

AN ADAPTIVE MULTIGRID FINITE-VOLUME SCHEME FOR INCOMPRESSIBLE NAVIER–STOKES EQUATIONS

SAN-YIH LIN* AND TSUEN-MUH WU†

Institute of Aeronautics and Astronautics, National Cheng Kung University, Tainan, Taiwan 70101, R.O.C.

SUMMARY

An algorithm for the solutions of the two-dimensional incompressible Navier–Stokes equations is presented. The algorithm can be used to compute both steady-state and time-dependent flow problems. It is based on an artificial compressibility method and uses higher-order upwind finite-volume techniques for the convective terms and a second-order finite-volume technique for the viscous terms. Three upwind schemes for discretizing convective terms are proposed here. An interesting result is that the solutions computed by one of them is not sensitive to the value of the artificial compressibility parameter. A second-order, two-step Runge–Kutta integration coupling with an implicit residual smoothing and with a multigrid method is used for achieving fast convergence for both steady- and unsteady-state problems. The numerical results agree well with experimental and other numerical data. A comparison with an analytically exact solution is performed to verify the space and time accuracy of the algorithm.

KEY WORDS Finite-volume method Multigrid method Incompressible Navier–Stokes equation

1. INTRODUCTION

Numerical solutions to the incompressible Navier–Stokes equations are in greater demand than ever before as the field of computational fluid dynamics becomes more important as an engineering tool. An efficient code is the key to developing a useful tool for flow analysis. Therefore, there is a continuing interest in finding solution methodologies which will produce results using the least amount of computing time and CPU memories. This is particularly true for problems with a high Reynolds number.

The algorithm is based on an artificial compressibility approach, which has been used successfully by a number of other authors.^{1–3} The advantages of using artificial compressibility are that it directly couples the pressure and velocity fields at the same time level, and produces a hyperbolic-dominated system of the equations. Since the equations are hyperbolic-dominated, some of the upwind finite-volume schemes which have recently been developed for the compressible Euler and Navier–Stokes equations by a number of authors^{4,5} can be utilized. From a theoretical point of view, the amount of artificial compressibility parameter β seems to be arbitrary and can be chosen at will in the differential equation level, or in the difference level when central difference is used. But in upwind schemes, the upwind difference is directly related to the eigensystem of the problem and is strongly affected by the choice of β . In this paper, we investigate three different upwind schemes to demonstrate the effect of the choice of β . The key result from

* Associate Professor

† Graduate student

this work is that the numerical results computed by one of these schemes are almost independent of the value of β . For the time integration, we use an explicit two-step Runge–Kutta time integration with an implicit residual smoothing. To improve the convergence speed of the scheme, we include a multigrid method in the numerical procedure. Trial results showed that our scheme is accurate and efficient and can compete with other implicit schemes coupling with multigrid techniques. An adaptive grid generation is included in the numerical method to achieve more accurate results under coarser grid systems.

In the following sections, details of the artificial compressibility method and its use in solving the incompressible Navier–Stokes equation for both steady- and unsteady-state problems are given. Three upwind finite-volume formulations are given and compared with each other by a standard test problem—cavity flow. Details of the boundary condition procedures are also included. The numerical examples include a driven cavity flow which serves as a standard test case, flow through a backward-facing step, inviscid flow past a circular cylinder and viscous flow past a circular cylinder. The numerical results showed good comparisons with experimental and other numerical data. Finally, a comparison with an analytically exact result is then performed to verify the space and time accuracy of the algorithm.

2. NUMERICAL FORMULATIONS

2.1. Governing equations and artificial compressibility

The conservative, integral form of the steady incompressible Navier–Stokes equations with artificial compressibility is

$$\frac{\partial}{\partial t} \iint_{\Omega} Q \, d\Omega + \oint_{\Gamma} (F, G) \cdot \mathbf{n} \, dl = \frac{1}{Re_{\infty}} \oint_{\Gamma} (\nabla\phi) \cdot \mathbf{n} \, dl, \tag{1}$$

where

$$Q = \begin{pmatrix} p \\ u \\ v \end{pmatrix}, \quad F = \begin{pmatrix} \beta u \\ u^2 + p \\ uv \end{pmatrix}, \quad G = \begin{pmatrix} \beta v \\ uv \\ v^2 + p \end{pmatrix}, \quad \phi = \begin{pmatrix} 0 \\ u \\ v \end{pmatrix},$$

in which Ω is the domain, Γ is the boundary of Ω , \mathbf{n} is the outer unit normal and dl is the arc length along the control surface. Q represents the vector of conservation variables. The second term on the left-hand side of equation (1) is the inviscid flux vector and the right-hand side of equation (1) is the viscous flux vector. The p and (u, v) are the pressure and the Cartesian velocity, β ($\beta > 0$) is the artificial compressibility parameter and Re_{∞} is the Reynolds number corresponding to the free-stream velocity.

