9

非圧縮性流の一計算法

賈　　為*　中村佳朗*　保原　充*

An Incompressible Flow Solver

by

Wei JIA, Yoshiaki NAKAMURA and Michiru YASUHARA
Department of Aeronautical Engineering
Nagoya University, Nagoya, 464-01, Japan

ABSTRACT

This paper presents an incompressible flow solver based on the primitive variable formulation and its several applications to both fundamental and practical flow problems. The coupled form of the momentum and energy equations written in generalized coordinates are solved by the FDM (finite difference method), where a two-step time integration method with second order accuracy, the approximation of convective terms by the QUICK method, and an accurate, fast convergence pressure Poisson solver are employed, as well as an efficient multi-domain technique. Studies on flows on a flat plate and around a circular cylinder verified the accuracy and efficiency of the present procedures. Treatments of flow problems with complicated geometry are also presented for several examples.

1. INTRODUCTION

Two numerical procedures to solve the unsteady incompressible Navier-Stokes equations in both primitive and $\psi-\omega$ variable forms are developed. The primitive variable procedure uses the coupled form of the momentum and energy equations as its governing equations and can treat not only pure flow problems but also problems with heat convection. The QUICK [1] algorithm extended to generalized coordinates [2, 3] is applied to model the convective terms and the time integration is performed by a two-step method with second order accuracy [4]. The pressure in the former procedure is simultaneously calculated between these two steps by solving the pressure Poisson equation with Neumann boundary condition. A new numerical method [5] is developed to do this which can obtain a fully converged pressure field at every time step. Furthermore, the primitive variable methods is coupled with an efficient Multi-Domain Technique [3, 6]. It can treat both external and internal flows around complicated multiple bodies by employing the multiple grid system containing overlapped regions.

Section 2 describes the governing equations, and section 3 the numerical treatments. In section 4, several examples of numerical results are shown.

2. GOVERNING EQUATIONS

Two kinds of the governing equations are considered; one is the primitive variable formulation and another is the $\psi-\omega$ formulation.

* 名古屋大学

2-1 Primitive Variable Formulation

The time dependent governing equations in the primitive variable form for incompressible flows including heat convection and buoyancy force by the Boussinesq approximation are written in generalized coordinates. For simplicity, the 2D equations are shown.

$$\hat{q}_t + E_\xi + \hat{F}_\eta = \frac{1}{Re^*}(\hat{R}_\xi + \hat{S}_\eta) + \frac{Gr}{Re^2}\hat{Y} \qquad (1)$$

$$\hat{q} = J^{-1}\begin{bmatrix}0\\u\\v\\\theta\end{bmatrix},\ \hat{E} = J^{-1}\begin{bmatrix}U\\uU+\xi_x p\\vU+\xi_y p\\\theta U\end{bmatrix},$$

$$\hat{F} = J^{-1}\begin{bmatrix}V\\uV+\eta_x p\\vV+\eta_y p\\\theta V\end{bmatrix},\ \hat{R} = J^{-1}\begin{bmatrix}0\\g_1u_\xi+g_2u_\eta\\g_1v_\xi+g_2v_\eta\\g_1\theta_\xi+g_2\theta_\eta\end{bmatrix},$$

$$\hat{S} = J^{-1}\begin{bmatrix}0\\g_2u_\xi+g_3u_\eta\\g_2v_\xi+g_3v_\eta\\g_2\theta_\xi+g_3\theta_\eta\end{bmatrix},\ \hat{Y} = J^{-1}\begin{bmatrix}0\\\theta e_x\\\theta e_y\\0\end{bmatrix}$$

$$U = \xi_x u + \xi_y v\ ,\quad V = \eta_x u + \eta_y v,$$
$$J = \xi_x\eta_y - \xi_y\eta_x\ ,\ g_1 = \xi_x^2 + \xi_y^2\ ,$$
$$g_2 = \xi_x\eta_x + \xi_y\eta_y\ ,\ g_3 = \eta_x^2 + \eta_y^2\ ,$$

where u and v are the velocity components in the x and y directions, respectively, and U and V the contravariant velocity components in the ξ and η directions, respectively. p is the pressure, θ the non-dimensional temperature, Re the Reyolds number and Gr the Grashof number. For convenience, Re^* is used to represent the Reynolds number in the momentum equations and the Peclet number ($Pe = Pr \cdot Gr$) in the energy equation. J is the transformation Jacobian and g_1, g_2, g_3 the transformation metrics.

In most cases where stream velocities are large compared with the buoyancy induced ones, the buoyancy effect is negligible, which leads to the case with forced convection. In such cases, the momentum equations are independent of the energy equation and can be solved without any information of temperature field. The temperature field is obtained from the velocity field along with the temperature distribution at the previous time step by solving the energy equation at every time step.

However, in the case of natural or mixed convection, where the buoyancy effect is dominant or significant, the momentum and energy equations are coupled and must be solved simultaneously at every time step.

When dealing with a natural convection flow, the characteristic velocity is not known, so that the Reynolds number is not determined. According to Ostrach [7], the Reynolds number may take the following values in a well balanced manner:

$$\begin{aligned}
&a.\ Gr \le 1\ and\ Ra \le 1 :\ Re = Gr\\
&b.\ Gr^{\frac{1}{2}} \ge 1\ and\ Pr \le 1 :\ Re = Gr^{\frac{1}{2}}\\
&c.\ Gr^{\frac{1}{2}} \ge 1\ and\ Pr > 1 :\ Re = (Gr/Pr)^{\frac{1}{2}}
\end{aligned} \qquad (2)$$

Incidentally, Ra denotes the Rayleigh number ($Ra = Pr \cdot Gr$).

