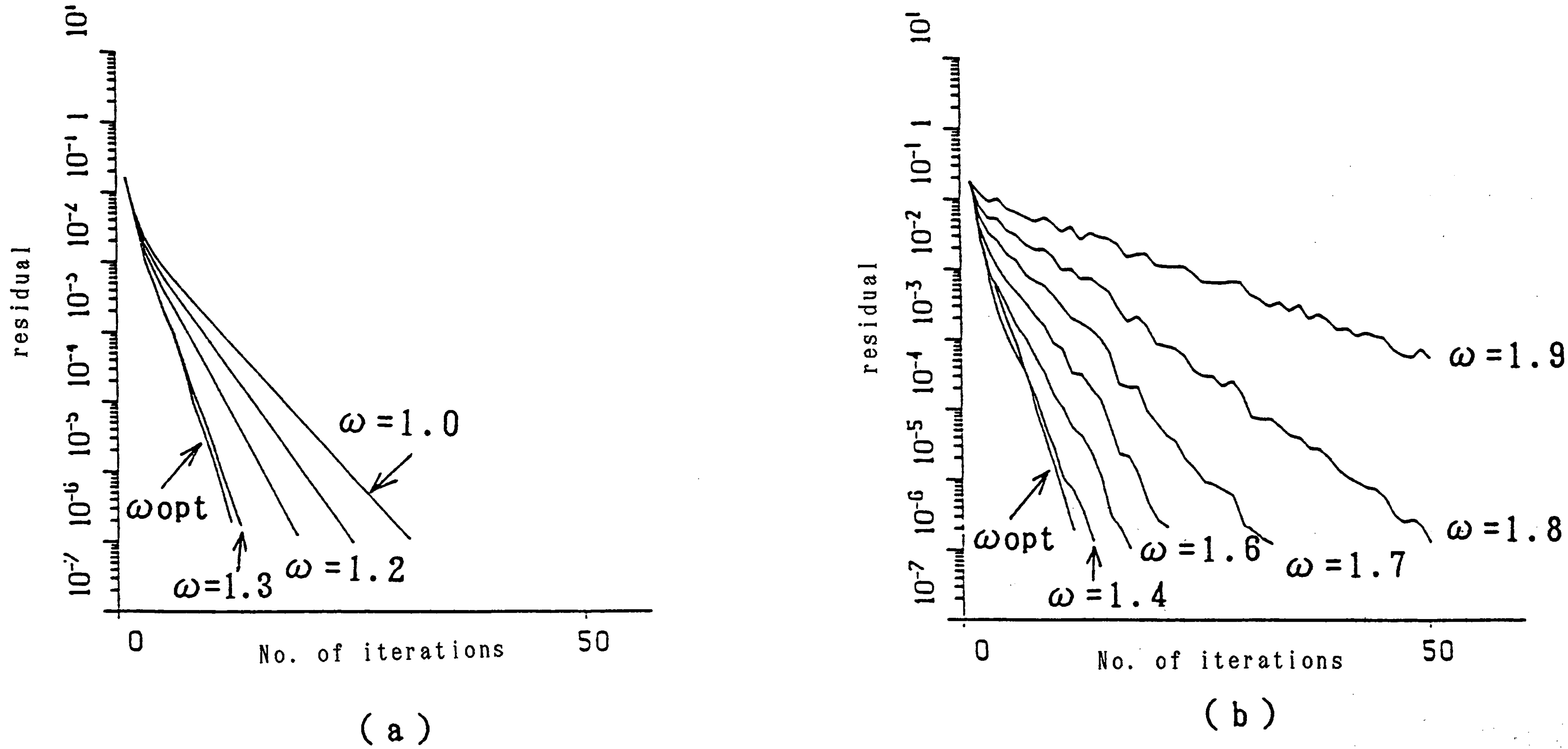
F i g . 6



residual
10^1
1
10^-1
10^-2
10^-3
10^-4
10^-5
10^-6
10^-7
0
No. of iterations
50
ω = ω opt
ω =1.0
ω =1.1
ω =1.2
ω =1.9
ω =1.8
ω =1.6
ω=1.5
ω =1.4
ω =1.3