Equation (1) has been non-dimensionalized with

$$x = \frac{\bar{x}}{\bar{L}}, \quad y = \frac{\bar{y}}{\bar{L}}, \quad t = \frac{\bar{t}}{\bar{L}/\bar{U}_{ref}}, \quad p = \frac{\bar{p} - \bar{p}_{ref}}{\bar{\rho}_{ref} \bar{U}_{ref}^2}, \quad Re_{\infty} = \frac{\bar{U}_{ref} \bar{L}}{\bar{\nu}},$$

where ν is the viscosity parameter. The subscript ‘ref’ denotes the reference condition, the overbar denotes the physical variable with dimensions and L denotes the reference characteristic length.

2.2. Space discretization: finite-volume formulation

We assume that the two-dimensional domain Ω can be discretized into a group of quadrilaterals $\{K_{i,j}\}$. The vertices, barycentres and edges of the quadrilateral $K_{i,j}$ are denoted by $V_{i\pm 1/2, j\pm 1/2}$, $C_{i,j}$ and $e_{i\pm 1/2, j}$, $e_{i, j\pm 1/2}$, respectively. In each quadrangle $K_{i,j}$ flow variables are

stored at the barycentre $C_{i,j}$ and flow conservation is enforced on the boundary $\partial K_{i,j}$. If we assume that the grid mesh is geometrically time-invariant and the flow variables stored at the barycentre $C_{i,j}$ are an area average of the integrated flow variables in the quadrilateral $K_{i,j}$, then equation (1) can be written as

$$\frac{dQ_{i,j}}{dt} = -\frac{1}{A_{i,j}} \oint_{\Gamma_{i,j}} (F, G) \cdot \mathbf{n} \, dl + \frac{1}{Re_\infty} \oint_{\Gamma_{i,j}} (\nabla\phi) \cdot \mathbf{n} \, dl, \tag{2}$$

where $A_{i,j}$ is the area of $K_{i,j}$ and $\Gamma_{i,j} = \partial K_{i,j}$.

To evaluate the first term on the right-hand side of (2), we sum all the flux vectors on the four edges of $K_{i,j}$:

$$\oint_{\Gamma_{i,j}} (F, G) \cdot \mathbf{n} \, dl \sim h_{i-1/2,j} |e_{i-1/2,j}| + h_{i+1/2,j} |e_{i+1/2,j}| + h_{i,j-1/2} |e_{i,j-1/2}| + h_{i,j+1/2} |e_{i,j+1/2}|, \tag{3}$$

where $h_{i+1/2,j}$ is the numerical approximation for the flux associated with the edge $e_{i+1/2,j}$ and $|e_{i+1/2,j}|$ is the length of the edge $e_{i+1/2,j}$.

In order to evaluate $h_{i+1/2,j}$ using an upwind scheme, it is necessary to have two fluid dynamic states, $Q_{i+1/2,j}^L$ and $Q_{i+1/2,j}^R$. The states $Q_{i+1/2,j}^L$ and $Q_{i+1/2,j}^R$ are interpolated from the cell-centred states by means of the MUSCL (Monotonic Upstream Schemes for Conservation Laws) type schemes.^{4,5} They can be expressed as

$$Q_{i+1/2,j}^R = Q_{i+1,j} - a_{i+1,j} \left[(c_{i+1,j} - 2\kappa b_{i+1,j})(Q_{i+2,j} - Q_{i+1,j}) + \frac{1}{c_{i+1,j}} (1 + 2\kappa b_{i+1,j})(Q_{i+1,j} - Q_{i,j}) \right],$$

