Parallel Computations of Incompressible Viscous Flow in a Lid-driven Square Cavity and Program Performance on the NWT Computer System*

Shigeki HATAYAMA*1

ABSTRACT

The NWT computer system available at the NAL since February 1993 comprises two system administrators, \( n \) processing elements (where \( n \) was 140 at the beginning, and is 166 at present) and a crossbar network, and operates as a distributed-memory message-passing MIMD computer. Each processing element itself is a vector computer. This paper reports parallel computations of incompressible viscous flow in a lid-driven square cavity on the NWT computer system. In order to obtain numerical solutions of this flow, consistent finite-difference approximations on non-staggered grids and four iterative solution methods are used. Computations are performed on the Reynolds number range of \( Re = 0 \sim 10^5 \), and effects of the Reynolds number, number of processing elements (\( pe \)) in the parallel processing, solution method and grid size on the computational results are examined. Actual rates of the parallelised square cavity programs on the NWT computer system are measured, and two characteristic parameters of these programs are estimated for the cases that the actual rate is considered as a function of \( pe \) and that the actual rate is considered as a function of the grid size. Measurements of the maximum actual rate and estimations of the speedup and efficiency against \( pe \) on the NWT computer system are indicated as well.

Keywords: square cavity problem, non-staggered grid, consistent finite-difference approximation, iterative method/Jacobi/red-black/CG/ADI, SIMD/MIMD computing, synchronization/data-transfer overhead, program performance, maximum performance, half-performance grain size, characteristic parameters of parallelised program, speedup/efficiency, NWT computer system

概要

昭和58年1月に運用を開始した航空宇宙技術研究所の数値風洞、シールド計算機システムは世界に類例を見ない新概念の並列計算機システムであって、台の制御計算機、台の要素計算機の値は運用開始時において台、現在は台、並びにクロスバーネットワークから成されていて、分散メモリ型メッセージパッキング方式の計算機として動作する。各要素計算機自体はベクトル計算機である。

従って、本研究の主たる動機はこの新しい計算機上の並列処理によって、実際に流れ場の数値計算がどの程度の精度で可能であるか、そのプログラム処理速度は使用台数を増していくとどの程度まで向上させることができるか、その処理速度に限界をもたらす各種のシステム・オーバヘッドなどの程度のものか、そして最後に他の計算機と容易に比較可能なように、一般的に用いられているプログラム処理性能を表す特性パラメータを推定することにある。
1. INTRODUCTION

The incompressible viscous flow in a lid-driven square cavity has been investigated from more than thirty years ago as a representative problem of closed separated flows. Experimental studies of this flow are found chronologically in [15, 16, 23, 19, 6], and analytical studies by means of the finite Fourier series expansions found in [15, 23, 5]. Studies on numerical solutions of this flow are found in [19, 8, 21, 4, 20, 3, 2, 7] which deal with only the vorticity ($\zeta$) and streamfunction ($\psi$), and found in [14, 16, 5, 6, 22, 17, 1] which deal with the pressure ($p$) as well as $\zeta$ and $\psi$ or the primitive variables. In general, as the Reynolds number is larger, the time required to obtain numerical solutions for $\zeta$ and $\psi$ is much longer, but the time required to obtain numerical solutions for $p$ is shorter. However there becomes more difficulty in treating the pressure Poisson equation with Neumann boundary conditions. By the way the maximum Reynolds number dealt with in previous studies mentioned above is $Re = 10^4$, and the maximum grid size is 257\*257 [3, 7], where we note that [8] and [17] had dealt with $Re = 10^5$ and $Re = 3\times10^4$, respectively, but that both numerical solutions could not capture the secondary and tertiary vortices located at the corners of the square cavity due to a rather small grid size.

We have extensively examined the difference between numerical solutions based on marching the unsteady equations in time and on treating the steady-state equations, and the difference between numerical solutions based on staggered grids and on non-staggered grids during this investigation. In this report we indicate numerical solutions based on consistent finite-difference approximations on non-staggered grids for steady-state Navier-Stokes equations [1], which look like to be most superior. We deal with $Re = 10^5$ as the maximum Reynolds number, $16k \times 16k$ ($1k = 1024$) as
the maximum grid size, and the Jacobi/red-black/CG/ADI methods as the iterative solution methods [18].

All the computations are performed in parallel on the Numerical Wind Tunnel (NWT) computer system available at the National Aerospace Laboratory since February 1993 [9, 10, 11, 12]. This system consists of two system administrators, n processing elements (where \( n = 140 \) at the beginning, and is 166 at present) and a crossbar network, and operates as a distributed-memory message-passing MIMD computer. Each processing element itself is a vector computer. The language to describe parallel processing is the NWT Fortran. Main memories of the NWT are physically distributed across the processing elements, but to ease programming, the logical model of the NWT assumed a hierarchical memory parallel computer system for programming offers the virtual global space (or global memory) shared by the selected processing elements to users. On the other hand, each local space (or local memory) is the memory specific to each processing element. Details about the machine architecture and logical model of the NWT, the NWT Fortran, communication and synchronization are shown in [10].

The prime motive of this study is to make clear the following problems which originate from parallel processing on the NWT computer system:

1. Is the numerical computation of flows possible to how degree of accuracy practically?
2. Can the processing speed of parallelised programs improve to how degree along with increase of \( pe \)?
3. Are all kinds of system overhead bringing the processing speed bounds how degree?
4. How degree of value are the characteristic parameters of the NWT computer system?

In this paper, for making clear the above problem (1), we adopt a lid-driven square cavity problem that has the long study history mentioned above, and obtain numerical solutions of this flow by the consistent finite-difference schemes on non-staggered grids. As the solution method used at this juncture, we adopt four iterative solution methods of Jacobi/red-black/CG/ADI. From the obtained results, we show the process of stabilization of the eddy system and the location of the center of the primary vortex in the square cavity on the Reynolds number range of \( Re = 0 \sim 10^5 \), and examine in detail effects of the Reynolds number, the selected number of the processing elements \( (pe) \) for MIMD computations, the solution method and the grid size on numerical solutions of \( \zeta \), \( \psi \) and \( p \).

Next for making clear the above problem (2), we examine in detail processing speeds of all kinds of parallelised programs by increase of \( pe \), and make its bounds clear. Next for making clear the above problem (3), we examine in detail overhead of each compiler directive of the NWT Fortran which is inserted into the general Fortran program. Finally for making clear the above problem (4), we measure actual rates of the parallelised square cavity program adopted each solution method, and estimate in detail two characteristic parameters, \( \tilde{R}, \tilde{p}_e, pe, n \) and \( (\tilde{R}, s_{1/2}) \) of these programs on the NWT computer system with respect to the cases that the actual rate is considered as a function of \( pe \) for a fixed grid size [13] and that the actual rate is considered as a function of the grid size for a fixed \( pe \), respectively.

For almost all results for the actual rate on the NWT computer system, we show both values when applied to the system software had been available at the NAL during the period April to June 1993 and during the period April to May 1994. Hence the degree of improvement on the NWT system software becomes self-evidently, which is the result had been vigorously continuing subsequent improvement during one year.

2. FORMULATION

We consider an incompressible viscous flow in a square cavity by a uniformly moving upper surface as shown in Figure 1, where \( H \) is the depth of cavity, \( L \) the width of cavity and \( H = L = 1 \). The dimensionless stream-function vorticity conservation form of the two-dimensional incompressible Navier-Stokes equations is as follows:

\[
\begin{align*}
\zeta_t &= -\psi \zeta_x + \psi x \zeta_y + \frac{1}{Re} (\zeta_{xx} + \zeta_{yy}), \\
\psi_t &= \zeta_x = -\zeta_x,
\end{align*}
\]

where \( \zeta \) is the vorticity, \( \psi \) the stream-function, \( Re \) the Reynolds number, \( t \), the time, and \( x \) and \( y \) the axial and normal coordinates, respectively. The subscripts \( t \), \( x \) and \( y \) refer to partial derivatives with respect to \( t \), \( x \), and \( y \).
The boundary conditions on the stream-function equation (2) for flow in a lid-driven cavity with the upper surface translating to the right with uniform velocity \( u = 1 \) and with no flow at the other boundaries are at the upper surface

\[
\psi_x = 1, \quad \psi_y = 0, \quad \psi = 0,
\]

and at the bottom, left and right surfaces

\[
\psi_x = 0, \quad \psi_y = 0, \quad \psi = 0.
\]

The boundary conditions on the vorticity equation (1) are obtained by applying the boundary conditions for the stream-function at the solid boundaries as follows:

\[
\zeta = -(\psi_x + \psi_y), \quad \text{at } x = 0, 1 \text{ and } y = 0, 1.
\]

The steady-state Navier-Stokes equations are deduced from (1) and (2) if we set \( \zeta_t = 0 \). When we could obtain the steady-state solutions \( \zeta \) and \( \psi \), the primitive variables can be computed from the following equations:

\[
\begin{align*}
u_x + \nu_y = (\nu \zeta_x - (u \zeta_y) = \sigma, \\
\end{align*}
\]

where \( \sigma \) is the total pressure, the velocity component in the \( x \)-direction and the velocity component in the \( y \)-direction, respectively.

The boundary conditions on the velocity equations (6) are at the upper surface

\[
u = 1, \quad \nu = 0,
\]

and at the bottom, left and right surfaces

\[
u = 0, \quad \nu = 0.
\]

The following Neumann boundary conditions on the pressure equation (7) are obtained by applying the momentum equations at the solid boundaries [1]:

\[
p_x = \nu \zeta - \frac{1}{Re} \zeta_x, \quad \text{at } x = 0, 1.
\]

Solutions to (7) with (10) and (11) are unique within an arbitrary constant, which can be determined by using the relation

\[
\int_0^1 \int_0^1 \rho dxdy = \text{constant}.
\]

The existence of a solution for (7) with (10) and (11) requires the satisfaction of the following compatibility condition:

\[
\int_0^1 \int_0^1 \sigma dxdy = \int n p dS,
\]

where \( n \) is the outward normal to the boundary contour \( S \), enclosing the solution domain.

3. **FINITE-DIFFERENCE APPROXIMATIONS**

In this Section we discuss the numerical method that we used to obtain the steady-state solutions to (1)-(11), on non-staggered and uniform grids. Let \( h = 1/N \) be the mesh size in both the \( x \)- and \( y \)-directions so that the plane \((x, y)\) is discretized as \( x = ih \) and \( y = jh \) where \( i, j = 0, 1, \ldots, N \). We denote \( \zeta_{ij} = \zeta (ih, jh) \), and so on. All partial derivatives are approximated using second order accurate formulas.

3.1 Finite-difference approximation for steady-state equation (1)

\[
(\psi_{i+1,j} - \psi_{i,j}) \left( \zeta_{i+1,j} - \zeta_{i,j} \right) - (\psi_{i,j+1} - \psi_{i,j}) \left( \zeta_{i,j+1} - \zeta_{i,j} \right)
= - \frac{4}{Re} \left( \zeta_{i+1,j} + \zeta_{i-1,j} + \zeta_{i,j+1} + \zeta_{i,j-1} - 4 \zeta_{i,j} \right),
\]

where \( i, j = 1, 2, \ldots, N - 1 \).

3.2 Finite-difference approximation for equation (2)

\[
\psi_{i+1,j} + \psi_{i-1,j} + \psi_{i,j+1} + \psi_{i,j-1} - 4 \psi_{i,j} = -h^2 \zeta_{i,j},
\]

where \( i, j = 1, 2, \ldots, N - 1 \).

3.3 Finite-difference approximations for equations (3) and (4)

\[
\psi_{i,j} = \psi_{N,j} = \psi_{i,0} = \psi_{i,N} = 0,
\]
where \( i, j = 0, 1, \ldots, N \).