2.2 $\psi - \omega$ Formulation [8]

The two-dimensional time dependent governing equations in the $\psi - \omega$ form for pure incompressible flow problems are written in generalized coordinates as follows:

$$(J^{-1}\omega)_t + (J^{-1}U\omega)_\xi + (J^{-1}V\omega)_\eta = Re^{-1}J^{-1}\nabla^2\omega$$
$$\nabla^2\psi = -\omega \qquad (3)$$

where ψ represents the stream function and ω the vorticity. The velocities are obtained by taking the derivatives of the stream function as follows:

$$\begin{aligned}
u &= \psi_y = \psi_\xi\xi_y + \psi_\eta\eta_y\\
v &= -\psi_x = -\psi_\xi\xi_x - \psi_\eta\eta_x
\end{aligned} \qquad (4)$$

The pressure field is calculated from the velocity field by solving the following Poisson equation.

$$\nabla^2 p = 2J(u_\xi v_\eta - v_\xi u_\eta) \qquad (5)$$

3. NUMERICAL TECHNIQUES

3.1 Approximation of Spatial Derivatives

The above governing equations are then discretized by the FDM and solved numerically. A regular mesh system is employed for easy treatment

of complicated geometry problems. It comes from the conservation form of the governing equations, where the control volume integrations of the convective and viscous terms are reduced to surface integrations on the control surfaces. In viscous terms, quantities on control surfaces are approximated by arithmetic averages. However in convective terms, a higher order upwind technique based on the QUICK method extended to generalized coordinates [1-3] is applied to suppress instability at higher Reynolds numbers. Incidentally, the transformation metrics are calculated by second order central differences except for the normal derivatives at boundaries which are calculated by one-sided differences with the same order accuracy.

3.2 Time Integration Method [4]

The time integration is performed by a two-step method. Here it is illustrated for the primitive variable formulation. Let q be the solution vector, H the convective terms, X the viscous terms, and Y the buoyancy force terms. Then the time integration is expressed in the following:

The first step is

$$\frac{q^* - q^n}{\Delta t} = \frac{3}{2}H^n - \frac{1}{2}H^{n-1} + \frac{1}{2}X^n + \frac{1}{2}Y^n \qquad (6)$$

The second step is

$$\frac{q^{n+1} - q^*}{\Delta t} = -\nabla p^{n+\frac{1}{2}} + \frac{1}{2}X^{n+1} + \frac{1}{2}Y^{n+1} \qquad (7)$$

The pressure Poisson equation is derived by taking the divergence of the second step equation (7), assuming $\nabla \cdot \vec{v}^{n+1} = 0$.

$$\nabla^2 p^{n+\frac{1}{2}} = \frac{1}{\Delta t}\nabla \cdot \vec{v}^* + \frac{1}{2}\frac{Gr}{Re^2}\nabla \cdot (\theta^{n+1}\vec{e}) \qquad (8)$$

The pressure is calculated by solving Eq. (8) between these two steps.

To summarize the time integration steps, we have the following order of calculation.

$$(\theta^n\ ,\ \vec{v}^n) \Longrightarrow (\theta^*\ ,\ \vec{v}^*) \Longrightarrow (\theta^{n+1}) \Longrightarrow (p^{n+1/2}) \Longrightarrow (\vec{v}^{n+1}) \qquad (9)$$

3.3 Pressure Poisson Solver

When the primitive variable method is employed, the pressure boundary conditions must be solved at every time step. In general, the pressure boundary conditions of incompressible flows are obtained from the normal direction momentum equations at boundaries. These result in Neumann type boundary conditions. It is well known that the direct application of a point relaxation method such as the SOR method to solve the Poisson equation with Newmann boundary conditions coverges very slowly.

Another problem arises when Neumann conditions are imposed at all boundaries, where the following constraint must be satisfied to ensure the existence of solutions [9].

$$\int_R DdR = \int_\Gamma (\partial p/\partial n)ds \qquad (10)$$

where D denotes the source of the Poisson equation, n the normal direction to the boundary, R the computational region or its area, and Γ the boundary. This constraint is usually not satisfied due to numerical errors, which lead to a non-converged solution. To automatically satisfy this constraint, a source modification method [9, 10] has been proposed.

At first, the following value is calculated by numerical integration over the whole computational region.

$$E = \int_R DdR - \int_\Gamma (\partial p/\partial n)ds \qquad (11)$$

Then the source of the Poisson equation is modified by subtracting this small value, E/R from the original source term D.

$$\nabla^2 p = D - E/R \qquad (12)$$

However, it is not expected to remarkably improve the convergence of the SOR method by such a modification. Actually, an application of the SOR method to the Poisson equation with Neumann condition at a part of boundary also suffers very slow convergence.