and

$$Q_{i+1/2,j}^L = Q_{i,j} + a_{i,j} \left[(c_{i,j} - 2\kappa b_{i,j})(Q_{i,j} - Q_{i-1,j}) + \frac{1}{c_{i,j}} (1 + 2\kappa b_{i,j})(Q_{i+1,j} - Q_{i,j}) \right], \tag{4}$$

where the non-uniformity of cell sizes is taken into account in $a_{i,j}$, $b_{i,j}$, $c_{i,j}$.⁶ Let $l_{i,j}$ represent the width of cell $K_{i,j}$ in the i direction; then

$$\begin{aligned} a_{i,j} &= \frac{l_{i,j}}{l_{i+1,j} + 2l_{i,j} + l_{i-1,j}}, \\ b_{i,j} &= \frac{l_{i,j}}{l_{i,j} + l_{i+1,j}}, \\ c_{i,j} &= \frac{l_{i,j} + l_{i-1,j}}{l_{i,j} + l_{i+1,j}}. \end{aligned} \tag{5}$$

For a uniform mesh, they become

$$a_{i,j} = \frac{1}{4}, \quad b_{i,j} = \frac{1}{2}, \quad c_{i,j} = 1;$$

then (4) becomes

$$Q_{i+1/2,j}^R = Q_{i+1,j} - \left[\frac{1-\kappa}{4} (Q_{i+2,j} - Q_{i+1,j}) + \frac{1+\kappa}{4} (Q_{i+1,j} - Q_{i,j}) \right]$$

and

$$Q_{i+1/2,j}^L = Q_{i,j} + \left[\frac{1+\kappa}{4} (Q_{i+1,j} - Q_{i,j}) + \frac{1-\kappa}{4} (Q_{i,j} - Q_{i-1,j}) \right]. \tag{6}$$

The value of κ determines the type of difference scheme: $\kappa = -1$ yields a one-sided upwind scheme; $\kappa = \frac{1}{3}$ yields a third-order upwind-biased scheme; $\kappa = 1$ yields a central difference scheme. For the calculations in this paper, a value of $\kappa = \frac{1}{3}$ was used.

Then the numerical flux at an interface is written as

$$h_{i+1/2,j} = h(Q_{i+1/2,j}^L, Q_{i+1/2,j}^R) = \frac{1}{2} \{ F(Q_{i+1/2,j}^L) n_x + G(Q_{i+1/2,j}^L) n_y + F(Q_{i+1/2,j}^R) n_x + G(Q_{i+1/2,j}^R) n_y - \alpha_{i+1/2,j} (Q_{i+1/2,j}^R - Q_{i+1/2,j}^L) \}. \tag{7}$$

Here, we used two local Lax–Friedrichs monotone flux methods and the Roe’s approximate Riemann solver to define the value $\alpha_{i+1/2,j}$:

scheme 1: local Lax–Friedrichs monotone flux I

$$\alpha_{i+1/2,j} = \max \{ |U_n^L| \pm \sqrt{[(U_n^L)^2 + \beta]}, |U_n^R| \pm \sqrt{[(U_n^R)^2 + \beta]} \}. \tag{8}$$

scheme 2: Roe’s approximate Riemann solver

$$\alpha_{i+1/2,j} = \frac{\Delta F_{i+1/2,j}}{\Delta Q_{i+1/2,j}} = A(Q_{i+1/2,j}^L, Q_{i+1/2,j}^R) = LAR, \tag{9}$$

where L and R are the left and right eigenmetrics of the Jacobian matrix A at the Roe averaged values. For details of L , R and A , please see Reference 2.

scheme 3: local Lax–Friedrichs monotone flux II

$$\alpha_{i+1/2,j} = \max (|U_n^L|, |U_n^R|, \varepsilon), \quad \varepsilon = 0.01 - 0.001, \tag{10}$$

where $U_n = u_{i+1/2,j} n_x + v_{i+1/2,j} n_y$ is the contravariant velocity normal to the edge $e_{i+1/2}$. The value α is selected as the minimum one to give a scheme where the Lax–Friedrichs monotone flux is stable and independent of the value of β . The numerical results also show that this scheme is better than schemes 1 and 2 (see Section 3.1).