### 3.4 Finite-difference approximations for equation (5)

The finite-difference approximations for (5) are obtained from (15) and (16) by enforcing reflection at the boundaries as follows: at the upper and bottom surfaces

\[
\zeta_{i,N} = -\frac{2}{h^2} (\psi_{i,N-1} + 1), \quad \zeta_{i,0} = -\frac{2}{h^2} \psi_{i,1},
\]

where \( i = 1, 2, \ldots, N-1 \), and at the left and right surfaces

\[
\zeta_{0,j} = -\frac{2}{h^2} \psi_{1,j}, \quad \zeta_{N,j} = -\frac{2}{h^2} \psi_{N-1,j},
\]

where \( j = 1, 2, \ldots, N-1 \).

### 3.5 Finite-difference approximations for equation (6)

\[
\begin{align*}
\frac{1}{2h} (\psi_{i,j+1} - \psi_{i,j-1}) & = u_{ij}, \\
\frac{1}{2h} (\psi_{i+1,j} - \psi_{i-1,j}) & = v_{ij},
\end{align*}
\]

where \( i, j = 1, 2, \ldots, N-1 \).

### 3.6 Finite-difference approximation for equation (7)

In order that the compatibility condition (13) is exactly satisfied on a non-staggered grid, we must use the following consistent finite-difference approximation [1]:

\[
p_{i,j+1} + p_{i-1,j} + p_{i,j+1} + p_{i,j-1} - 4p_{ij} = \frac{h}{4} (u_{i,j+1} + v_{i,j+1}) (\zeta_{i,j+1} + \zeta_{i,j}) - (u_{i,j} + v_{i,j-1}) (\zeta_{i,j} + \zeta_{i,j-1}) - (u_{i+1,j} + u_{i,j}) (\zeta_{i+1,j} + \zeta_{i,j}) + (u_{i,j} + u_{i,j-1}) (\zeta_{i,j} + \zeta_{i,j-1}),
\]

where \( i, j = 1, 2, \ldots, N-1 \).

### 3.7 Finite-difference approximations for equations (8) and (9)

\[
\begin{align*}
\frac{1}{2h} (\psi_{i+1,j} - \psi_{i-1,j}) & = u_{i+1,j} - u_{i-1,j} = u_{ij}, \\
\frac{1}{2h} (\psi_{i,j+1} - \psi_{i,j-1}) & = v_{i,j+1} - v_{i,j-1} = v_{ij},
\end{align*}
\]

where \( i, j = 1, 2, \ldots, N-1 \).

### 3.8 Finite-difference approximations for equations (10) and (11)

For consistency, the pressure gradients \( p_x \) and \( p_y \) computed from (10) and (11) should be evaluated at \( x = h/2, x = -h/2, y = h/2 \) and \( y = -h/2 \), respectively, as follows [1]: at the upper and bottom surfaces

\[
p_{i,0} = p_{i,1} - \frac{1}{4Re} (\zeta_{i,1} + \zeta_{i,0} - \zeta_{i,1} - \zeta_{i,0}) + \frac{h}{4} v_{i,1} (\zeta_{i,1} + \zeta_{i,0}),
\]

\[
p_{i,N} = p_{i,N-1} + \frac{1}{4Re} (\zeta_{i,N-1} + \zeta_{i,N} - \zeta_{i,N-1} - \zeta_{i,N}),
\]

where \( i = 1, 2, \ldots, N-1 \), and at the left and right surfaces

\[
p_{0,j} = p_{1,j} - \frac{1}{4Re} (\zeta_{1,j} + \zeta_{0,j} - \zeta_{1,j} - \zeta_{0,j}) - \frac{h}{4} u_{1,j} (\zeta_{1,j} + \zeta_{0,j}),
\]

\[
p_{N,j} = p_{N-1,j} - \frac{1}{4Re} (\zeta_{N-1,j} + \zeta_{N,j} - \zeta_{N-1,j} - \zeta_{N,j}) + \frac{h}{4} u_{N-1,j} (\zeta_{N,j} + \zeta_{N-1,j}),
\]

where \( j = 1, 2, \ldots, N-1 \).

### 4. Iterative Methods for Numerical Steady-State Solutions

In this section we discuss four iterative methods that we used to obtain the steady-state solutions to equations (14)-(25) [18]. Let \( x \) be an \( n \)-column vector, let the superscript \( T \) refer to the transposition, and let us be

\[
x^T = (\zeta_{1,1}, \zeta_{1,2}, \ldots, \zeta_{N,1}; \zeta_{1,N}; \ldots; \zeta_{N,1,N}).
\]

or

\[
x^T = (\psi_{1,1}; \psi_{1,2}; \ldots; \psi_{N,1}; \psi_{N,2}; \ldots; \psi_{1,N-1}; \psi_{2,N-1}; \ldots; \psi_{N,N-1}).
\]

or

\[
x^T = (p_{1,1}; p_{2,1}; \ldots; p_{N,1}; p_{1,2}; \ldots; p_{N,2}; \ldots; p_{1,N-1}; p_{2,N-1}; \ldots; p_{N,N-1}).
\]
where the $(N-1)^2$ variables $\zeta_{ij}$, $\varphi_{ij}$ and $p_{ij}(i, j = 1, 2, ..., N-1)$ at the interior grid points are unknowns in (14), (15) and (20), respectively. Hence we can write (14), (15) and (20) as a linear system of $n = (N-1)^2$ equations in the form $Ax = b$, where $A = \{a_{ij}\}$ is an $n \times n$ coefficient matrix, and $b$ an $n$-vector. Let $D = \text{diag}(a_{11}, ..., a_{nn})$ be a diagonal matrix containing the diagonal elements of $A$. The matrix $A$ for (14), (15) or (20) is a block tridiagonal matrix of the form

$$A = \begin{bmatrix} C_{11} & D_{12} & \cdots & D_{1 N-2} & N_{1 N-1} \\ D_{21} & C_{22} & \cdots & D_{2 N-2} & N_{2 N-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ D_{N-1 N-2} & D_{N-1 N-3} & \cdots & C_{NN} \end{bmatrix},$$

where all the tridiagonals $C_i$ and diagonals $D_{ij}(i, j = 1, 2, ..., N-1)$ are $N1 \times N1$ (where $N1 = N - 1$) matrices of the forms

$$C_1 = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1 N-1} \\ a_{21} & a_{22} & \cdots & a_{2 N-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1 N-1} \end{bmatrix},$$

$$D_1 = \begin{bmatrix} a_{11} & a_{11} & \cdots & a_{1 N-1} \\ a_{21} & a_{21} & \cdots & a_{2 N-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1 N-1} \end{bmatrix}.$$

4.1 Jacobi’s method

Given the linear system $Ax = b$, the Jacobi iteration follows from the splitting $A = D - (D - A)$, which leads to the iteration

$$x^{k+1} = Hx^k + d, \quad k = 0, 1, 2, ..., (32)$$

where the superscript indicates the iteration number, and

$$H = D^{-1}(D - A), \quad d = -D^{-1}b.$$

4.2 Red-black ordering

The grid points are divided into two classes, red (or odd) and black (or even), and then ordered left to right, bottom to top within each class. The unknowns at the grid points are ordered analogously. Then the given linear system $Ax = b$ will become of the form

$$\begin{bmatrix} D_R & C_R \\ C_R & D_B \end{bmatrix} \begin{bmatrix} x_R \\ x_B \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix},$$

(34)

where under reorderings

$$D = \begin{bmatrix} D_R \\ D_B \end{bmatrix}, \quad x = \begin{bmatrix} x_R \\ x_B \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}.$$ (35)

Then the red-black iteration is

$$x_R^{k+1} = D_R^{-1}(b_1 - C_R x_B^k),$$

$$x_B^{k+1} = D_B^{-1}(b_1 - C_B x_R^{k+1}),$$

(36)

(37)

which show to uncouple into the two separate parts.

4.3 CG method

Given the linear system $Ax = b$, let $(x, y) = x^T y$ be the inner product and $r^k = b - Ax^k$ the residual at the $k$th step. The conjugate gradient (CG) iteration is the following scheme:

Choose $x^0$, set $p^0 = \tau^0$, compute $(\tau^0, \tau^0)$, for $k = 0, 1, 2, ...$

$$\alpha_k = -(\tau^k, \tau^k)/(p^k, Ap^k),$$

$$x^{k+1} = x^k - \alpha_k p^k,$$

$$\tau^{k+1} = \tau^k + \alpha_k Ap^k,$$

if not converge, continue,

$$\beta_k = (\tau^{k+1}, \tau^{k+1})/(\tau^k, \tau^k),$$

$$p^{k+1} = \tau^{k+1} + \beta_k p^k.$$ (38)

4.4 ADI method

Given the linear system $Ax = b$, let $A = H + V$ be the following splitting:
where $H$ is a block diagonal, $V$ is a block tridiagonal, and all the tridiagonals $T_{ii}$ and diagonals $D_{ii} (i = 1, 2, \ldots, N - 1)$ are $N \times N$ matrices of the forms

\[
T_{ii} = \begin{bmatrix}
T_{11} & & \\
& \ddots & \\
& & T_{N-1 \times N-1}
\end{bmatrix},
\]

\[
V = \begin{bmatrix}
D_{11} & D_{12} & & \\
D_{21} & D_{22} & & \\
& & \ddots & \\
& & & D_{N-1 \times N-1}
\end{bmatrix},
\]

\[
T_{ii} = \begin{bmatrix}
a_{(i-1)\times N+1} & a_{(i-1)\times N+2} & & \\
& \ddots & & \\
& & \ddots & \\
& & & a_{(N-1)\times N+1}
\end{bmatrix},
\]

\[
D_{ii} = \begin{bmatrix}
a_{(i-1)\times N+1} & & \\
& \ddots & \\
& & \ddots & \\
& & & a_{N\times N+1/2}
\end{bmatrix}
\]

respectively. Then the alternating direction implicit (ADI) iteration is

\[
(\alpha I + H)x^{k+1/2} = \left(\alpha I - V\right)x^{k} + b,
\]

\[
(\alpha I + V)x^{k+1} = \left(\alpha I - H\right)x^{k+1/2} + b,
\]

where $\alpha_i > 0$ ($i = \zeta, \psi$ or $p$) is a constant.

### 4.5 Remarks

(1) **Initial values**

We set in all the iterative solution methods

\[
x^0 = 0.
\]

(2) **Iteration number of $\psi$**

Let $M_{\psi}$ be the iteration number of $\psi$ per each iteration of $\zeta$. For the red-black, CG or ADI method, we set $M_{\psi} = 1$. For the Jacobi method, we set $M_{\psi} = 4$, of which case we call $J1$. Talking from the point of view of overheads, only the Jacobi method incurs overheads due to hold and update the current values of $\psi_i$, but only when we set $M_{\psi} = 1$, we can remove these overheads, of which case we call $J2$.

(3) **Relaxation parameters**

We used the following relaxation parameters, $\omega$,

\[
(\zeta = \zeta, \psi$ or $p)$:
\]

\[
\hat{x}^{k+1} = x^{k} + \omega(x^{k+1} - x^{k}),
\]

\[
x^{k+1} = \hat{x}^{k+1}.
\]

(4) **Convergence test**

In order to test for convergence of the iterates, we used the following test:

\[
\max_{1 \leq i \leq n} |x^{k+1} - x^{k}| \leq 10^{-6}.
\]

(5) **Steady flow methods**

It is known that some steady-state flow iterative methods are strictly equivalent to time-dependent unsteady flow methods, with under- and over-relaxation adjustments being equivalent to $\Delta t$ changes, and most steady-state iterative methods are at least analogous to time-dependent unsteady methods [24]. Hence, we can see the transient behavior of the flow by steady flow methods too until iterative solutions converge to the steady-state solution.