A new method is developed here to overcome

this deficiency, which can produce a rapid convergence for almost all cases. It comes from the linearities of the equation and boundary condition, where the Poisson equation with Neumann boundary conditions can be divided into two equations. One is the Poisson equation with Dirichlet boundary conditions at the whole bounday, and another is the Laplace equation with new Neumann boundary conditions. As an example, let the original equation and boundary condition be

$$\begin{aligned} &\nabla^2 p = D \\ &\partial p/\partial n = S \qquad on\ \Gamma \end{aligned} \tag{13}$$

Eq. (13) is divided into two equations

$$\begin{aligned} &\nabla^2 p_1 = D \\ &p_1 = 0 \qquad on\ \Gamma \end{aligned} \tag{14}$$

$$\begin{aligned} &\nabla^2 p_2 = 0 \\ &\partial p_2/\partial n = S - \partial p_1/\partial n \qquad on\ \Gamma \end{aligned} \tag{15}$$

The constraint on Eq. (15) can automatically be satisfied by modifying the source of Eq. (14), following Eq. (12). Thus the solution of Eq. (13) is obtained as the sum of the two solutions, p_1 and p_2.

$$p = p_1 + p_2 \tag{16}$$

Since the boundary condition for Eq. (14) is the Dirichlet type, the application of the SOR method produces a rapid convergence. Once Eq. (14) is solved, the boundary condition for Eq. (15) is obtained by evaluating the normal derivative of p_1 at the boundary. Since Eq. (15) is a Laplace equation, it can be transformed into a boundary integral form.

$$c\, p_2 + \frac{1}{2\pi}\int_\Gamma p_2 \frac{\partial}{\partial n}\left(ln\frac{1}{r}\right) d\Gamma \tag{17}$$

$$-\frac{1}{2\pi}\int_\Gamma \frac{\partial p_2}{\partial n}\left(ln\frac{1}{r}\right) d\Gamma = 0 \quad ,$$

$$c = \begin{cases} 1 & in\ R \\ \dfrac{1}{2\pi}\alpha & on\ \Gamma \end{cases}$$

where α denotes the inner angle of the boundary. The above boundary integral equation can be solved by the Bounday Element Method (BEM).

In the present study, the BEM with linear elements is applied. The grid points at the boundary for the FDM are taken as the boundary nodes of the BEM. The integral equation (17) is discretized to obtain the system of linear equations, and the unknown values are solved by a vectorized LU decomposition procedure.

Since all the boundary values and the normal derivatives at the boundary have thus been known, the values at the interior grid points can be calculated by the discretized boundary integral formulation (17). However, as the number of the interior grid points is about the square of the boundary grid point number, this process is time-consuming. An alternative way which can remarkably reduce the CPU time is to solve the Laplace equation with the calculated Dirichlet boundary condition by the SOR method.

From now on, the method which solves Eq. (14) by the SOR method, the boundary values of Eq. (15) by the BEM, and the values at the interior grid points of Eq. (15) by the BEM is named the SBB (SOR-BEM-BEM) method, and the method which is the same as the SBB method except that it solves the values at the interior grid points of Eq. (15) by the SOR method is named the SBS (SOR-BEM-SOR) method.

3.4 Multi-Domain Technique [3]

To simulate problems with complicated geometry, an efficient Multi-Domain Technique is also developed. A multiple grid system can be employed and combined with overlapped regions, where the connection at each grid boundary is made by a triangular interpolation method during iteration processes at every time step. As an example, two overlapped regions marked by I and II are considered here (see Fig. 1). The value at the boundary point of region II denoted by the star mark is interpolated by the neighboring three points in region I denoted by the solid circle mark. The same treatment is made at the connection boundary of region I. The

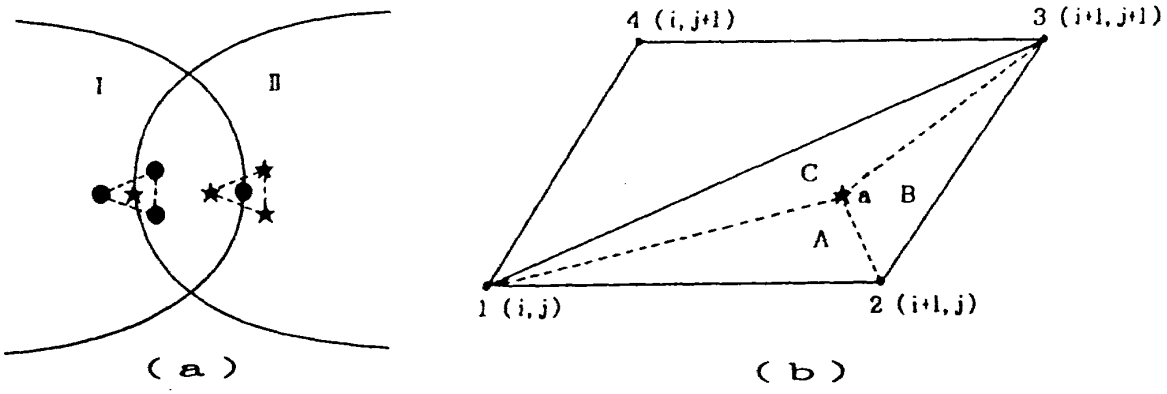


Fig. 1 Connection of two overlapped regions: regions I and II. (a) Interpolation of the point at connection boundary, and (b) the way of searching for the interpolated point and the points used for interpolation.