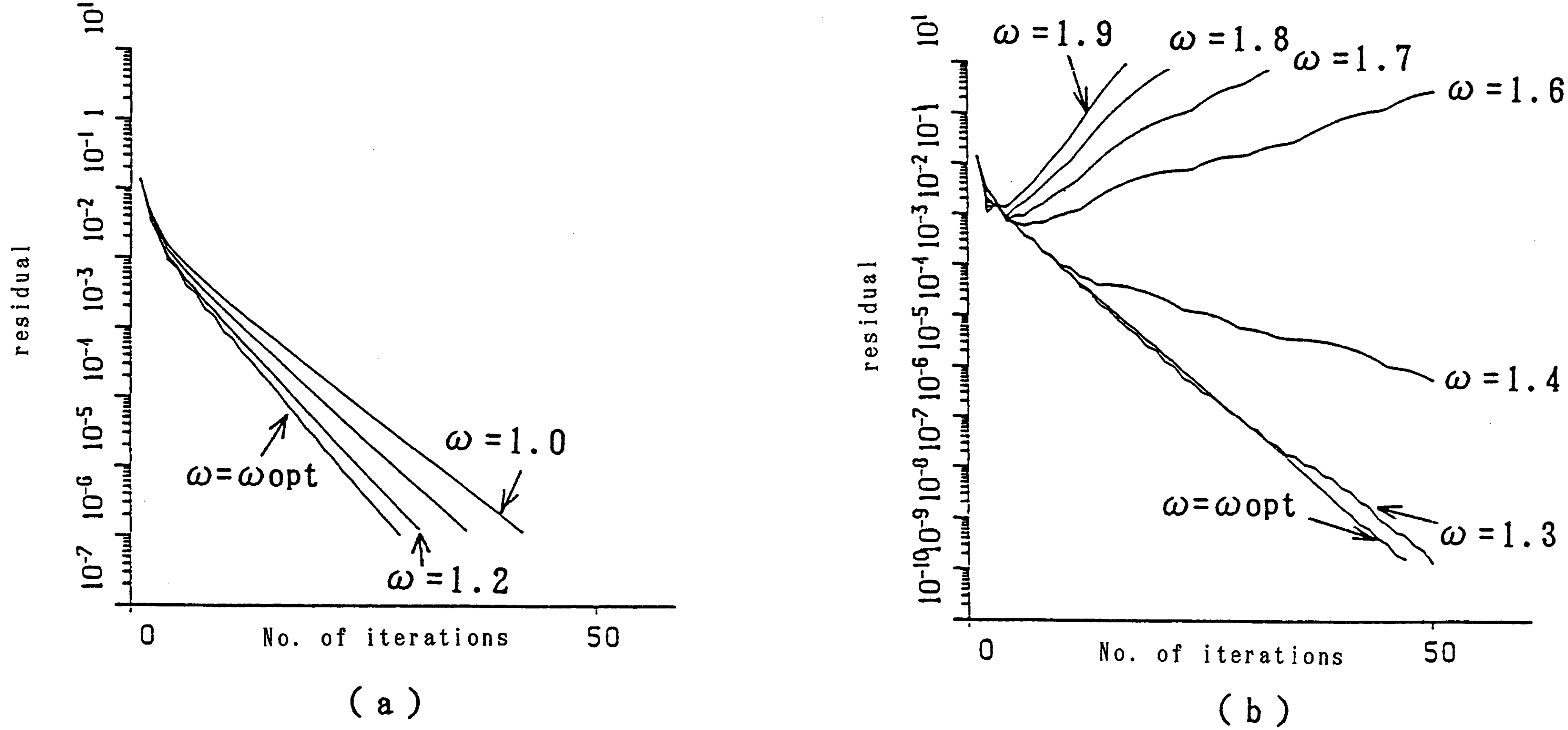
(a) (b)

Fig. 7



residual
10¹
1
10⁻¹
10⁻²
10⁻³
10⁻⁴
10⁻⁵
10⁻⁶
10⁻⁷
0
No. of iterations
50
ω=1.0
ωopt
ω=1.3
ω=1.2
(a)
ω=1.9
ω=1.8
ω=1.7
ω=1.6
ω=1.4
(b)

Fig. 8



residual
10^1
1
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
0
No. of iterations
50
ω=1.0
ω=ωopt
ω=1.2
(a)
residual
10^1
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-8}
10^{-9}
10^{-10}
0
No. of iterations
50
ω=1.9
ω=1.8
ω=1.7
ω=1.6
ω=1.4
ω=ωopt
ω=1.3
(b)