### 5. RESULTS OF PARALLEL COMPUTATIONS OF SQUARE CAVITY FLOWS ON THE NWT COMPUTER SYSTEM

A large amount of numerical information on the Reynolds number range of $Re = 10^{-6} \sim 10^{5}$ has been collected during this investigation. All the computations were executed in parallel on the NWT computer system.

In this Section some selected results are presented with the particular aim of describing the asymptotic behaviour of square cavity flows at the high Reynolds numbers. Furthermore we show comparisons between results of square cavity flows at a fixed Reynolds number in the cases that we vary the selected number, pe, of the processing elements, the solution method and the grid size, respectively.

#### 5.1 Effect of the Reynolds number on the behaviour of square cavity flows

First of all, we indicate the process of stabilization of the eddy system in the square cavity (see Figure 1). Square cavity flows at the representative Reynolds numbers are shown in Figures 2-10, where we note
that (1) and (2) of Figure 6 agree almost completely with Figure 3 and Figure 4 in [7], respectively. The computational conditions used to obtain these Figures are given in Table 1, and the values of minimum, maximum and each contour for $\zeta$, $\psi$ and $p$ given in Table 2.

The flows on the low Reynolds number range of $Re = 10^{-6} \sim 10^3$ are characterized by three eddies: one, the primary eddy and two other, secondary eddies located at the corners of the bottom wall (see Figures 2-4). As $Re$ increases beyond $10^3$, the secondary eddy located at the upper part of the left wall appears within the cavity (see Figure 5). As $Re$ increases furthermore, the tertiary and fourth-order eddies located at the corners of the bottom wall become to appear (see Figures 6-7). As $Re > 2 \times 10^4$, however, the secondary or/and tertiary eddies located at the left or/and right corners of the bottom wall become unsteady (see Figures 8-10). Figure 11 shows effect of $Re$ on the profiles of $u$, $v$, $\zeta$ and $p$ through the center of the primary vortex at $Re = 10^{-6}$, $10^2$, $10^3$, $10^4$ and $2 \times 10^4$ corresponding to Figures 2-4 and 6-7, respectively.

(1) Primary vortex

Figure 12 shows the location of the center of the primary vortex as a function of the Reynolds number. Until the Reynolds number comes up to 120, the center of the primary vortex moves upstream. As the Reynolds number increases further, it moves monotonously towards the center of the cavity. In the inviscid limit the total pressure is conserved along the streamlines, in which case the total-pressure contours should coincide with the streamlines. Hence the close similarity between (1) and (3) of Figures 5-10 in the core region is evident. On the other hand, the vorticity changes rapidly at the boundaries and is flat in the middle of the cavity at the high Reynolds numbers.

(2) Secondary vortices

The secondary eddy located at the right corner of the bottom wall grows with $Re$, attains maximum at $Re = 10^3$, and at $Re > 10^3$ begins to thin out monotonously. It is steady at $Re \leq 2 \times 10^4$, but becomes unsteady at $Re > 5 \times 10^4$. The third secondary eddy located at the upper part of the left wall begins to appear at $Re > 10^3$, and grows monotonously with $Re$. It is steady at all the Reynolds numbers.

(3) Tertiary and fourth-order vortices

The tertiary eddies located at the corners of the bottom wall appear slightly at $Re = 10^4$, and grows with $Re$. The right tertiary eddy is steady until $Re \leq 2 \times 10^4$, but at $Re > 2 \times 10^4$ becomes unsteady. The left tertiary eddy is steady until $Re \leq 5 \times 10^4$, but becomes unsteady at $Re > 5 \times 10^4$. On the other hand, the right fourth-order eddy at the corner of the bottom wall becomes to appear at $Re \geq 2 \times 10^3$, and the left fourth-order eddy at $Re \geq 3 \times 10^4$. These repeat to slightly grow and decay, as the secondary and tertiary eddies at both the corners of the bottom wall become to be unsteady.

5.2 Comparison between results obtained by varying $pe$ in the parallel processing

Comparison between results obtained for $2k$ at $pe = 2$ $(i = 0, 1, \ldots, 4)$ for each solution method was done for square cavity flows at $Re = 10^3$, where computational conditions are as shown in Table 3. Complete agreement was obtained for the Jacobi, red-black and ADI methods. For the CG method, complete agreement was obtained among $u$’s, $v$’s and $\zeta$’s, but slightly disagreement was indicated among $p$’s as shown in Figure 13, of which reason is due to be necessary for this method to compute a global summation in the parallel processing each iteration, as seen from (38).

5.3 Comparison between results obtained by varying the solution method

Comparison between results obtained for the four solution methods in case of $Re = 10^3$ and $pe = 16$ showed that complete agreement with profiles for $Re = 10^3$ of (1)-(4) of Figure 11 was obtained among $u$’s, $v$’s and $\zeta$’s, but that a little disagreement was indicated among $p$’s as shown in Figure 14, where computational conditions are as shown in Table 3.

5.4 Comparison between results obtained by varying the grid size

Comparison between results obtained for $2k \times 2k$ $(i = -2, 0, 1, \ldots, 4; 1k = 1024)$ grids at $Re = 10^3$ by the red-black method is shown in Figure 15, where
computational conditions are given in Table 4, including the values of the execution time until convergence and the center of the primary vortex obtained when we executed in parallel under the selected number of the processing elements mentioned in this Table. Sufficient agreement was shown among $u$’s, $v$’s and $\zeta$’s, but considerable disagreement was indicated among $p$’s. Figure 16 shows contours of the total-pressure solution for each grid size, of which values are same as given for Fig. 4 in Table 2. We cannot make its reason clear at present, but there is a possibility of hiding an elementary problem. By the way, we note the followings:

(1) For $8k*8k$ and $16k*16k$ grids, data of $128*128$ grid points of the intermediate computational results for $\zeta$ and $\psi$ were stored at regular intervals into the secondary memory device, and retrieved to restore full data by means of interpolation before start of the successive iteration. This process had been continued until convergence.

(2) Computations for a $16k*16k$ grid were conducted for another purpose of obtaining data concerned with the mean time between failure (MTBF) of the NWT computer system.

6. PROGRAM PERFORMANCE ON THE NWT COMPUTER SYSTEM

In this Section we indicate results for the actual performance of the parallelised square cavity programs on the NWT computer system. Most of the overheads incurred in the parallel processing arise from the time spent in system software routines supporting user programs. Hence the obtained results in this Section apply only to the system software had been available at the NAL during the period April to June 1993 which we call v1, and during the period April to May 1994 which we call v2. The v2 is an improved version of the v1. By the way all the programs were run with other users on the NWT computer system.

6.1 Measurements of overheads of the used NWT-Fortran-compiler directives

We had used the following NWT-Fortran-compiler directives for MIMD computings of the square cavity programs:

(1) PARALLEL REGION(CD1)/END PARALLEL REGION (CD2); Only the program portion between these statements is executed in parallel by multiple processing elements.

(2) SPREAD DO (CD3); This statement specifies partitioning of repetitions of the DO statement, and execution of each segment on the specified processing element.

(3) SPREAD MOVE (CD4); This statement specifies assignment between local variables and global variables on each processing element.

(4) END SPREAD (CD5); This statement verifies the exit of the SPREAD DO/MOVE statement.

(5) END SPREAD SUM (abc) (CD6); By this statement the global summation operation among the subregions generated by the spread-do-construct is performed at termination of the spread-do-construct.

(6) OVERLAPFIX (CD7); This statement specifies replacement of the overlap range of partitioned local arrays with overlap, with the values of the range corresponding on another processing elements.

(7) MOVEWAIT (CD8); This statement verifies completion of assignment and data transfer that are started by the OVERLAPFIX and SPREAD MOVE statements, respectively.

Figure 17 shows measurements of each overhead for the above-mentioned compiler directives against the selected number of the processing elements, $pe = 2^i$ ($i = 1, 2, \ldots, 7$). For (5) of this Figure we note the following: There is an optimum number of the processing elements corresponding to the amount of the data transfer. The smaller $pe$ is than the optimum number, the more the data-transfer overhead increases due to bottleneck of the amount of the data transfer. The larger $pe$ is than the optimum number, the more the data-transfer overhead increases due to bottleneck of the packet creation.

Furthermore we had used the following NWT-Fortran-compiler directives for MIMD computings without any overhead:

(8) PROCESSOR (CD9); This statement is used to declare the number of the processing elements required to execute a program in parallel.

(9) INDEX PARTITION (CD10); This statement is used to indicate the index range of the association of DO loops and arrays with processing elements and the partition type, and to specify the overlap if necessary.

(10) GLOBAL (CD11)/LOCAL (CD12); These statements specify the allocation of variables to the
global and local memory spaces, respectively.

(11) EQUIVALENCE (CD13); This statement specifies the sharing of the storage by two or more variables in a program. When a global variable and a local variable are combined, a global and local memory spaces are shared.

6.2 Some remarks on the parallelised square cavity programs

The flow of the square cavity programs is as follows:

(A) Declare the CD9-CD13 statements, etc.,
(B) insert the CD1 statement,
(C) set the initial values of (16), (21), (44), etc.,
(D) for \( k = 1, 2, 3, \ldots \) (iterations for \( \zeta \) and \( \psi \)),
(D.1) hold the current values of \( \zeta \) and \( \psi \),
(D.2) update the boundary values of \( \zeta \) by (17) and (18),
(D.3) compute the internal values of \( \zeta \) by (14), (15) and (45),
(D.4) test for convergence of the iterates by (46), and if converge, continue,
(E) compute the internal values of \( u \) and \( v \) by (19),
(F) compute the values of the right hand side of (20),
(G) for \( k = 1, 2, 3, \ldots \) (iterations for \( p \)),
(G.1) hold the current values of \( p \),
(G.2) update the boundary values of \( p \) by (22)-(25),
(G.3) compute the internal values of \( p \) by (20) and (45),
(G.4) test or convergence of the iterates by (46), and if converge, continue,
(H) compute the particular solution of \( p \) by (12),
(I) insert the CD2 statement,
(J) store the computational results,

where only the program portions for (D.3), (F) and (G.3) differ with each other every solution method.

Let \( N_c \) be the number used the pairs of CD3 and CD5 in the program portion (C), and so on provided that the numbers for (D) and (G) are the values per each iteration. These numbers for the portions common to all the parallelised square cavity programs are as follows: \( N_c = N_e = 1 \) (pair of CD3 and CD5), \( N_{d.1} = N_{g.1} = 1 \) (pair of CD3 and CD5) and 1 (pair of CD7 and CD8), \( N_{d.2} = N_{g.2} = 2 \) (pair of CD3 and CD5) and 1 (pair of CD7 and CD8), \( N_{d.4} = N_{g.4} = 1 \) (pair of CD3 and CD6), \( N_p = 1 \) (pair of CD3 and CD6) and 1 (pair of CD3 and CD5). On the other hand, the numbers of \( N_{d.3}, N_{f} \) and \( N_{g.3} \) are different every solution method used as follows: Let HU used below mean to hold or update the current values of \( \zeta \) or \( \psi \).

(1) Jacobi method

The scheme of this method is given by (32). Hence for \( J_1 \), \( N_{d.3} = 10 \) of which 5 are used for HU (pair of CD3 and CD5) and 4 (pair of CD7 and CD8), \( N_f = 1 \) (pair of CD3 and CD5), an \( N_{g.3} = 2 \) (pair of CD3 and CD5). For \( J_2 \), \( N_{d.3} = 2 \) (pair of CD3 and CD5), and \( N_f = 1 \) (pair of CD3 and CD5).