interpolation function is the linear function widely used in the finite element method,

$$u = \phi_1 u_1 + \phi_2 u_2 + \phi_3 u_3 \tag{18}$$

where

$$\phi_1 = S_B/S \ , \ \phi_2 = S_C/S \ , \ \phi_3 = S_A/S ,$$

$$2S = \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix} \ , \ 2S_A = \begin{vmatrix} 1 & x & y \\ 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \end{vmatrix} ,$$

$$2S_B = \begin{vmatrix} 1 & x & y \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix} \ , \ 2S_C = \begin{vmatrix} 1 & x & y \\ 1 & x_3 & y_3 \\ 1 & x_1 & y_1 \end{vmatrix} .$$

In a computer program, the interpolated points and points used for interpolations can automatically be searched by satisfying the following relation.

$$|\phi_1| + |\phi_2| + |\phi_3| = 1 . \tag{19}$$

This is swept over all quadrilateral mesh cells which consist of two triangles. The addresses associated with the points are memorized and referenced in the computer program.

4. RESULTS

At first, two fundamental flow problems are examined here to check the accuracy and efficiency of the calculation procedures. Then, numerical results for several different flow problems are presented.

Figure 2 shows the calculated results (dashed lines) of the viscous flow on a flat plate with a leading edge ($Re = 10^4$) by both the primitive variable and the $\psi - \omega$ method, compared with the Blasius' analytic solution (solid lines). Good agreement is obtained between the two solutions. However, unlike the Blasius' boundary layer solution, a pressure peak appears at the leading edge in the numerical solution of the Navier-Stokes equations.

Figure 3 shows the velocity vectors around a circular cylinder ($Re = 10^3$) calculated by the $\psi - \omega$ method, and the pressure is then calculated from the velocity field by solving the pressure Poisson equation. In Fig. 4, the convergence history of the SOR method is plotted, where the L_2 residual is used as the monitor of convergence. It is seen that the Poisson equation with Neumann boundary condition converges very slowly. In contrast to this, a fairly rapid convergence is obtained for the alternative one with Dirichlet boundary condition.

The pressure distribution obtained by the SOR method for two iteration numbers are plotted in Fig. 5. The distribution becomes unchanged when the iteration number goes over 5000, which corresponds to the L_2 residual of 7.34×10^{-3}. For comparison, the pressure distribution by the SBS method is plotted in Fig. 6, which shows good agreement with the final result by the SOR method. By use of the SBS method, the L_2 residual is made smaller than 10^{-4} at the iteration number of about 100 in its first and third steps. Table 1 gives a

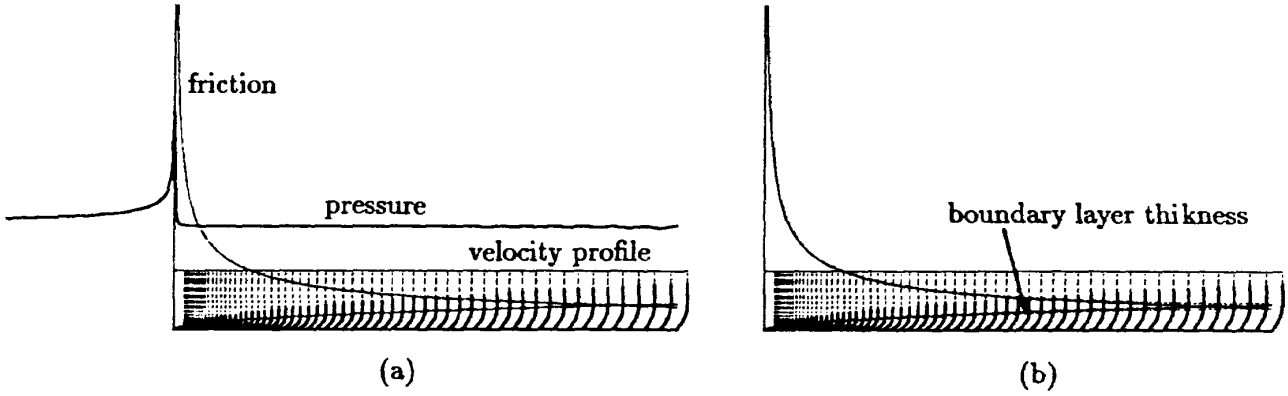


Fig. 2 Flow on flat plate with leading edge. (a) the primitive variable method and (b) $\psi - \omega$ method.

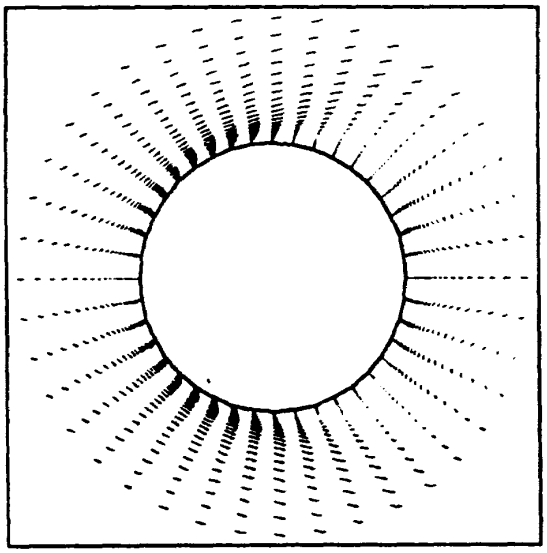


Fig. 3 Velocity vectors around circular cylinder: $\psi - \omega$ method, Re = 103, and t = 5.