Fig. 9

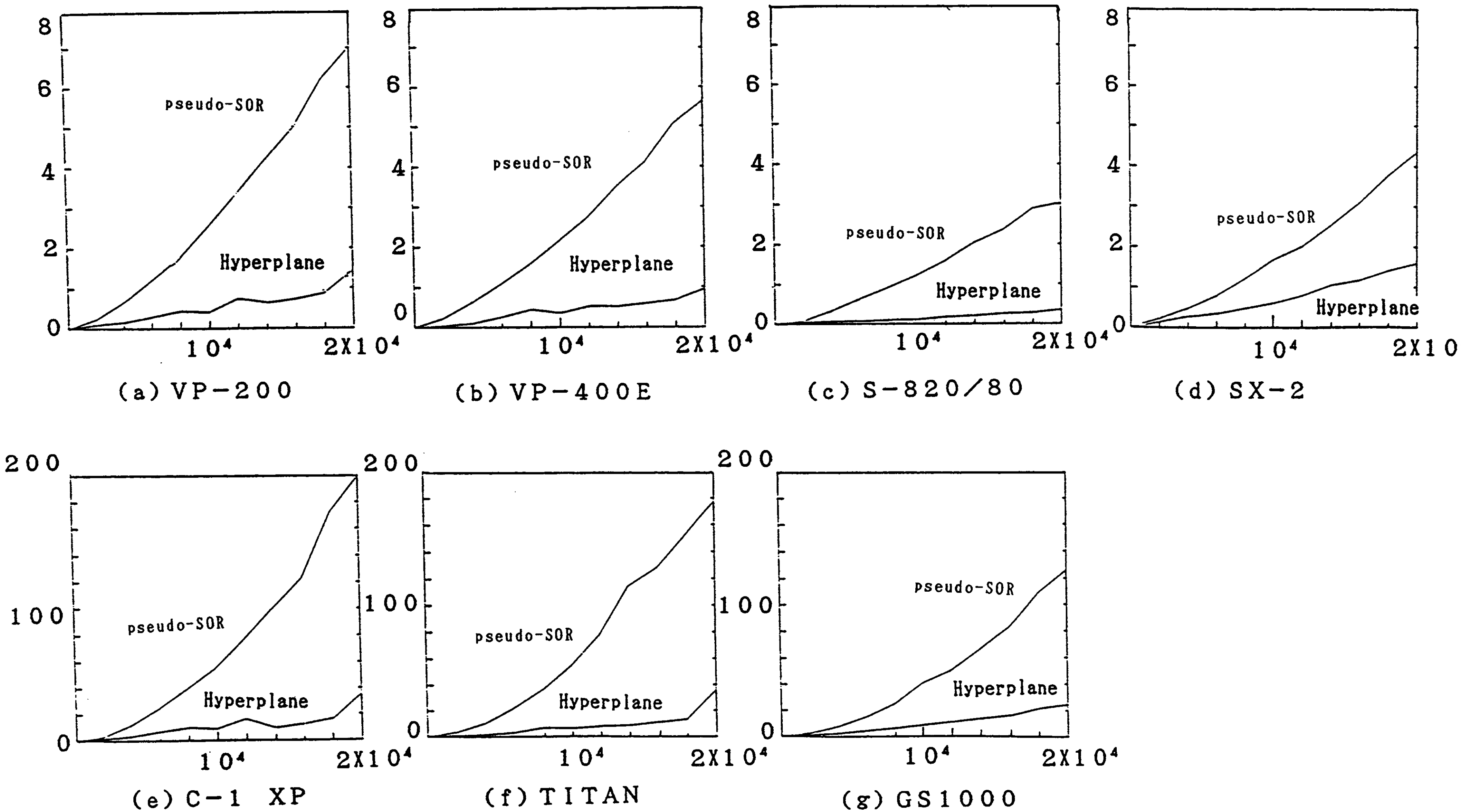

pseudo-SOR
Hyperplane
0
2
4
6
8
10^4
$2X10^4$
(a) VP−200
(b) VP−400E
(c) S−820/80
(d) SX−2
0
100
200
(e) C−1 XP
(f) TITAN
(g) GS1000