We also use the finite-volume discretizations to compute the viscous terms. Since the viscous flux components are functions of the velocity gradients, we have to estimate appropriate values of these gradients on the cell faces. A general procedure, valid for an arbitrary control in two and three dimensions, can be derived by application of the divergence theorem, but for a non-uniform grid system, one needs to be very careful about truncation errors. One can see References 7 and 8 for the derivations of the viscous terms and the relative error analysis. Here, for completeness, we give the formulation of the viscous terms in a quadrilateral K_{ij} (see Figure 1) as

$$\oint_{\Gamma} (\nabla \phi) \cdot \mathbf{n} \, dl \sim \{ (\phi_E - \phi_P) [(\Delta y_{12})^2 + (\Delta x_{12})^2] / J_e + (\phi_2 - \phi_1) [-\Delta x_{PE} \Delta x_{12} - \Delta y_{PE} \Delta y_{12}] / J_e + (\phi_W - \phi_P) [(\Delta y_{43})^2 + (\Delta x_{34})^2] / J_w + (\phi_4 - \phi_3) [-\Delta x_{PW} \Delta x_{34} - \Delta y_{PW} \Delta y_{34}] / J_w + (\phi_N - \phi_P) [(\Delta y_{23})^2 + (\Delta x_{23})^2] / J_n + (\phi_3 - \phi_2) [-\Delta x_{PN} \Delta x_{23} - \Delta y_{PN} \Delta y_{23}] / J_n + (\phi_S - \phi_P) [(\Delta y_{14})^2 + (\Delta x_{14})^2] / J_s + (\phi_1 - \phi_4) [-\Delta x_{PS} \Delta x_{41} - \Delta y_{PS} \Delta y_{41}] / J_s \}, \tag{11}$$

where

$$J_e = \Delta y_{12} \Delta x_{PE} - \Delta y_{PE} \Delta x_{12}, \quad J_w = \Delta y_{43} \Delta x_{WP} - \Delta y_{WP} \Delta x_{43}, \\ J_n = \Delta y_{PN} \Delta x_{32} - \Delta y_{32} \Delta x_{PN}, \quad J_s = \Delta y_{SP} \Delta x_{41} - \Delta y_{41} \Delta x_{SP}. \tag{12}$$

2.3. Time integrations: Runge–Kutta algorithms and residual smoothing

2.3.1. Steady-state formulation. An explicit multistage Runge–Kutta algorithm is used to discretize the remaining time derivative in equation (2).

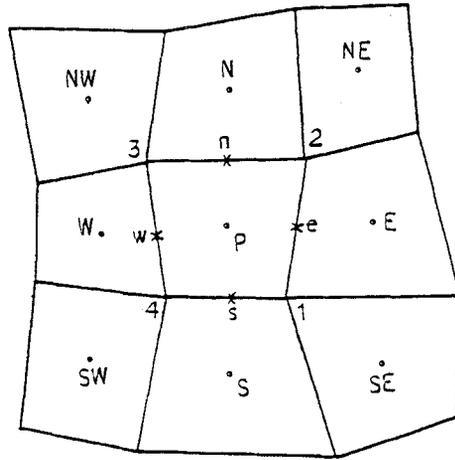


Figure 1. The quadrilateral control volumes on physical domain

Define a residual as

$$\text{Res}_{ij} = -\frac{1}{A_{ij}} \oint_{\Gamma_{ij}} (F, G) \cdot \mathbf{n} dl + \frac{1}{Re_\infty} \oint_{\Gamma_{i,j}} (\nabla\phi) \cdot \mathbf{n} dl; \tag{13}$$

then, equation (2) becomes

$$\frac{dQ_{ij}}{dt} = \text{Res}_{ij}. \tag{14}$$

In this paper, we used a second-order TVD Runge–Kutta time integration^{9, 10} to integrate the ODE (14). The two-step Runge–Kutta time integration is as follows:

$$\begin{aligned} Q_{i,j}^{(0)} &= Q_{i,j}^n, \\ Q_{i,j}^{(1)} &= Q_{i,j}^{(0)} + \Delta t_{i,j} \text{Res}_{i,j}(Q_{i,j}^{(0)}), \\ Q_{i,j}^{(2)} &= Q_{i,j}^{(1)} + \Delta t_{i,j} \text{Res}_{i,j}(Q_{i,j}^{(1)}), \\ Q_{i,j}^{n+1} &= 0.5(Q_{i,j}^{(0)} + Q_{i,j}^{(2)}). \end{aligned} \tag{15}$$