(2) Red-black ordering

The scheme of this method is given by (36)-(37). Hence \( N_{d.3} = N_{g.3} = 8 \) (pair of CD3 and CD5) and 1 (pair of CD7 and CD8), and \( N_f = 1 \) (pair of CD3 and CD5).

(3) CG method

The scheme of this method is given by (38). Hence \( N_{d.3} = 10 \) of which 2 are used for HU (pair of CD3 and CD5), 2 (pair of CD7 and CD8) and 4 (pair of CD3 and CD6), \( N_f = 1 \) (pair of CD3 and CD5), and \( N_{g.3} = 5 \) of which 1 is used for HU (pair of CD3 and CD5), 1 (pair of CD7 and CD8) and 2 (pair of CD3 and CD6).

(4) ADI method

The scheme of this method is given by (42)-(43). Hence \( N_{d.3} = 26 \) of which 4 are used for HU (pair of CD3 and CD5), 4 (set of CD4, CD5 and CD6) and 4 (pair of CD7 and CD8), \( N_f = 1 \) (pair of CD3 and CD5) and 1 (set of CD4, CD5 and CD8), and \( N_{g.3} = 15 \) of which 2 are used for HU (pair of CD3 and CD5), 2 (set of CD4, CD5 and CD8) and 2 (pair of CD7 and CD8).

Finally we note the followings with respect to the partition of two-dimensional arrays adopted in this study: All the computations of (12), (14)-(25), (32), (36)-(38) and (42)-(46) can be executed in parallel. For the Jacobi, red-black or CG method, we had adopted the band partition of the index range in the y-direction.
Hence in FORTRAN we can take advantage of continuous loading and continuous storage on each register-to-register processing element except when (18) and (24)-(25) are executed, which time becomes to periodic loading and periodic storage. On the other hand, as is obvious from (42) and (43), for the ADI method, we must adopt the band partition of the index range in the \( y \)-direction for computation of (42) and of the index range in the \( x \)-direction for computation of (43), where the case of the former is continuous loading and continuous storage and the case of the latter becomes both periodic. If we would adopt the band partition of the index range in only the \( y \)-direction for computations of (42) and (43), the executing time should increase about three hundred times. Furthermore we must suppress the occurrence of the memory-bank conflict in case of periodic loading and periodic storage. When the bank conflict would occur, the performance of the execution should deteriorate about half.

6.3 Definition of several actual rates for the square cavity programs

We use below the following actual rates defined for the square cavity programs:

\[
\begin{align*}
\tau_1 &= \text{actual rate of the part (D.3) in (47)} \\
&= \text{actual rate of the kernel part for iterations of } \zeta \text{ and } \psi, \\
\tau_2 &= \text{actual rate of the part (D) in (47)} \\
&= \text{actual rate of the overall part for iterations of } \zeta \text{ and } \psi, \\
\tau_3 &= \text{actual rate of the part (D) excepted (D.4) in (47)} \\
&= \text{actual rate of the overall part for iterations of } \zeta \text{ and } \psi, \text{ when the convergence test is excepted,} \\
\tau_4 &= \text{actual rate of the part (D) excepted (D.1), a pair of CD7 and CD8 in (D.2), and pairs of CD3 and CD5 used for HU and pairs of CD7 and CD8 in (D.3) as well as (D.4) in (47),} \\
\tau_5 &= \text{actual rate of the part (D) excepted (D.4) in (47) in case of using the loop coalescing technique,}
\end{align*}
\]

where \( \tau_5 \) indicates the actual rate when execution of the local memory access statements with no arithmetic operation, etc. is intentionally suppressed, and hence we define the following:

\[
d_{\text{mu}} = \frac{\tau_4}{\tau_3} = \text{degree of degradation of the actual rate of the part (D) excepted (D.4) in (47) mainly due to the local memory access bottleneck.}
\]

By the way the loop coalescing is the technique that restructures certain types of multiply nested loops into single parallel loops. When we adopt this technique to the square cavity programs, each code segment must be executed conditionally in the transformed loop to assure correct execution of the original code segment corresponding. Hence we can almost double the amount of arithmetic operations every parallel loop by the loop coalescing, but a new source of degradation of the actual rate is introduced.

6.4 The characteristic parameters of the square cavity programs on the NWT computer system

In practise we find for many parallelised programs that as \( pe \) becomes larger, synchronisation and other overheads usually increase rapidly with \( pe \), with the result that there is often a maximum in the program performance and with a subsequent reduction in performance as \( pe \) further increases. This measured behaviour can be fitted to the function [13]

\[
\tau (pe) = \frac{\tau_{\text{u}}}{1 + \frac{pe_{1/2}}{pe}}, \quad \tau_{\text{u}} = \frac{1}{n - 1} \left( \frac{pe}{pe_{1/2}} \right)^{1/2}, \quad (48)
\]

where

\[
\begin{align*}
\tau_{\text{u}} &= \frac{1}{t_{\text{ser}}} , \\
pe_{1/2} &= \frac{t_{\text{ser}}}{t_{\text{ser}}},
\end{align*}
\]

and

\[
\begin{align*}
\tau &= \text{actual performance of the parallelised program,} \\
\tau_{\infty} &= \text{asymptotic performance of the parallelised program} \\
 &= \text{maximum possible performance of the parallelised program when overheads are negligible} \\
 &= \text{Amdahl ceiling,} \\
pe_{1/2} &= \text{number of the processing elements required to achieve half of the asymptotic performance},
\end{align*}
\]
6.5 Effect of the grid size on the actual performance of the square cavity programs on the 128-processing element system

Figure 22 shows effect of the grid sizes of $2^i \times 2^i$ ($i = 8, 9, ..., 14$) provided that the maximum grid size for only the CG method is $12,000 \times 12,000$, on the actual rate $\tau$ of the parallelised square cavity program used each solution method for $v_2$ when $pe = 128$. As shown in this Figure, the actual rate $\tau$ of each parallelised program is fitted very well by the equation

$$\tau = \tau_\infty \left[ 1 + \left( \frac{s_{\sqrt{2}}}{s} \right)^{1/2} \right]^{-1},$$  

(51)

with the following estimated values of the characteristic parameters:

- **Jacobi (J1) for $v_2$** :
  \[ \tilde{\tau}_\infty = 1.037376 \times 10^5 \text{ (Mflop/sec)}, \]
  \[ s_{1/2} = 2.116524 \times 10^3 \text{ (flop)}, \]

(52)

- **Jacobi (J2) for $v_2$** :
  \[ \tilde{\tau}_\infty = 1.346469 \times 10^5 \text{ (Mflop/sec)}, \]
  \[ s_{1/2} = 2.286338 \times 10^3 \text{ (flop)}, \]

(53)

- **red-black for $v_2$** :
  \[ \tilde{\tau}_\infty = 6.835579 \times 10^4 \text{ (Mflop/sec)}, \]
  \[ s_{1/2} = 2.574064 \times 10^3 \text{ (flop)}, \]

(54)

- **CG for $v_2$** :
  \[ \tilde{\tau}_\infty = 1.099030 \times 10^5 \text{ (Mflop/sec)}, \]
  \[ s_{1/2} = 3.223610 \times 10^3 \text{ (flop)}, \]

(55)

where

- $s = N + 1$ where the grid size is $s \times s$,
- $\tilde{\tau}_\infty$ = maximum performance of the parallelised program,
- $s_{1/2}$ = square root of the half-performance grain size of the parallelised program.

We note here the followings compared results of [11] with results of this Subsection: The former has considered the timing of the single work segment which can be distributed amongst multiple processing elements, and the latter the timing of the complete program comprising many work segments. Both these actual performances, however, can be fitted very well by the pipeline function of the same form as (51), because the programs of the latter except for the ADI...
method have been parallelised so as to become almost all the local memory access. Furthermore, for example, \( \tau \) for the Jacobi method (J2) of (53) is 2.74 times (dyads) and 1.93 times (triads) as large as the values for \( pe = 128 \) in Table 2 of [11], and \( s_{1/2}^2 \) is 3.21 times (dyads) and 2.59 times (triads). This means that for the complete program increases frequency of the chaining of two vector instructions and the simultaneous use of both the floating-point multiply and add pipelines with all vectors, so that the maximum program performance raises considerably and hence the half-performance grain size increases as well.

6.6 Measurements of the maximum actual performance of the square cavity programs on the NWT computer system

Let \( \tau_{3\text{num}} \) be the measured maximum of the actual rate \( \tau \). Table 6 and Figure 23 show measurements of \( \tau_{3\text{num}} \) of the parallelised square cavity program used each solution method against the selected number of the processing elements, \( pe = 2^i \) \((i = 0, 1, \ldots, 7)\), where the grid sizes used to obtain these results are given in Table 7. As shown in Figure 23, \( \tau_{3\text{num}} \) of the parallelised programs except for the ADI method are fitted very well by the following equations:

- **Jacobi (J1) for v1:**
  \[
  \tau_{3\text{num}} = 7.984252 \times 10^2 \text{ pe (Mflop/sec)},
  \]
  \( (56) \)

- **Jacobi (J1) for v2:**
  \[
  \tau_{3\text{num}} = 8.336185 \times 10^2 \text{ pe (Mflop/sec)},
  \]
  \( (57) \)

- **Jacobi (J2) for v2:**
  \[
  \tau_{3\text{num}} = 1.058900 \times 10^3 \text{ pe (Mflop/sec)},
  \]
  \( (58) \)

- **red-black for v1:**
  \[
  \tau_{3\text{num}} = 5.419430 \times 10^2 \text{ pe (Mflop/sec)},
  \]
  \( (59) \)

- **red-black for v2:**
  \[
  \tau_{3\text{num}} = 5.362091 \times 10^2 \text{ pe (Mflop/sec)},
  \]
  \( (60) \)

- **CG for v1:**
  \[
  \tau_{3\text{num}} = 7.814897 \times 10^2 \text{ pe (Mflop/sec)},
  \]
  \( (61) \)

- **CG for v2:**
  \[
  \tau_{3\text{num}} = 8.135356 \times 10^2 \text{ pe (Mflop/sec)}. \]
  \( (62) \)

By the way there is no appropriate equation fitted to measurements of \( \tau_{3\text{num}} \) for the ADI method.

We here remark the followings for the obtained results:

1. The reason that actual rates of the red-black method are inferior to ones of the other methods is because the amount of works in each parallel section becomes half the others, as seen from (36) and (37).
2. As seen from (1) and (2) of Table 6, the degree of degradation of \( \tau \), mainly due to the local memory access bottleneck is 1.10- to 1.42-fold. The reason that the values of \( d_{ma} \) for the Jacobi method (J1) are larger than ones for the other methods is because dimensional computer variables are stored to and retrieved from the local memory each minor iteration for \( p \).
3. For example, \( \tau_{3\text{num}} \) for the Jacobi method (J2) of (58) is 2.75 times (dyads) and 1.94 times (triads) as large as the values of (14) and (15) of [11], respectively. These values are same as ones mentioned in Subsection 6.5.
4. To apply the ADI method to the square cavity program is not advisable, because of incurring a large amount of communication overheads due to the global memory access. For three-dimensional cavity programs, however, the scheme of the ADI method as well as the other methods can be parallelised so as to become almost all the local memory access, and hence the ADI method becomes useful as well [12]. Furthermore the clear improvement that \( \tau_{3\text{num}} \) for v2 are about 1.17 times as large as ones for v1 is based on the fact that communication overheads for v2 have decreased about 5.0- to 5.7-fold for v1 [11].