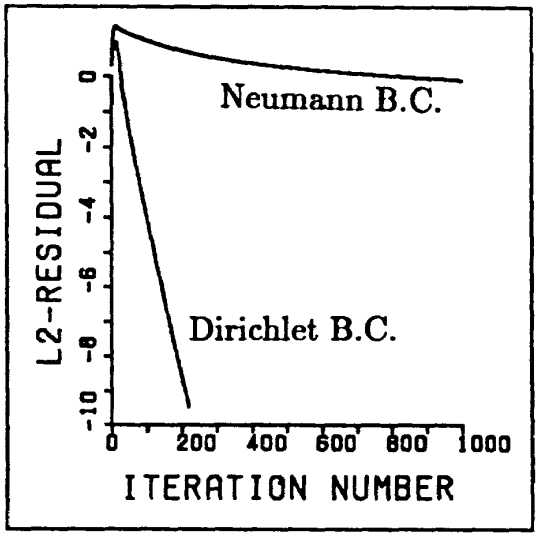


Fig. 4 Comparison of convergence history of the SOR method applied to Poisson equations with Neumann boundary condition and with Dirichlet boundary condition.

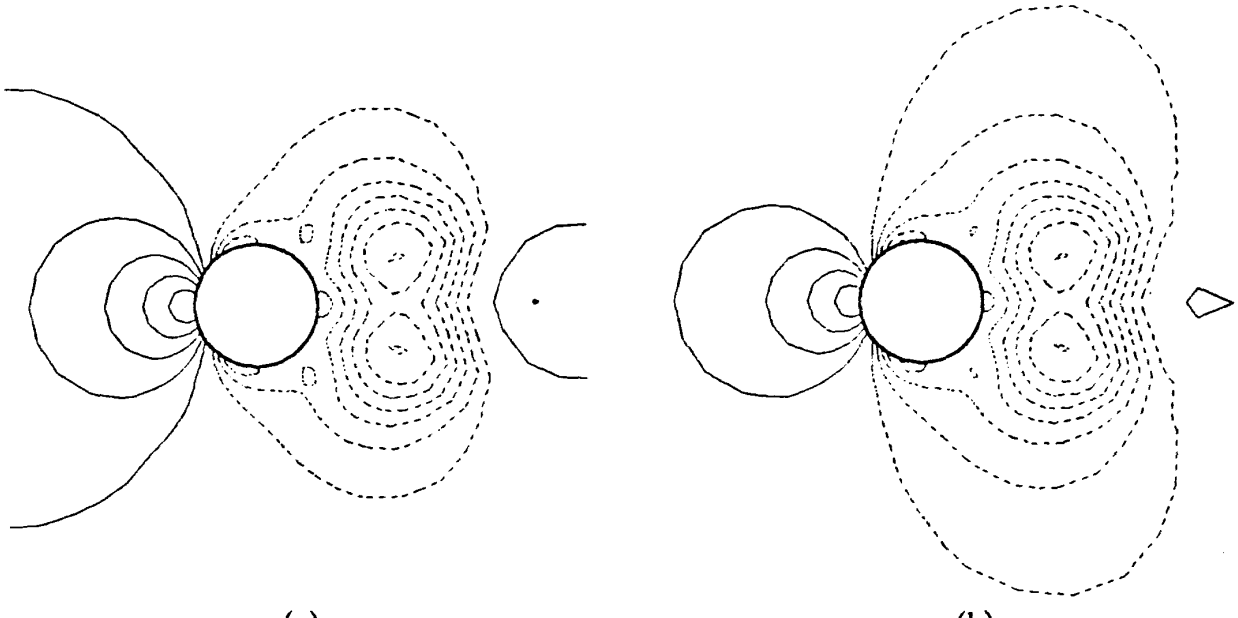


Fig. 5 Pressure distribution around circular cylinder obtained by the SOR method. Iteration number is (a) 1000, and (b) 5000 (L_2 = 7.34 x 10^{-3}).

Table 1 Comparison of three methods (SOR, SBB, SBS) tested on FACOM M-782, where the L_2 residual and iteration number correspond to the two parts calculated with the SOR method. Incidentally, ( ) + ( ) + ( ) denotes the iteration number or the CPU time needed to calculate p_1, p_2 at boundary points, and p_2 at interior points.

	L_2 Residual	Iteration Number	CPU Time ($\times 10^{-3}$ sec)	Acceleration Rate
SOR method	7.34×10^{-3}	5000 (given)	16104	1
SBB method	10^{-4} (given)	101	320 + 109 + 2203	6.1
SBS method	10^{-4} (given)	101 + 107	320 + 109 + 325	21.3