Fig. 10

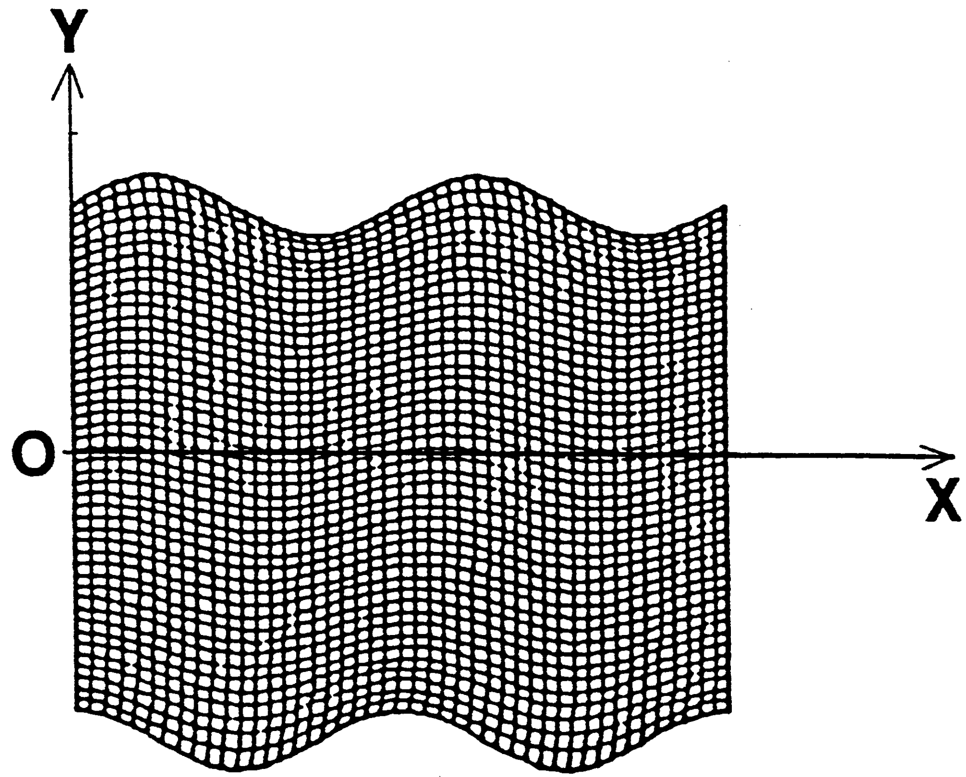
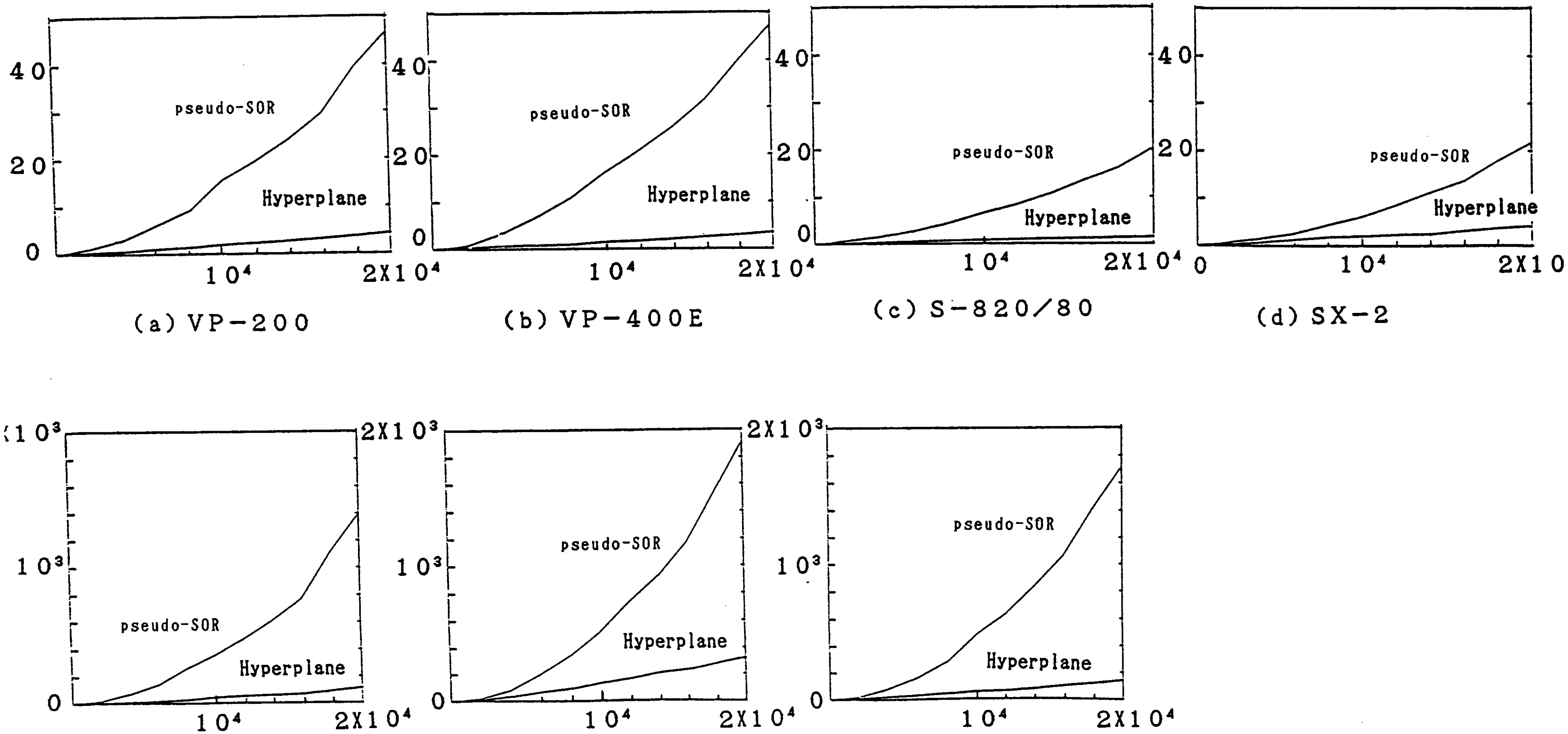