6.7 Speedup and efficiency of the parallelised programs on the NWT computer system

The speedup, \( S_{pe} \), of a parallelised program and the efficiency or effect of the number of the processing elements, \( E_{pe} \), of a parallelised program with respect to itself are defined as follows [18]:

\[
S_{pe} = \frac{T_1}{T_{pe}}, \quad E_{pe} = \frac{S_{pe}}{pe},
\]

where

- \( T_1 = \) execution time of a program using a single processing element by SIMD computing,
- \( T_{pe} = \) execution time of a parallelised program using \( pe \) processing elements.
The square cavity program of such a large-scale that the grid size is greater than 1,500 * 1,500, however, cannot be executed by SIMD computing on the NWT computer system. Hence we must devise the method to estimate the value of $T_1$ for such the large-scale programs.

We can use two time subroutines measured in microseconds on the NWT computer system: One, the GETTOD which measures the wall clock time and the other, the CLOCKV which measures the operating time of the central processing unit and the operating time of the pipeline unit or vector processing unit for each processing element. The elapsed time, $ELP_{pe}$, and the operating time of the central processing unit, $CPU_{pe}$, and the operating time of the vector processing unit, $VPU_{pe}$, between two points in a program using $pe$ processing elements can be measured by using a pair of the GETTOD and CLOCKV, respectively, and generally $ELP_{pe} \geq CPU_{pe} \geq VPU_{pe}$, where these values mean the time for SIMD computing only when $pe = 1$ and the time for MIMD computing when $pe > 1$. For MIMD computing on the NWT computer system, the compiler directives of CD3-CD8 are treated so as to wait for completion of these processes by means of the spin-loop on purpose to be decreased delays due to insert these directives, and hence overheads due to these directives are added in $CPU_{pe}$.

Let $I$ be the number of iterations, $VTC_{pe}$ the ratio of $VPU_{pe}$ to $CPU_{pe}$, $DRW_{pe}$ the division ratio of works into each processing element, $TOH_{pe}$ the total overhead, $WK_1_{pe}$ the work per one iteration, $OH_1_{pe}$ the overhead per one iteration, and $DDS_{pe}$ the degree of degradation of the speedup. Then we have the following relations:

$$VTC_{pe} = \frac{VPU_{pe}}{CPU_{pe}}, \quad DRW_{pe} = \frac{VPU_{pe}}{VPU_1},$$
$$TWK_{pe} = \frac{VPU_{pe}}{VTC_1}, \quad TOH_{pe} = CPU_{pe} - TWK_{pe},$$
$$WK_1_{pe} = \frac{TWK_{pe}}{I}, \quad OH_1_{pe} = \frac{TOH_{pe}}{I},$$
$$DDS_{pe} = \frac{WK_1_{pe}}{WK_1_{pe} + OH_1_{pe}},$$

from which we can obtain

$$S_{pe} = \frac{DDS_{pe}}{DRW_{pe}} = \frac{CPU_1}{CPU_{pe}}.$$  \hspace{1cm} (66)

We now consider the case that the grid size is considerably large. Then, from the reason mentioned above, we have the following relations:

$$CPU_1 = ELP_1 = T_1, \quad CPU_{pe} = ELP_{pe} = T_{pe},$$
$$from which and (66) we can obtain the definition of $S_{pe}$ of (63). As seen from Table 7, the grid sizes used to obtain Table 6 satisfy the condition necessary to derive (67). Therefore we can estimate the value of $T_1$ from the following relation:

$$T_1 = CPU_1 = \frac{VPU_1}{VTC_1} = \frac{VPU_{pe}}{DRW_{pe} * VTC_1},$$

from which and (66)-(67) we can obtain

$$S_{pe} = \frac{VPU_{pe} * pe}{ELP_{pe} * VTC_1}.$$  \hspace{1cm} (69)

Various experiments have shown that the larger the values of $VTC_{pe}$ for the parallelised program are, i.e., the larger the grid size is, the better the value of $S_{pe}$ computed from (69) becomes to coincide with one computed from (63).

Table 8, as one instance, shows comparison between the values of $S_{pe}$ computed from (63) and (69). As seen from this Table, two values of $S_{pe}$ coincide comparatively well despite for such a rather smaller grid size as 1,024*1,024. Hence we can expect that the larger the grid size is, the more the accuracy of $S_{pe}$ estimated from (69) will be raised. Table 9 shows the speedup and efficiency computed from (69) and (63) for the case of $\tau_{3mm}$ and $v_2$ of Table 6 provided that $VTC_1 = 1.0$ which results in the most underestimation, respectively.


Firstly we remark the followings for the numerical solution of the square cavity problem:

1) We think that the non-staggered grid adopted in this study is superior to the staggered grid when we
use the consistent finite-difference approximations for the square cavity problem, and that numerical solutions obtained by treating the steady-state equations are superior to ones obtained by marching the unsteady equations in time.

(2) The solution of $\zeta$ and $\psi$ is scarcely influenced by varying the number of the processing elements in the parallel processing, the solution method and the grid size.

(3) The solution of $p$, however, is slightly influenced by varying the number of the processing elements in the parallel processing only when the CG method is adopted, a little influenced by varying the solution method, and considerably influenced by varying the grid size, as shown in Figure 16. We can’t make its reason clear yet. This indicates difficulty in treating the pressure Poisson equation with Neumann boundary conditions that is yet in question.

Secondly we remark the followings for the solution method:

(1) The solution of $\zeta$ and $\psi$ on the Reynolds number range of $Re = 0 \sim 5 \times 10^3$ can be easily obtained whichever solution method is adopted. At $Re = 5 \times 10^3 \sim 10^4$, however, it gradually becomes difficult to obtain this solution, and at $Re > 10^4$, at last, it seems hardly possible to obtain this solution by any single iterative solution method. Hence it is necessary to alter the solution method adopted for computations until development of the boundary layer and after development of the boundary layer. The Jacobi and CG methods are very promising for computation of the former, and for computation of the latter, only the red-black method is hopeful.

(2) The Jacobi preconditioning, preconditioned CG and incomplete Choleski CG methods are useful to obtain the solution of $\zeta$ and $\psi$ at small Reynolds numbers, but become useless as $Re$ is larger.

(3) The solution of $p$ at $Re \geq 10$ can be easily obtained whichever solution method is adopted. At $Re < 1$, however, we had better avoid adoption of the Jacobi method to obtain this solution.

Thirdly we remark the followings for the convergence time:

(1) At $Re \leq 600$, the time required until $\zeta_{i,j}$ and $\psi_{i,j}$ converge becomes longer in order of red-black, Jacobi (J1), Jacobi (J2), CG and ADI for SIMD computings provided that $\phi_i (i = \zeta, \psi)$ are optimum, and at $600 < Re \leq 5 \times 10^3$, longer in order of Jacobi (J1), Jacobi (J2), red-black, CG and ADI. This time becomes much longer as $Re$ increases. At $Re \geq 10^4$, we find difficulty in being converged to a solution if we did not use the Jacobi-red-black or CG-red-black combination as a solution method. For example, this time by the red-black method becomes more than double as large as one by the CG-red-black combination.

(2) At $Re < 100$, the time required until $p_{i,j}$ converge becomes longer in order of red-black, CG, ADI and Jacobi for SIMD computing, at $100 \leq Re \leq 600$ longer in order of red-black, Jacobi, ADI and CG, and at $Re > 600$ longer in order of Jacobi, red-black, ADI and CG. This time becomes shorter as $Re$ increases, and at $Re > 10^5$ nearly constant for a fixed grid size.

Finally we mention that the solution method presented a larger actual rate is not always advantageous to be converged to a numerical solution of $\zeta$, $\psi$ and $p$, as seen from discussions mentioned above.

8. CONCLUSIONS

In this paper we reported parallel computations of incompressible viscous flow in a lid-driven square cavity on the NWT computer system, described the asymptotic behaviour of this flow at the high Reynolds numbers, examined effects of $Re$, $pe$, the solution method and the grid size on numerical solutions of $\zeta$, $\psi$ and $p$ by MIMD computings, and measured the actual performance of the parallelised square cavity program adopted each solution method on the NWT computer system.

In Section 5 we firstly indicated the process of stabilization of the eddy system and the location of the center of the primary vortex in the square cavity on the Reynolds number range of $Re = 0 \sim 10^4$, secondly comparison between results obtained by varying $pe$ in the parallel processing, thirdly comparison between results obtained by varying the solution method, and finally comparison between results obtained by varying the grid size. Its results showed that the degree of difference of $\zeta$, $\psi$ and $p$ by variation of $pe$ is negligible, but that the degree of difference of $\zeta$, $\psi$ and
by variation of the solution method is significant. Also its results showed that the degree of difference of \( \zeta \) and \( \psi \) by variation of the grid size is negligible, but that the degree of difference of \( pe \) by variation of the grid size is very significant when the grid size is huge. Since this phenomenon does not happen in the ordinary grid size, the new problem in treating the pressure Poisson equation with the Neumann boundary condition is possible to happen. We cannot make its reason clear yet. This is our answer for the problem (1) mentioned in Introduction.

In Section 6 we firstly mentioned some remarks on the parallelised square cavity programs, and measured overheads of the NWT-Fortran-compiler directives used in these parallelised programs. This is our answer for the problem (3) mentioned in Introduction. Secondly we estimated the characteristic parameters, \( (\tau, pe_{1/2}, pe, n) \), of each parallelised program on the NWT computer system in case that the actual rate is considered as a function of \( pe \) for a fixed grid size. Thirdly we estimated the characteristic parameters, \( (\tau, s_{1/2}) \), of each parallelised program on the NWT computer system in case that the actual rate is considered as a function of the grid size for a fixed \( pe \). This is our answer for the problem (4) mentioned in Introduction. Finally we measured the maximum actual performance of the parallelised square cavity program every solution method against \( pe \) on the NWT computer system, and estimated the speedup and efficiency of these programs. This is our answer for the problem (2) mentioned in Introduction.

We mention at the end that the system software of the NWT computer system had been available at the NAL during the period April to May 1994 (v2) has been drastically improved in comparison with the system software had been available during the period April to June 1993 (v1), as shown in Section 6. The degree of improvement on the actual rate \( \tau \), of v2 to v1 is at its maximum for a 256*256 grid, 2.85-fold (Jacobi (J1)), 2.84-fold (red-black), 3.73-fold (CG), and for a 255*255 grid, 1.28-fold (ADI).

9. REFERENCES


Table 1 Computational conditions used to obtain Figures 2-10

<table>
<thead>
<tr>
<th>Fig.</th>
<th>Re</th>
<th>grid</th>
<th>solution method</th>
<th>relaxation parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>$1 \times 10^{-6}$</td>
<td>256 x 256</td>
<td>$(\zeta, \psi, p):$ red - black</td>
<td>$\omega_{\zeta} = 1.4$, $\omega_{\psi} = 1.4$, $\omega_{p} = 1.5$</td>
</tr>
<tr>
<td>3</td>
<td>$1 \times 10^2$</td>
<td>256 x 256</td>
<td>$(\zeta, \psi, p):$ red - black</td>
<td>$\omega_{\zeta} = 1.4$, $\omega_{\psi} = 1.4$, $\omega_{p} = 1.5$</td>
</tr>
<tr>
<td>4</td>
<td>$1 \times 10^4$</td>
<td>256 x 256</td>
<td>$(\zeta, \psi, p):$ Jacob(11)</td>
<td>$0.1$, $0.1$, $1.0$</td>
</tr>
<tr>
<td>5</td>
<td>$2 \times 10^4$</td>
<td>256 x 256</td>
<td>$(\zeta, \psi, p):$ Jacob(11)</td>
<td>$0.1$, $0.1$, $1.0$</td>
</tr>
<tr>
<td>6</td>
<td>$1 \times 10^4$</td>
<td>1024 x 1024</td>
<td>$\zeta, \psi:$ CG - rb, $p:$ CG</td>
<td>$CG : 0.009$, $CG : 0.8$, $rb : 0.45$, $rb : 0.45$</td>
</tr>
<tr>
<td>7</td>
<td>$2 \times 10^4$</td>
<td>1024 x 1024</td>
<td>$\zeta, \psi:$ CG - rb, $p:$ CG</td>
<td>$CG : 0.006$, $CG : 0.8$, $rb : 0.35$, $rb : 0.35$</td>
</tr>
<tr>
<td>8</td>
<td>$3 \times 10^4$</td>
<td>1024 x 1024</td>
<td>$\zeta, \psi:$ CG - rb, $p:$ CG</td>
<td>$CG : 0.002$, $CG : 0.8$, $rb : 0.25$, $rb : 0.25$</td>
</tr>
<tr>
<td>9</td>
<td>$5 \times 10^4$</td>
<td>1024 x 1024</td>
<td>$\zeta, \psi:$ CG - rb, $p:$ CG</td>
<td>$CG : 0.002$, $CG : 0.8$, $rb : 0.10$, $rb : 0.09$</td>
</tr>
<tr>
<td>10</td>
<td>$6 \times 10^4$</td>
<td>1024 x 1024</td>
<td>$\zeta, \psi:$ CG - rb, $p:$ CG</td>
<td>$CG : 0.002$, $CG : 0.8$, $rb : 0.09$, $rb : 0.09$</td>
</tr>
</tbody>
</table>

Note: CG-rb indicates that we used the CG method until development of the boundary layer and that we used the red-black method for computation after development of the boundary layer.