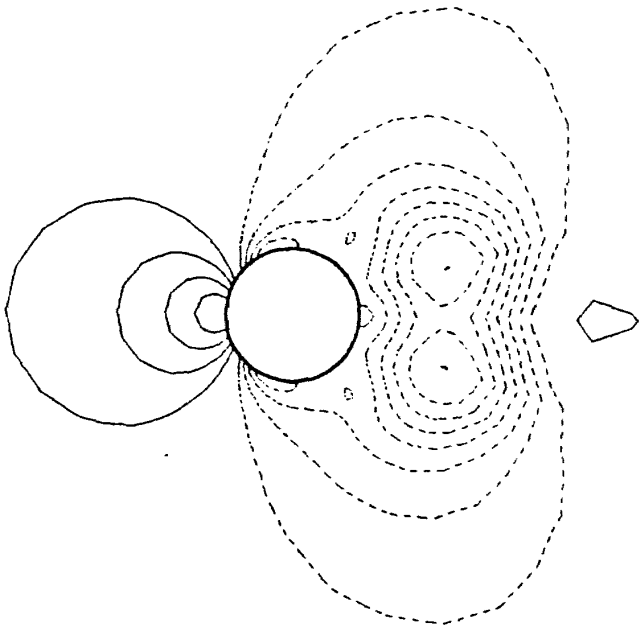


Fig. 6 Pressure distribution around circular cylinder obtained by the SBS method ($L_2 < 10^{-4}$, and iteration number = 101 + 107).

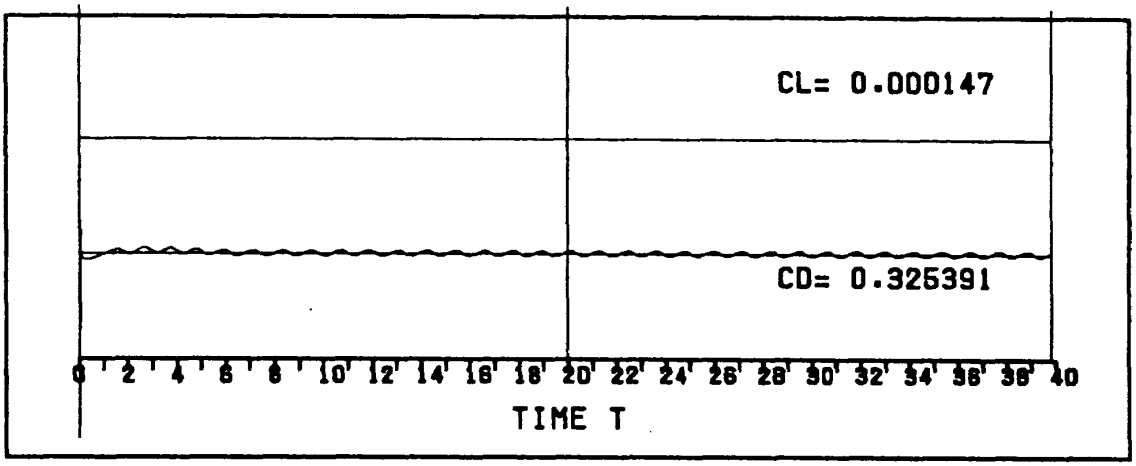
detailed comparison of the three methods: SOR, SBB, SBS. The SBS method is the most efficient which achieves an acceleration rate of about 21 as compared to the SOR method.

The SBS method is then incorporated into the primitive variable procedure and applied to solve the pressure Poisson equation at every time step. A comparison between the SOR method and the SBS method as a pressure Poisson solver in the primitive variable procedure is made by solving the flow around a circular cylinder at the Reynolds number of 10^5. Here the iteration number of the SOR method is fixed to 100, while the L_2 residual of the SBS method is again made smaller than 10^{-4}. Figure 7 shows the time variaton of the aerodynamic coefficients (Cl, Cd) together with time averaged values for each case. The SOR Poisson solver fails to capture the time variation of Cl and results in a time averaged Cd (≈ 0.32) much deviated from the experimental value (≈ 1.22). On the other hand, the SBS Poisson solver captures the time variation of Cl and obtains a time averaged Cd (≈ 1.33) close to the measured value. Incidentally, the iteration number needed in the first and the third steps of the SBS method at each time level is about 50, which is due to the good initial value for the Poisson equation produced during the successive time integration process.

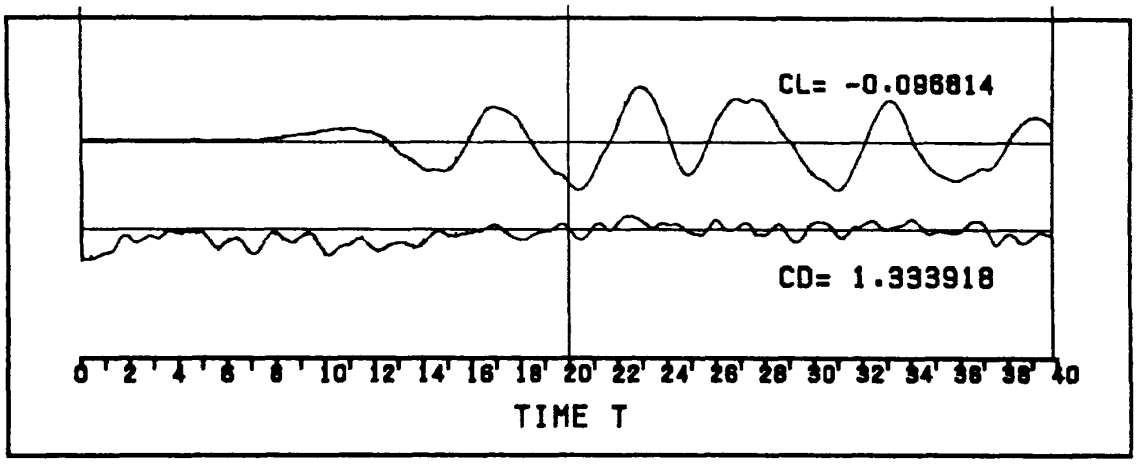
In the following, several applications of the primitive variable procedure are presented. Figure 8 shows the numerical results of the internal flows through complicated multiple passages by using the Multi-Domain Technique, and Figure 9 the flow patterns and the temperature fields of natural convection in a complicated enclosure. It should be noted that the pressure boundary conditions enforced here must not be non-gradient of pressure at solid boundaries, but should be obtained from the balance between the pressure gradient term and the buoyancy force term of the normal direction momentum equations. Figure 10(a) shows the overlapped grid system used to calculate the flow around an airfoil (NACA4412) with a slat and flap. The resulting particle pathes are plotted in Fig. 10(b) and the time-averaged surface pressure distributions in Fig. 10(c).