Fig. 11



(a) VP−200 (b) VP−400E (c) S−820/80 (d) SX−2

(e) C−1 XP (f) TITAN (g) GS1000

F i g . 1 2

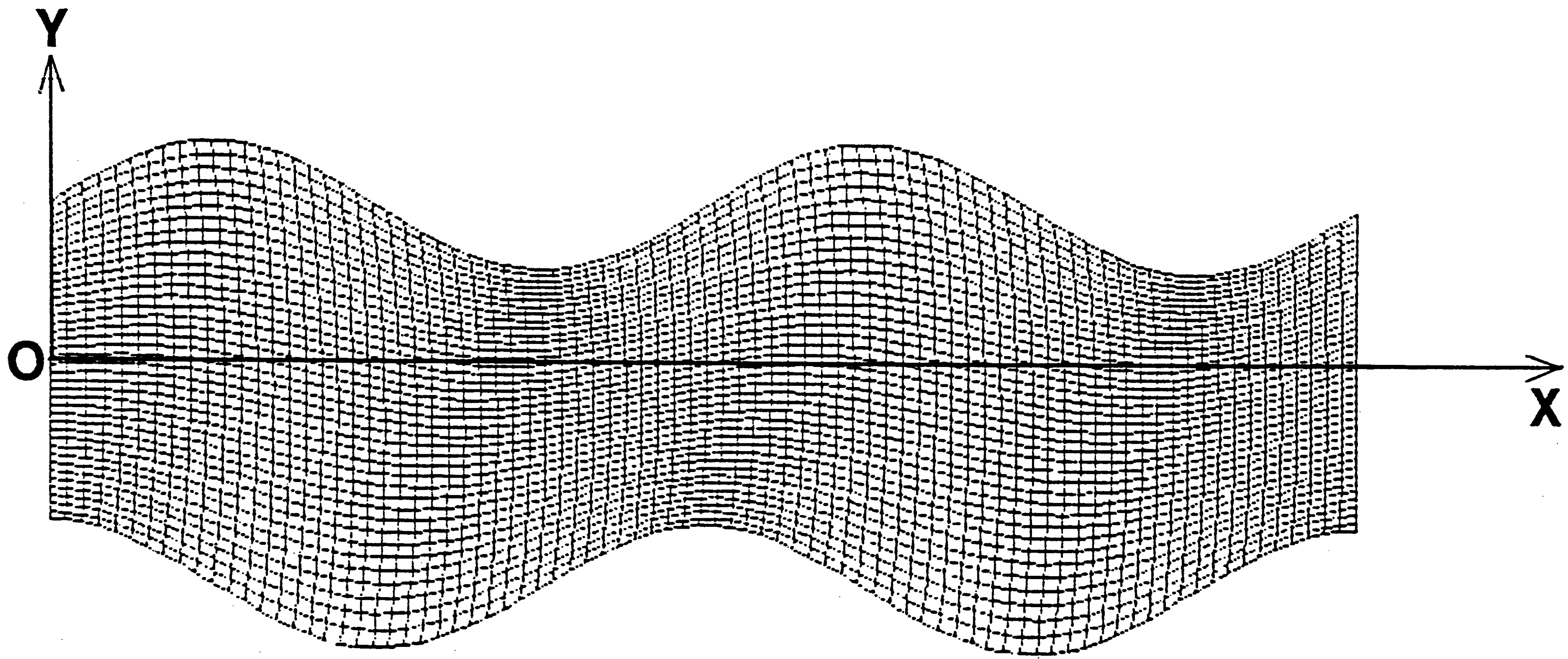


Fig. 13

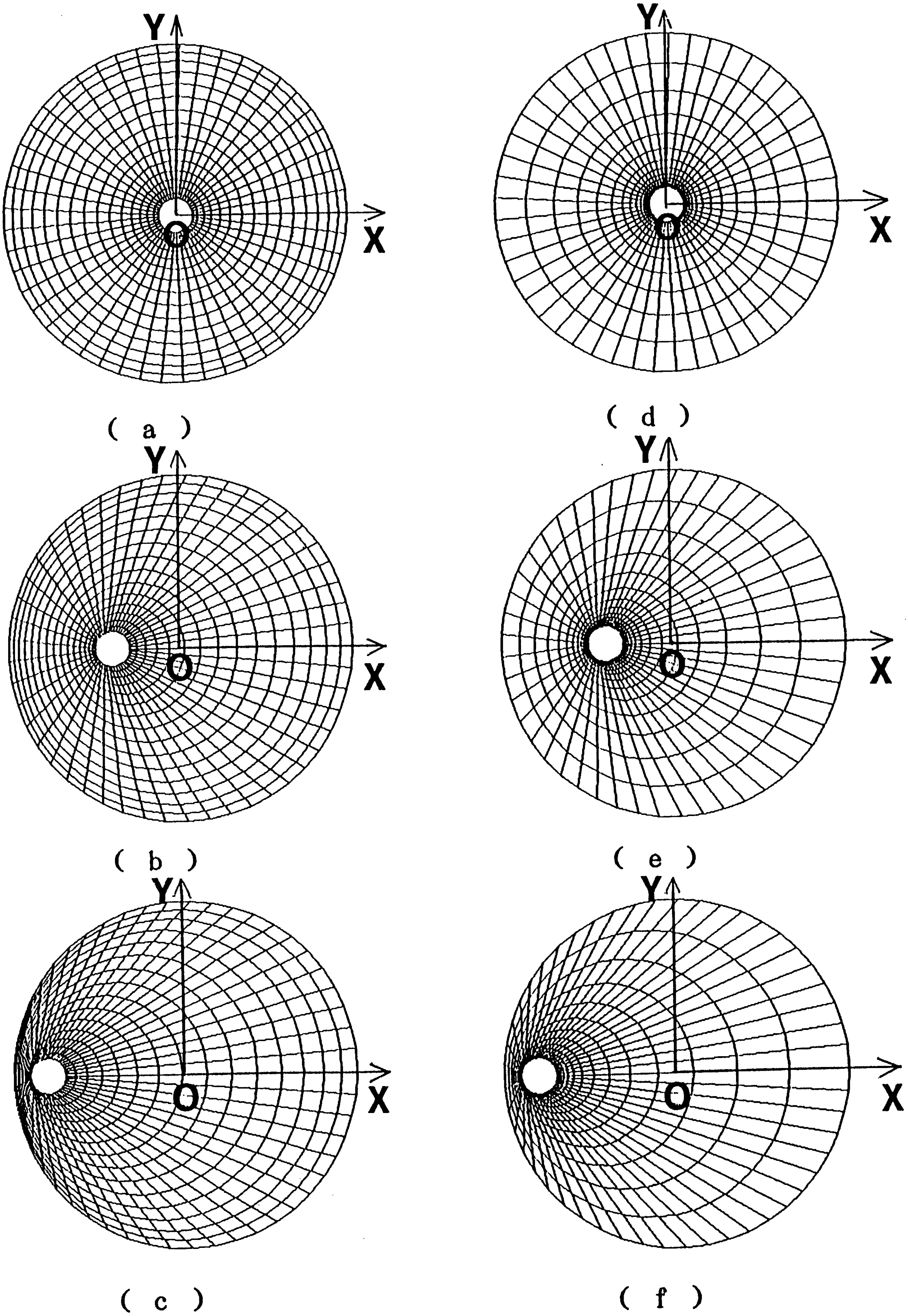


Fig. 14

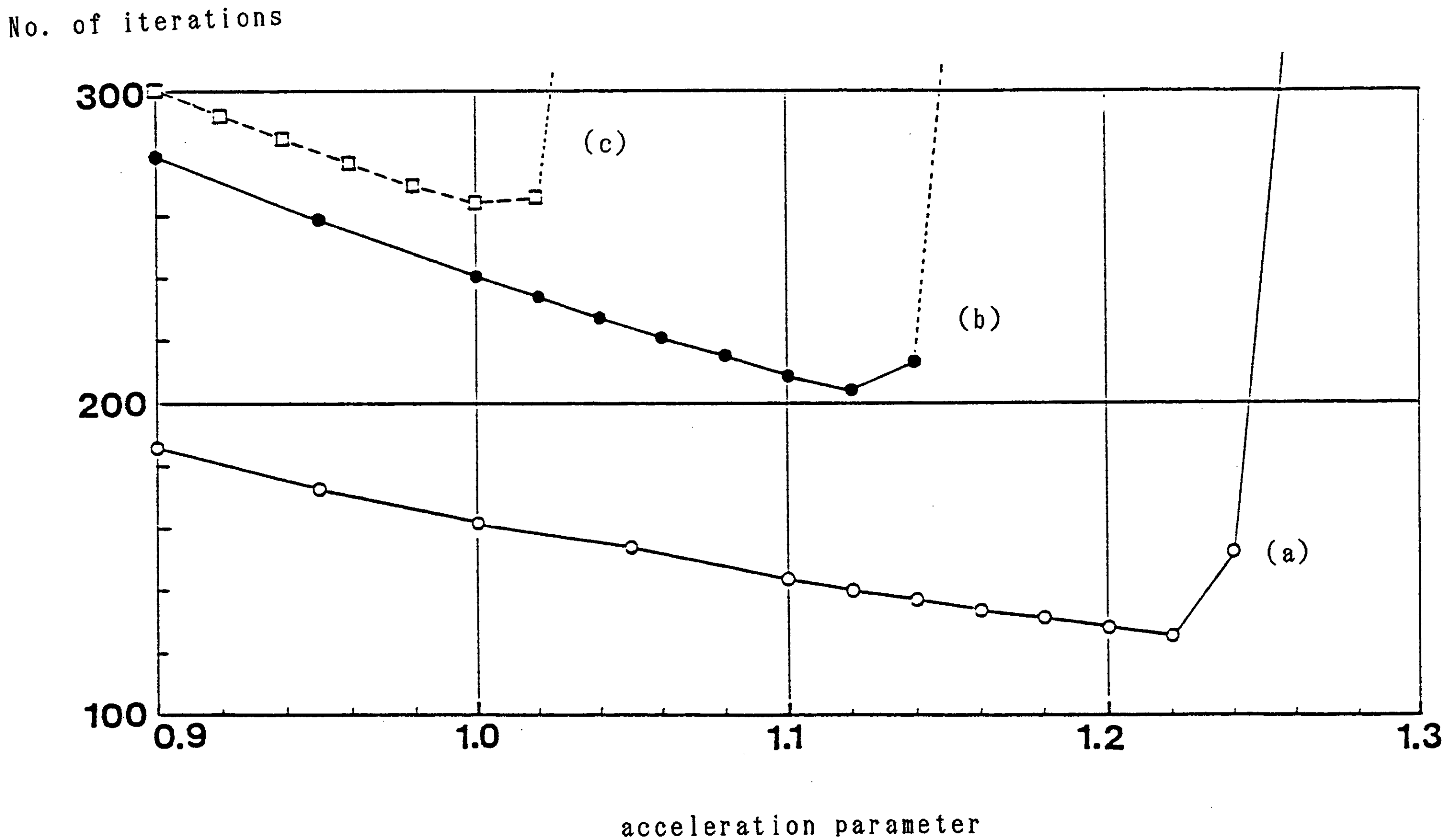

No. of iterations
300
200
100
0.9
1.0
1.1
1.2
1.3
(c)
(b)
(a)
acceleration parameter

Fig. 15 (I)