Table 2 Values of minimum, maximum and each contour for $\zeta$, $\psi$ and $p$ in Figures 2-10

<table>
<thead>
<tr>
<th>Fig.</th>
<th>minimum $\psi$</th>
<th>maximum $\psi$</th>
<th>minimum $\zeta$</th>
<th>maximum $\zeta$</th>
<th>minimum $p$</th>
<th>maximum $p$</th>
<th>contours</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>$-1.00 \times 10^{-3}$</td>
<td>$2.00 \times 10^{-5}$</td>
<td>$-3.69 \times 10^{3}$</td>
<td>$1.41 \times 10^{3}$</td>
<td>$-5.12 \times 10^{3}$</td>
<td>$5.12 \times 10^{3}$</td>
<td>5.1 $\times 10^{3}$</td>
</tr>
<tr>
<td>3</td>
<td>$-1.02 \times 10^{-3}$</td>
<td>$1.00 \times 10^{-4}$</td>
<td>$-3.75 \times 10^{3}$</td>
<td>$1.17 \times 10^{3}$</td>
<td>$-4.700$</td>
<td>$6.292$</td>
<td>6.30 $\times 10^{3}$</td>
</tr>
<tr>
<td>4</td>
<td>$-1.14 \times 10^{-3}$</td>
<td>$1.72 \times 10^{-4}$</td>
<td>$-4.15 \times 10^{3}$</td>
<td>$2.22 \times 10^{3}$</td>
<td>$-3.461$</td>
<td>$1.814$</td>
<td>0.033 $\times 10^{3}$</td>
</tr>
<tr>
<td>5</td>
<td>$-1.19 \times 10^{-3}$</td>
<td>$2.45 \times 10^{-4}$</td>
<td>$-4.42 \times 10^{3}$</td>
<td>$2.90 \times 10^{3}$</td>
<td>$-0.172$</td>
<td>$1.567$</td>
<td>0.033 $\times 10^{3}$</td>
</tr>
<tr>
<td>6</td>
<td>$-1.22 \times 10^{-3}$</td>
<td>$3.17 \times 10^{-4}$</td>
<td>$-1.80 \times 10^{3}$</td>
<td>$1.26 \times 10^{3}$</td>
<td>$-0.142$</td>
<td>$1.653$</td>
<td>0.1 $\times 10^{3}$</td>
</tr>
<tr>
<td>7</td>
<td>$-1.21 \times 10^{-3}$</td>
<td>$3.70 \times 10^{-4}$</td>
<td>$-1.80 \times 10^{3}$</td>
<td>$1.65 \times 10^{3}$</td>
<td>$-0.068$</td>
<td>$1.540$</td>
<td>0.006 $\times 10^{3}$</td>
</tr>
<tr>
<td>8</td>
<td>$-1.26 \times 10^{-3}$</td>
<td>$5.30 \times 10^{-4}$</td>
<td>$-1.93 \times 10^{3}$</td>
<td>$1.91 \times 10^{3}$</td>
<td>$-0.097$</td>
<td>$1.648$</td>
<td>0.008 $\times 10^{3}$</td>
</tr>
<tr>
<td>9</td>
<td>$-1.18 \times 10^{-3}$</td>
<td>$4.90 \times 10^{-4}$</td>
<td>$-1.97 \times 10^{3}$</td>
<td>$2.12 \times 10^{3}$</td>
<td>$-0.086$</td>
<td>$1.610$</td>
<td>0.006 $\times 10^{3}$</td>
</tr>
<tr>
<td>10</td>
<td>$-1.17 \times 10^{-3}$</td>
<td>$5.25 \times 10^{-4}$</td>
<td>$-1.98 \times 10^{3}$</td>
<td>$2.64 \times 10^{3}$</td>
<td>$-0.089$</td>
<td>$1.654$</td>
<td>0.006 $\times 10^{3}$</td>
</tr>
</tbody>
</table>

Note1: The value of each contour for $\psi$ in Figures 2-10 is as follows: initial = $-60.0$, terminal = $69.0$, increment = $1.0$.

Note2: The value of each contour for $\zeta$ in Figures 2-10 is as follows:

<table>
<thead>
<tr>
<th>velocity</th>
<th>initial</th>
<th>terminal</th>
<th>increment</th>
</tr>
</thead>
<tbody>
<tr>
<td>primary</td>
<td>$-1.5 \times 10^{-3}$</td>
<td>$1.0 \times 10^{-3}$</td>
<td>$3.0 \times 10^{-3}$</td>
</tr>
<tr>
<td>secondary left</td>
<td>$0.0$</td>
<td>$1.0 \times 10^{-4}$</td>
<td>$1.0 \times 10^{-3}$</td>
</tr>
<tr>
<td>secondary right</td>
<td>$0.0$</td>
<td>$5.2 \times 10^{-4}$</td>
<td>$4.0 \times 10^{-4}$</td>
</tr>
<tr>
<td>tertiary</td>
<td>$-1.5 \times 10^{-4}$</td>
<td>$0.0$</td>
<td>$3.0 \times 10^{-5}$</td>
</tr>
<tr>
<td>additional</td>
<td>$-8.0 \times 10^{-6}$</td>
<td>$0.0$</td>
<td>$4.0 \times 10^{-6}$</td>
</tr>
</tbody>
</table>

Table 3 Computational conditions used to compare an effect on the number of the processing elements and the solution methods

<table>
<thead>
<tr>
<th>solution method</th>
<th>Re</th>
<th>grid</th>
<th>relaxation parameters, etc.</th>
<th>$\omega_{\zeta}$</th>
<th>$\omega_{\psi}$</th>
<th>$\omega_{p}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jacob(11)</td>
<td>$10^3$</td>
<td>256 x 256</td>
<td>$0.5$, $-0.5$, $-1.0$, $-1.0$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>red - black</td>
<td>$10^3$</td>
<td>256 x 256</td>
<td>$1.0$, $-1.0$, $-1.5$, $-1.5$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CG</td>
<td>$10^3$</td>
<td>256 x 256</td>
<td>$0.8$, $-0.8$, $-0.8$, $-0.8$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADI</td>
<td>$10^3$</td>
<td>256 x 256</td>
<td>$1.4; 5.0$, $1.6; 5.0$, $1.4; 2.0$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 4 Computational conditions used to compare an effect on the grid size for the red-black method at $Re = 10^3$, and computational results

<table>
<thead>
<tr>
<th>grid size</th>
<th>relaxation parameters</th>
<th>computational results</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\omega_x$</td>
<td>$\omega_y$</td>
</tr>
<tr>
<td>256 * 256</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>1k * 1k</td>
<td>1.60</td>
<td>1.60</td>
</tr>
<tr>
<td>2k * 2k</td>
<td>1.75</td>
<td>1.75</td>
</tr>
<tr>
<td>4k * 4k</td>
<td>1.86</td>
<td>1.86</td>
</tr>
<tr>
<td>8k * 8k</td>
<td>1.90</td>
<td>1.90</td>
</tr>
<tr>
<td>16k * 15k</td>
<td>1.93</td>
<td>1.93</td>
</tr>
</tbody>
</table>

Note: $1k = 1024$.

Table 5 Characteristic parameters of square cavity programs on the NWT computer system

(1) Jacobi method ($J_1$)

<table>
<thead>
<tr>
<th>parameter</th>
<th>grid size</th>
<th>256 * 256</th>
<th>1024 * 1024</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau_0$</td>
<td>$v_1$</td>
<td>$v_2$</td>
<td>$v_1$</td>
</tr>
<tr>
<td>$\rho$</td>
<td>$v_2$</td>
<td>$v_1$</td>
<td>$v_2$</td>
</tr>
<tr>
<td>$\zeta_{1/2}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\rho_{1/2}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\rho_{e}$</td>
<td>$v_1$</td>
<td>$v_2$</td>
<td>$v_1$</td>
</tr>
<tr>
<td>$\rho_{n}$</td>
<td>$v_2$</td>
<td>$v_1$</td>
<td>$v_2$</td>
</tr>
</tbody>
</table>

(2) red-black method

<table>
<thead>
<tr>
<th>parameter</th>
<th>grid size</th>
<th>256 * 256</th>
<th>1024 * 1024</th>
<th>2048 * 2048</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau_0$</td>
<td>$v_1$</td>
<td>$v_2$</td>
<td>$v_1$</td>
<td>$v_2$</td>
</tr>
<tr>
<td>$\rho$</td>
<td>$v_2$</td>
<td>$v_1$</td>
<td>$v_2$</td>
<td>$v_1$</td>
</tr>
<tr>
<td>$\zeta_{1/2}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\rho_{1/2}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\rho_{e}$</td>
<td>$v_1$</td>
<td>$v_2$</td>
<td>$v_1$</td>
<td>$v_2$</td>
</tr>
<tr>
<td>$\rho_{n}$</td>
<td>$v_2$</td>
<td>$v_1$</td>
<td>$v_2$</td>
<td>$v_1$</td>
</tr>
</tbody>
</table>

(3) CG method

<table>
<thead>
<tr>
<th>parameter</th>
<th>grid size</th>
<th>256 * 255</th>
<th>1024 * 1024</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau_0$</td>
<td>$v_1$</td>
<td>$v_2$</td>
<td>$v_1$</td>
</tr>
<tr>
<td>$\rho$</td>
<td>$v_2$</td>
<td>$v_1$</td>
<td>$v_2$</td>
</tr>
<tr>
<td>$\zeta_{1/2}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\rho_{1/2}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\rho_{e}$</td>
<td>$v_1$</td>
<td>$v_2$</td>
<td>$v_1$</td>
</tr>
<tr>
<td>$\rho_{n}$</td>
<td>$v_2$</td>
<td>$v_1$</td>
<td>$v_2$</td>
</tr>
</tbody>
</table>

(4) ADI method

<table>
<thead>
<tr>
<th>parameter</th>
<th>grid size = 255 * 255</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau_0$</td>
<td>$v_1$</td>
</tr>
<tr>
<td>$\rho$</td>
<td>$v_2$</td>
</tr>
<tr>
<td>$\zeta_{1/2}$</td>
<td></td>
</tr>
<tr>
<td>$\rho_{1/2}$</td>
<td></td>
</tr>
<tr>
<td>$\rho_{e}$</td>
<td>$v_1$</td>
</tr>
<tr>
<td>$\rho_{n}$</td>
<td>$v_2$</td>
</tr>
</tbody>
</table>
Table 6  Measurements of maximum actual performance of the square cavity programs on the NWT computer system (note: The values for $pe = 1$ are ones for SIMD computing)