5. CONCLUDING REMARKS

Incompressible flow solvers based on primitive variables as well as the $\psi - \omega$ formulation are pre-



(a)



(b)

Fig. 7 Time variation of aerodynamic coefficients (*Cl*, *Cd*) around circular cylinder together with time averaged values: $Re = 10^5$ and the primitive variable method. The pressure Poisson solver is (a) the SOR method and (b) the SBS method.

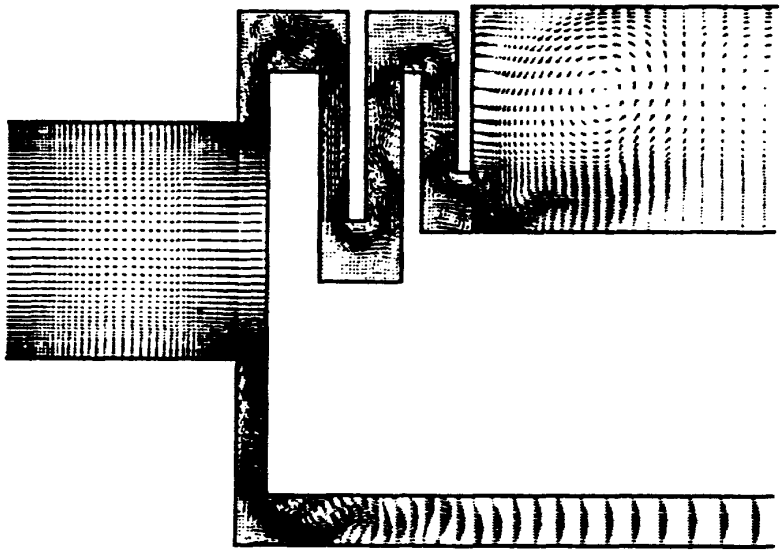
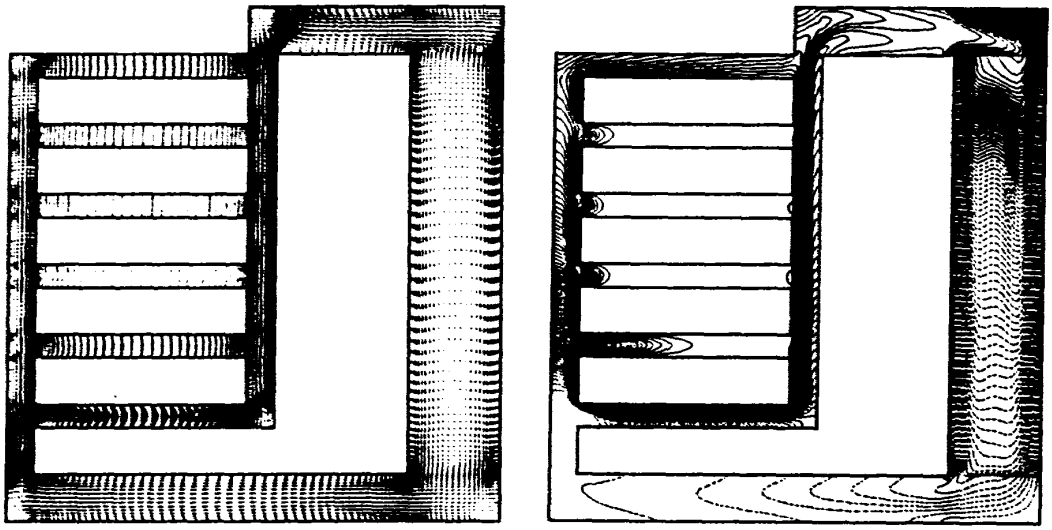
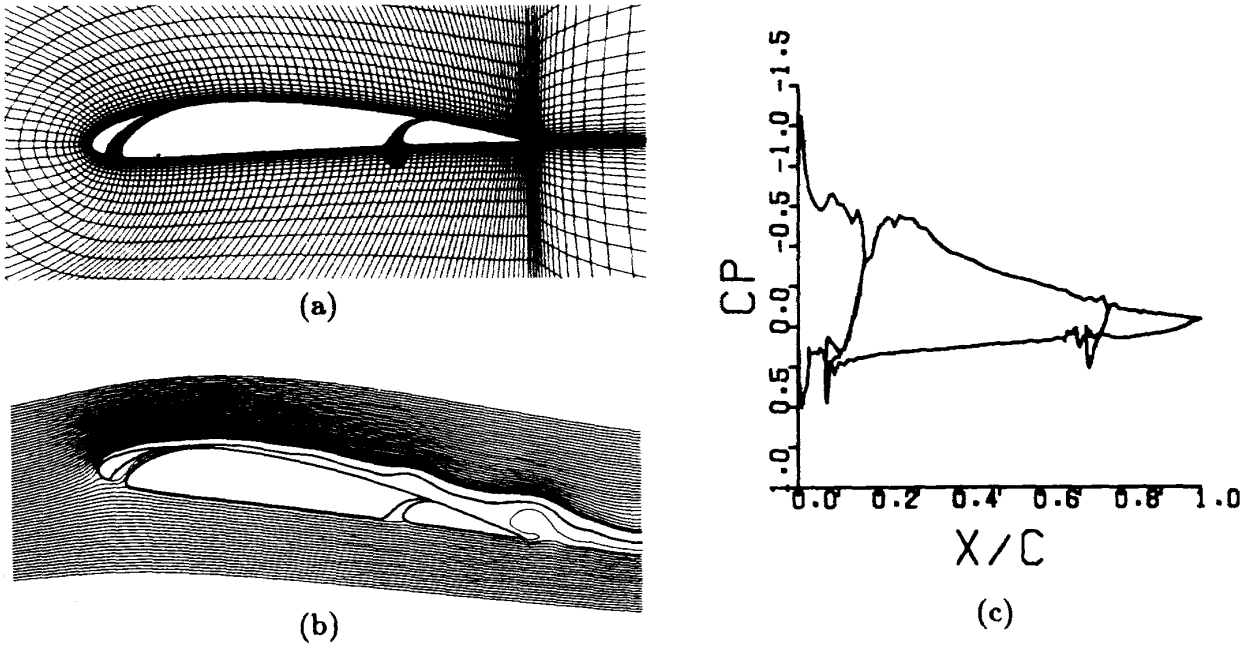