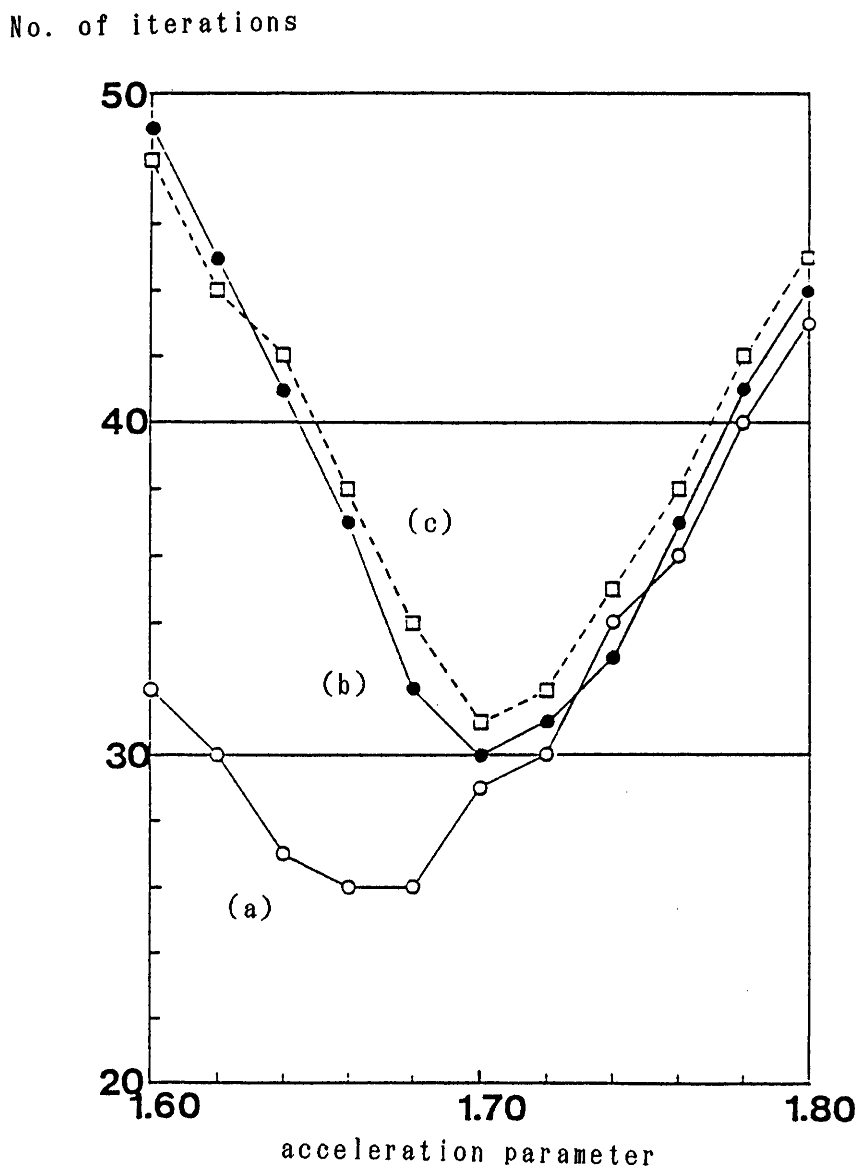

No. of iterations
50
40
30
20
1.60
1.70
1.80
acceleration parameter
(a)
(b)
(c)

Fig. 15 (II)

INSTITUTE OF INFORMATION SCIENCES AND ELECTRONICS
UNIVERSITY OF TSUKUBA
TSUKUBA-SHI, IBARAKI 305 JAPAN

REPORT DOCUMENTATION PAGE	REPORT NUMBER ISE-TR-90-86

TITLE

On the Efficiency of an SOR-like Method Suited to Vector Processor *

AUTHOR(S)

Masaaki Sugihara, *Department of Economics, Hitotsubashi University*
Yoshio Oyanagi, *Institute of Information Sciences, University of Tsukuba*
Masatake Mori, *Faculty of Engineering, University of Tokyo*
Seiji Fujino, *Institute of Computational Fluid Dynamics*

REPORT DATE	NUMBER OF PAGES
October 1990	36
MAIN CATEGORY	CR CATEGORIES
numerical analysis	

KEY WORDS

SOR, vector processor, acceleration parameter, hyperplane method

ABSTRACT A simple implementation on vector processors of the SOR method for solving linear system of equations arising from the discretization of partial differential equations makes the SOR method into another, which, though, looks like the SOR method. In this paper, the efficiency of this SOR-like method (pseudo-SOR method) is investigated. For the Poisson equation in a rectangular region, in both five-point and nine-point discretization, we prove analytically that the optimal acceleration parameter is smaller and the optimal convergence rate is lower in the pseudo-SOR method. Comparison on several vector computers was made between the SOR method vectorized with the hyperplane technique and the pseudo-SOR method. It turned out that the hyperplane SOR is superior to the pseudo-SOR method although the latter is easily vectorizable on vector processors. We also tested an example of Poisson equation descretized in a curved coordinates in a region bounded by two eccentric circles and found numerically that the optimal acceleration parameter becomes small and the convergence becomes slow in the pseudo-SOR method, while the optimal acceleration parameter in the SOR method does not change appreciably. The genuin SOR method is superior to the pseudo-SOR method.

SUPPLEMENTARY NOTES

*Invited talk presented by Oyanagi at *International Congress on Computational and Applied Mathematics* held at Katholieke Universiteit Leuven, Belgium, on July 23–28, 1990.