### (1) Jacobi method (J1)

<table>
<thead>
<tr>
<th>$pe$</th>
<th>$v_{1/2}$</th>
<th>$\tau_1 (Mflop/s)$</th>
<th>$\tau_2 (Mflop/s)$</th>
<th>$\tau_3 (Mflop/s)$</th>
<th>$\tau_4 (Mflop/s)$</th>
<th>$\tau_5 (Mflop/s)$</th>
<th>$d_{ma}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>v1</td>
<td>696.34</td>
<td>760.48</td>
<td>830.94</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>v2</td>
<td>993.57</td>
<td>809.20</td>
<td>904.00</td>
<td>1439.81</td>
<td>1511.93</td>
<td>1.693</td>
</tr>
<tr>
<td>2</td>
<td>v1</td>
<td>1771.97</td>
<td>1872.66</td>
<td>1692.52</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>v2</td>
<td>1813.18</td>
<td>1614.27</td>
<td>1615.99</td>
<td>2242.01</td>
<td>3019.95</td>
<td>1.233</td>
</tr>
<tr>
<td>4</td>
<td>v1</td>
<td>3232.06</td>
<td>2871.19</td>
<td>3153.80</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>v2</td>
<td>3316.69</td>
<td>2975.14</td>
<td>3324.60</td>
<td>4338.44</td>
<td>602.88</td>
<td>1.305</td>
</tr>
<tr>
<td>8</td>
<td>v1</td>
<td>6635.02</td>
<td>5963.08</td>
<td>6836.21</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>v2</td>
<td>6948.61</td>
<td>6166.67</td>
<td>6979.90</td>
<td>8387.02</td>
<td>1203.81</td>
<td>1.270</td>
</tr>
<tr>
<td>16</td>
<td>v1</td>
<td>13145.14</td>
<td>11611.82</td>
<td>13149.84</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>v2</td>
<td>13736.69</td>
<td>12167.42</td>
<td>13720.87</td>
<td>17670.03</td>
<td>2405.97</td>
<td>1.268</td>
</tr>
<tr>
<td>32</td>
<td>v1</td>
<td>26799.56</td>
<td>23817.70</td>
<td>26638.66</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>v2</td>
<td>27385.34</td>
<td>24650.76</td>
<td>27201.76</td>
<td>35697.49</td>
<td>4791.38</td>
<td>1.309</td>
</tr>
<tr>
<td>64</td>
<td>v1</td>
<td>51103.70</td>
<td>45353.47</td>
<td>52934.53</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>v2</td>
<td>55580.43</td>
<td>47765.35</td>
<td>55689.25</td>
<td>70817.29</td>
<td>9565.78</td>
<td>1.315</td>
</tr>
<tr>
<td>128</td>
<td>v1</td>
<td>104003.26</td>
<td>91268.30</td>
<td>106884.89</td>
<td>133972.70</td>
<td>--</td>
<td>1.327</td>
</tr>
<tr>
<td></td>
<td>v2</td>
<td>107277.71</td>
<td>94212.27</td>
<td>106333.47</td>
<td>142364.71</td>
<td>19072.32</td>
<td>1.349</td>
</tr>
</tbody>
</table>

### (2) Jacobi method (J2)

<table>
<thead>
<tr>
<th>$pe$</th>
<th>$v_{1/2}$</th>
<th>$\tau_1 (Mflop/s)$</th>
<th>$\tau_2 (Mflop/s)$</th>
<th>$\tau_3 (Mflop/s)$</th>
<th>$\tau_4 (Mflop/s)$</th>
<th>$\tau_5 (Mflop/s)$</th>
<th>$d_{ma}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>v2</td>
<td>1125.67</td>
<td>891.02</td>
<td>1173.79</td>
<td>1257.17</td>
<td>324.71</td>
<td>1.119</td>
</tr>
<tr>
<td>2</td>
<td>v2</td>
<td>2285.16</td>
<td>1788.17</td>
<td>2266.42</td>
<td>2526.72</td>
<td>651.83</td>
<td>1.115</td>
</tr>
<tr>
<td>4</td>
<td>v2</td>
<td>4432.07</td>
<td>3408.93</td>
<td>4307.02</td>
<td>5001.47</td>
<td>1298.71</td>
<td>1.161</td>
</tr>
<tr>
<td>8</td>
<td>v2</td>
<td>9047.68</td>
<td>7050.29</td>
<td>8734.22</td>
<td>10119.67</td>
<td>2594.06</td>
<td>1.159</td>
</tr>
<tr>
<td>16</td>
<td>v2</td>
<td>17987.69</td>
<td>12474.39</td>
<td>17008.94</td>
<td>20150.27</td>
<td>5173.42</td>
<td>1.160</td>
</tr>
<tr>
<td>32</td>
<td>v2</td>
<td>30059.37</td>
<td>27893.81</td>
<td>31553.51</td>
<td>40400.65</td>
<td>10299.30</td>
<td>1.149</td>
</tr>
<tr>
<td>64</td>
<td>v2</td>
<td>72443.49</td>
<td>54359.25</td>
<td>68064.06</td>
<td>80356.20</td>
<td>20586.35</td>
<td>1.181</td>
</tr>
<tr>
<td>128</td>
<td>v2</td>
<td>145065.44</td>
<td>105453.77</td>
<td>134983.03</td>
<td>157598.09</td>
<td>40892.21</td>
<td>1.168</td>
</tr>
</tbody>
</table>

### (3) Red-black ordering

<table>
<thead>
<tr>
<th>$pe$</th>
<th>$v_{1/2}$</th>
<th>$\tau_1 (Mflop/s)$</th>
<th>$\tau_2 (Mflop/s)$</th>
<th>$\tau_3 (Mflop/s)$</th>
<th>$\tau_4 (Mflop/s)$</th>
<th>$\tau_5 (Mflop/s)$</th>
<th>$d_{ma}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>v1</td>
<td>601.23</td>
<td>503.22</td>
<td>576.03</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>v2</td>
<td>607.32</td>
<td>508.22</td>
<td>581.88</td>
<td>617.45</td>
<td>41.32</td>
<td>1.061</td>
</tr>
<tr>
<td>2</td>
<td>v1</td>
<td>1206.57</td>
<td>999.34</td>
<td>1149.04</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>v2</td>
<td>1206.04</td>
<td>1013.47</td>
<td>1152.15</td>
<td>1242.81</td>
<td>82.49</td>
<td>1.070</td>
</tr>
<tr>
<td>4</td>
<td>v1</td>
<td>2356.78</td>
<td>1952.26</td>
<td>2253.02</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>v2</td>
<td>2387.37</td>
<td>1979.46</td>
<td>2279.09</td>
<td>2472.13</td>
<td>164.87</td>
<td>1.085</td>
</tr>
<tr>
<td>8</td>
<td>v1</td>
<td>4786.64</td>
<td>3990.67</td>
<td>4554.39</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>v2</td>
<td>4753.20</td>
<td>3969.34</td>
<td>4571.29</td>
<td>4853.45</td>
<td>329.49</td>
<td>1.084</td>
</tr>
<tr>
<td>16</td>
<td>v1</td>
<td>9475.17</td>
<td>7863.72</td>
<td>9050.39</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>v2</td>
<td>9471.13</td>
<td>7840.44</td>
<td>8967.29</td>
<td>9811.55</td>
<td>658.48</td>
<td>1.105</td>
</tr>
<tr>
<td>32</td>
<td>v1</td>
<td>19016.27</td>
<td>15747.52</td>
<td>17844.49</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>v2</td>
<td>18775.89</td>
<td>15625.85</td>
<td>17839.21</td>
<td>19801.77</td>
<td>1307.84</td>
<td>1.108</td>
</tr>
<tr>
<td>64</td>
<td>v1</td>
<td>37518.93</td>
<td>30776.77</td>
<td>35333.05</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>v2</td>
<td>35920.60</td>
<td>30696.73</td>
<td>34799.87</td>
<td>39311.40</td>
<td>2630.01</td>
<td>1.130</td>
</tr>
<tr>
<td>128</td>
<td>v1</td>
<td>74168.23</td>
<td>66702.51</td>
<td>68553.69</td>
<td>76721.15</td>
<td>--</td>
<td>1.114</td>
</tr>
<tr>
<td></td>
<td>v2</td>
<td>72604.31</td>
<td>69932.71</td>
<td>69144.19</td>
<td>79179.64</td>
<td>5234.37</td>
<td>1.162</td>
</tr>
</tbody>
</table>
### Table 6 Continued

(4) **CG method**

<table>
<thead>
<tr>
<th>pe</th>
<th>( v1/2 )</th>
<th>( r_1(M/flop/s) )</th>
<th>( r_2(M/flop/s) )</th>
<th>( r_3(M/flop/s) )</th>
<th>( r_4(M/flop/s) )</th>
<th>( r_5(M/flop/s) )</th>
<th>( d_{ma} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( v1 )</td>
<td>513.22</td>
<td>823.35</td>
<td>885.91</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>( v2 )</td>
<td>518.92</td>
<td>824.03</td>
<td>908.76</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>2</td>
<td>( v1 )</td>
<td>1643.34</td>
<td>1510.85</td>
<td>1683.49</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>( v2 )</td>
<td>1732.66</td>
<td>1507.70</td>
<td>1744.82</td>
<td>1841.04</td>
<td>297.44</td>
<td>1.055</td>
</tr>
<tr>
<td>4</td>
<td>( v1 )</td>
<td>3427.48</td>
<td>3100.03</td>
<td>3435.19</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>( v2 )</td>
<td>3556.33</td>
<td>3187.51</td>
<td>3563.72</td>
<td>3739.03</td>
<td>592.99</td>
<td>1.049</td>
</tr>
<tr>
<td>8</td>
<td>( v1 )</td>
<td>6388.46</td>
<td>5727.53</td>
<td>6361.37</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>( v2 )</td>
<td>6689.05</td>
<td>6041.97</td>
<td>6645.19</td>
<td>7051.08</td>
<td>1183.89</td>
<td>1.061</td>
</tr>
<tr>
<td>16</td>
<td>( v1 )</td>
<td>13193.54</td>
<td>11825.71</td>
<td>13114.86</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>( v2 )</td>
<td>15827.01</td>
<td>12415.07</td>
<td>13751.11</td>
<td>14547.81</td>
<td>2360.67</td>
<td>1.068</td>
</tr>
<tr>
<td>32</td>
<td>( v1 )</td>
<td>25836.20</td>
<td>23132.94</td>
<td>25571.54</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>( v2 )</td>
<td>27605.42</td>
<td>24219.60</td>
<td>26804.95</td>
<td>28511.96</td>
<td>4701.43</td>
<td>1.064</td>
</tr>
<tr>
<td>64</td>
<td>( v1 )</td>
<td>51665.29</td>
<td>46009.02</td>
<td>49594.88</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>( v2 )</td>
<td>53728.95</td>
<td>48068.44</td>
<td>53500.58</td>
<td>56370.13</td>
<td>9352.25</td>
<td>1.064</td>
</tr>
<tr>
<td>128</td>
<td>( v1 )</td>
<td>100915.97</td>
<td>88091.34</td>
<td>100005.14</td>
<td>105547.36</td>
<td>--</td>
<td>1.055</td>
</tr>
<tr>
<td></td>
<td>( v2 )</td>
<td>103640.27</td>
<td>92122.34</td>
<td>103110.37</td>
<td>110265.77</td>
<td>18591.26</td>
<td>1.065</td>
</tr>
</tbody>
</table>