Fig. 8 Velocity vectors of internal flow through multiple passages.



(a) (b)

Fig. 9 Velocity vectors and temperature distributions of natural convection in complicated enclosure, where five blocks have higher temperatures relative to the walls of the right passage. The Prandtl number is 0.84 and the Grashof number is 10^6.



(a) (b) (c)

Fig. 10 Calculation of the flow around NACA 4412 airfoil with slat and flap. $Re = 1.6 \times 10^5$, the attach angle is 10° and the flap angle is 0. (a) Overlapped grid system, (b) instantaneous particle pathes, and (c) the time-averaged surface pressure distribution.

sented. The explanation, however, is mainly made on the former which has a general use. In the procedure, the coupled form of the momentum and energy equations written in generalized coordinates are solved by the FDM. To construct the FDM code, a two-step time integration method with second order accuracy, the third order approximation of convective terms by the QUICK method, and an accurate, fast convergence pressure Poisson solver developed by us are employed. The calculation of practical flow problems with complicated geometry is made easy by incorporating an efficient multi-domain technique into the procedure.

Fundamental flow problems on a flat plate with a leading edge and around a circular cylinder are calculated by the present procedure. The results verified the accuracy and efficiency of the procedure and showed superiority over other methods as the

Reynolds number becomes high. The procedure can easily be used to solve practical flow problems with complicated geometry, and several examples were presented for demonstration.

References

1. B.P. Leonard: A Stable and Accurate Convective Modelling Procedure Based on Quadratic Upstream Interpolation, Computer Methods in Applied Mechanics and Engineering, 19 (1979), pp. 58-98.
2. Y. Nakamura and Y. Takemoto: Solutions of Incompressible Flows Using a Generalized QUICK Method, Proceedings of ISCFD-Tokyo, 2 (1985), edited by K. Oshima, pp. 285-296.
3. W. Jia, Y. Nakamura, M. Yasuhara and K. Kuwabara: Calculation of Flow over Airfoil with Slat and Flap, Journal of the Japan Society for Aeronautical and Space Sciences, 37 (1989), pp. 430-440, in Japanese.
4. W. Jia, Y. Nakamura and M. Yasuhara: Natural Convection Flow Solver in Primitive Variable Form, Journal of the Japan Society of Fluid Mechanics, 9 (1990), pp. 34-52, in Japanese.
5. Y. Nakamura, W. Jia and M. Yasuhara: A New Method to Solve Poisson Equation with Neumann Bounday Conditions, Journal of the Japan Society for Aeronautical and Space Sciences, 38 (1990) (to appear), in Japanese.
6. Y. Nakamura, W. Jia and M. Yasuhara: Incompressible Flow through Multiple Passages, Numerical Heat Transfer, Part A, 16 (1989), pp. 451-465.
7. S. Ostrach: Natural Convection in Enclosures, ASME J. Heat Transfer, 110 (1987), pp. 1075-1190.
8. W. Jia, Y. Nakamura and M. Yasuhara: Some Improvements of an Incompressible Flow Solver in $\psi-\omega$ Variables, Journal of the Japan Society for Aeronautical and Space Sciences, 38 (1990), pp. 309-320, in Japanese.
9. P.J. Roache: Computational Fluid Dynamics, Hermosa, Albuquerque, N.M., 1976.
10. W.R. Briley: Numerical Method for Predicting Three-Dimensional Steady Viscous Flows in Ducts, J. Comput. Phys., 14 (1974), pp. 8-28.