(5) **ADI method**

<table>
<thead>
<tr>
<th>pe</th>
<th>( v1/2 )</th>
<th>( r_1(M/flop/s) )</th>
<th>( r_2(M/flop/s) )</th>
<th>( r_3(M/flop/s) )</th>
<th>( r_4(M/flop/s) )</th>
<th>( r_5(M/flop/s) )</th>
<th>( d_{ma} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( v1 )</td>
<td>642.44</td>
<td>612.00</td>
<td>649.47</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>( v2 )</td>
<td>643.59</td>
<td>612.92</td>
<td>651.15</td>
<td>702.02</td>
<td>260.47</td>
<td>1.078</td>
</tr>
<tr>
<td>2</td>
<td>( v1 )</td>
<td>326.71</td>
<td>316.23</td>
<td>250.06</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>( v2 )</td>
<td>344.68</td>
<td>344.09</td>
<td>351.03</td>
<td>423.63</td>
<td>243.69</td>
<td>1.207</td>
</tr>
<tr>
<td>4</td>
<td>( v1 )</td>
<td>514.46</td>
<td>515.93</td>
<td>469.31</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>( v2 )</td>
<td>520.21</td>
<td>518.79</td>
<td>539.02</td>
<td>654.02</td>
<td>376.36</td>
<td>1.213</td>
</tr>
<tr>
<td>8</td>
<td>( v1 )</td>
<td>757.19</td>
<td>754.89</td>
<td>689.88</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>( v2 )</td>
<td>809.08</td>
<td>791.01</td>
<td>792.26</td>
<td>905.72</td>
<td>552.00</td>
<td>1.144</td>
</tr>
<tr>
<td>16</td>
<td>( v1 )</td>
<td>1189.79</td>
<td>1194.45</td>
<td>1131.48</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>( v2 )</td>
<td>1251.69</td>
<td>1236.71</td>
<td>1294.24</td>
<td>1515.69</td>
<td>758.25</td>
<td>1.172</td>
</tr>
<tr>
<td>32</td>
<td>( v1 )</td>
<td>1591.71</td>
<td>1618.20</td>
<td>1480.48</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>( v2 )</td>
<td>1769.16</td>
<td>1781.88</td>
<td>1722.00</td>
<td>1893.16</td>
<td>972.87</td>
<td>1.099</td>
</tr>
<tr>
<td>64</td>
<td>( v1 )</td>
<td>2483.59</td>
<td>2567.48</td>
<td>2390.18</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>( v2 )</td>
<td>2645.45</td>
<td>2687.21</td>
<td>2788.38</td>
<td>3031.59</td>
<td>1264.77</td>
<td>1.084</td>
</tr>
<tr>
<td>128</td>
<td>( v1 )</td>
<td>3323.38</td>
<td>3391.06</td>
<td>3143.13</td>
<td>3253.06</td>
<td>--</td>
<td>1.045</td>
</tr>
<tr>
<td></td>
<td>( v2 )</td>
<td>3178.33</td>
<td>3202.02</td>
<td>3671.87</td>
<td>3930.80</td>
<td>1599.50</td>
<td>1.071</td>
</tr>
</tbody>
</table>

Table 7 Grid size used to obtain Table 6 and Figure 23

<table>
<thead>
<tr>
<th>pe</th>
<th>( Jacobi(J1/J2) )</th>
<th>( red - black )</th>
<th>( CG )</th>
<th>( AD1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1,500 * 1,500</td>
<td>1,024 * 1,024</td>
<td>1,025 * 1,025</td>
<td>1,025 * 1,025</td>
</tr>
<tr>
<td>2</td>
<td>2,048 * 2,048</td>
<td>1,500 * 1,500</td>
<td>1,501 * 1,501</td>
<td>1,501 * 1,501</td>
</tr>
<tr>
<td>4</td>
<td>3,000 * 3,000</td>
<td>2,048 * 2,048</td>
<td>2,049 * 2,049</td>
<td>2,049 * 2,049</td>
</tr>
<tr>
<td>8</td>
<td>4,096 * 4,096</td>
<td>3,000 * 3,000</td>
<td>3,001 * 3,001</td>
<td>3,001 * 3,001</td>
</tr>
<tr>
<td>16</td>
<td>5,800 * 5,800</td>
<td>4,096 * 4,096</td>
<td>4,097 * 4,097</td>
<td>4,097 * 4,097</td>
</tr>
<tr>
<td>32</td>
<td>8,192 * 8,192</td>
<td>5,800 * 5,800</td>
<td>5,801 * 5,800</td>
<td>5,801 * 5,800</td>
</tr>
<tr>
<td>64</td>
<td>12,000 * 12,000</td>
<td>8,192 * 8,192</td>
<td>8,193 * 8,193</td>
<td>8,193 * 8,193</td>
</tr>
<tr>
<td>128</td>
<td>16,384 * 16,384</td>
<td>12,000 * 12,000</td>
<td>12,001 * 12,001</td>
<td>12,001 * 12,001</td>
</tr>
</tbody>
</table>
Table 8 Comparison between the values of $S_p$ computed from (63) and (69) when the grid size is 1,024 x 1,024.

<table>
<thead>
<tr>
<th>pc</th>
<th>Jacobi(J1) (63)</th>
<th>Jacobi(J1) (69)</th>
<th>red-black (63)</th>
<th>red-black (69)</th>
<th>CG (63)</th>
<th>CG (69)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1.98</td>
<td>1.95</td>
<td>2.00</td>
<td>1.95</td>
<td>1.69</td>
<td>1.65</td>
</tr>
<tr>
<td>4</td>
<td>3.57</td>
<td>3.77</td>
<td>3.89</td>
<td>3.80</td>
<td>2.63</td>
<td>2.75</td>
</tr>
<tr>
<td>8</td>
<td>7.17</td>
<td>7.05</td>
<td>7.38</td>
<td>7.18</td>
<td>5.02</td>
<td>4.86</td>
</tr>
<tr>
<td>16</td>
<td>12.17</td>
<td>12.72</td>
<td>12.38</td>
<td>12.84</td>
<td>8.93</td>
<td>8.54</td>
</tr>
<tr>
<td>64</td>
<td>27.38</td>
<td>27.57</td>
<td>30.01</td>
<td>27.85</td>
<td>19.07</td>
<td>17.59</td>
</tr>
<tr>
<td>128</td>
<td>30.10</td>
<td>28.58</td>
<td>32.23</td>
<td>28.28</td>
<td>17.21</td>
<td>15.19</td>
</tr>
</tbody>
</table>

Note: The value of $VTC_1$ is 0.988(Jacobi(J1)), 0.988(red-black) and 0.800(CG), respectively.

Table 9 Speedup and efficiency for the case of $\tau_{min}$ and $v_2$ of Table 6 provided that $VTC_1 = 1.0$.

<table>
<thead>
<tr>
<th>pc</th>
<th>Jacobi(J1) $S_{pe}$</th>
<th>Jacobi(J2) $S_{pe}$</th>
<th>red-black $S_{pe}$</th>
<th>CG $S_{pe}$</th>
<th>ADI $S_{pe}$</th>
<th>$E_{pe}$</th>
<th>$E_{pe}$</th>
<th>$E_{pe}$</th>
<th>$E_{pe}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1.967</td>
<td>1.966</td>
<td>1.959</td>
<td>1.915</td>
<td>0.726</td>
<td>0.363</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3.903</td>
<td>3.903</td>
<td>3.886</td>
<td>3.799</td>
<td>1.143</td>
<td>0.288</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>7.774</td>
<td>7.716</td>
<td>7.713</td>
<td>7.503</td>
<td>2.414</td>
<td>0.302</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>15.328</td>
<td>15.296</td>
<td>15.153</td>
<td>14.816</td>
<td>2.647</td>
<td>0.184</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>30.318</td>
<td>30.411</td>
<td>29.950</td>
<td>29.164</td>
<td>6.708</td>
<td>0.210</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>58.374</td>
<td>60.048</td>
<td>58.540</td>
<td>57.047</td>
<td>7.336</td>
<td>0.115</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>128</td>
<td>116.643</td>
<td>117.354</td>
<td>113.860</td>
<td>110.631</td>
<td>22.750</td>
<td>0.179</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Fig. 1 Cavity flow configuration, coordinates, nomenclature and boundary conditions
Fig. 2 Square cavity flows at $Re = 10^6$

Fig. 3 Square cavity flows at $Re = 10^2$

Fig. 4 Square cavity flows at $Re = 10^3$
Parallel Computations of Incompressible Viscous Flow in a Lid-driven Square Cavity

Fig. 5 Square cavity flows at $Re = 2 \times 10^3$

Fig. 6 Square cavity flows at $Re = 10^4$

Fig. 7 Square cavity flows at $Re = 2 \times 10^4$
Fig. 8 Square cavity flows at $Re = 3 \times 10^4$ (a snapshot)

Fig. 9 Square cavity flows at $Re = 5 \times 10^4$ (a snapshot)

Fig. 10 Square cavity flows at $Re = 6 \times 10^4$ (a snapshot)
Fig. 11 Effect of Re on profiles of $u$, $v$, $\zeta$ and $p$ through center of primary vortex (note: The vertical and horizontal beelines indicate the values of $x$ and $y$ of the center of the primary vortex for each $Re$)
Fig. 12 Location of center of primary vortex as a function of Reynolds number
Fig. 13 Effect of $pe$ on profiles of $p$ through center of primary vortex in case of $Re = 10^3$ and the CG method

Fig. 14 Effect of the solution method on profiles of $p$ through center of primary vortex in case of $Re = 10^3$ and $pe = 16$
Fig. 15 Effect of grid size on profile of $u$, $v$, $\zeta$ and $p$ through center of primary vortex in case of $Re = 10^3$ and the red-black method.
Fig. 16 Effect of the grid size on total pressure contours in case of $Re = 10^3$ and the red-black method ($1k = 1024$)
Fig. 17 Overheads for the NWT-Fortran-compiler directives against the number of processing elements
Parallel Computations of Incompressible Viscous Flow in a Lid-driven Square Cavity

Fig. 17 Continued

(3) SPREAD DO plus END SPREAD
(note: This result is not very precise due to the indirect measurement.)

(4) SPREAD DO plus END SPREAD SUM(abc)
(note: This result is not very precise due to the indirect measurement.)
(5) SPREAD MOVE plus END SPREAD plus MOVEWAIT (note: This result indicates the mean values of six data obtained when an array data of the dimension shown above is transferred between the global and local memory spaces.)

(6) OVERLAPFIX plus MOVEWAIT (note: This indicates the results when one or two arrays of the overlap range = \((1,1)\) are specified in the OVERLAPFIX statement.)

Fig. 17 Continued
Fig. 18 Actual performance $\tau_2$ of the square cavity program used the Jacobi method (J1).
Fig. 19 Actual performance $\tau_2$ of the square cavity program used the red-black method
(3) Grid size = 2048x2048.

(4) Parameter = grid size.
Fig. 20 Actual performance $\tau_2$ of the square cavity program used the CG method

(1) Grid size $= 256 \times 256$.

(2) Grid size $= 1024 \times 1024$. 

This document is provided by JAXA.
Parallel Computations of Incompressible Viscous Flow in a Lid-driven Square Cavity

Fig. 21 Actual performance $\tau_1$ of the square cavity program used the ADI method

Fig. 22 Actual performance $\tau_3$ of the square cavity program against the grid size ($pe = 128$)
Fig. 23 Measured maximum of actual performance $\tau_3$ of the square cavity programs against $pe$
(note: The values for $pe = 1$ are ones for SIMD computing)