We define

$$\text{CFL}_{i,j} = \frac{\Delta t_{i,j} C_{\max}}{|e_{i,j}|_{\min}}, \tag{16}$$

where $|e_{i,j}|_{\min} = \min(|e_{i+1/2,j}|, |e_{i-1/2,j}|, |e_{i,j+1/2}|, |e_{i,j-1/2}|)$, and

$$\begin{aligned} C_{i+1/2,j} &= \max\{\sqrt{[(U_n^L)^2 + \beta]}, \sqrt{[(U_n^R)^2 + \beta]}\}, \\ U_n &= u_{i+1/2,j} n_x + v_{i+1/2,j} n_y, \\ C_{\max} &= \max(C_{i+1/2,j}, C_{i-1/2,j}, C_{i,j+1/2}, C_{i,j-1/2}). \end{aligned} \tag{17}$$

At this moment, the whole scheme is stable for Courant number $\text{CFL}_{i,j} \leq 1$. For this reason, we apply the implicit residual smoothing method in our scheme to reduce the CFL number constraint.

2.3.2. *Time-accurate formulation.* We apply the Crank–Nicolson method to the original incompressible Navier–Stokes equation, and then use pseudo-time τ and the artificial compressibility method to solve the time-accurate problems. Now we assume that Q^n at the n th time step was known. As in Section 2.3.1, we define a residual as

$$\text{Res}_{ij}^{n+1} = Q_{i,j}^{n+1} - Q_{i,j}^n - \frac{\Delta t}{2} [\text{Res}_{i,j}(Q^{n+1}) + I_d \text{Res}_{i,j}(Q^n)], \tag{18}$$

with

$$I_d = \text{diag}(0, 1, 1).$$

Then use the two-step Runge–Kutta time integration (15) to solve the following ordinary differential equation to reach a steady-state solution which is the $(n + 1)$ th time step solution, Q^{n+1} :

$$\frac{dQ_{ij}^{n+1}}{d\tau} = \text{Res}_{ij}^{n+1}. \tag{19}$$

2.3.3. *Implicit residual smoothing.* For simplicity, we use $\text{Res}_{i,j}$ to represent both $\text{Res}_{i,j}$ (for steady-state problems) and $\text{Res}_{i,j}^{n+1}$ (for unsteady problems). The implicit residual smoothing step replaces $\text{Res}_{i,j}$ by $\text{Res}_{i,j}^{\text{ew}}$ as follows:

$$(1 - \varepsilon_y \delta_y^2)(1 - \varepsilon_x \delta_x^2) \Delta t_{i,j} \text{Res}_{i,j}^{\text{ew}}(Q_{i,j}) = \Delta t_{i,j} \text{Res}_{i,j}(Q_{i,j}), \tag{20}$$

where δ_x^2 and δ_y^2 are the symmetrical, central operators which approximate the second derivatives in the i and j directions and $\varepsilon_x, \varepsilon_y$ are the corresponding smooth coefficients.

Linear stability analysis has shown that the Runge–Kutta scheme with implicit residual smoothing may be made unconditionally stable provided that ε is sufficiently large.^{9, 11} For example, in one dimension

$$\varepsilon \geq \frac{1}{4} \left[\left(\frac{\text{CFL}}{\text{CFL}^*} \right)^2 - 1 \right] \tag{21}$$

gives a scheme that is unconditionally stable, where CFL^* is the stability limit of the CFL number of the TVD Runge–Kutta scheme without smoothing. In two dimensions different values of ε are often used in each grid direction. If a constant value of ε is chosen based on the largest value of CFL/CFL^* , the scheme can remain stable for large CFL. For example, if we take ε equal to 0.5, the CFL number can be chosen to be $\sqrt{3}$. In this paper, we take $\text{CFL} = 0.8$ without residual smoothing and $\text{CFL} = 1.6$ with residual smoothing ($\varepsilon = 0.5$).

2.4. Multigrid methods

To improve the convergence speed of the scheme, we introduce a multigrid method to the explicit two-step Runge–Kutta scheme with/without implicit residual smoothing. As has been noted by Holl¹² and Jameson,⁹ multigrid time stepping methods also expand the domain of dependence of the discrete scheme in a way that corresponds to signal propagation in a physical system. This allows a large effective time step to be attained by a multigrid cycle without the need of introducing an implicit time stepping scheme.

The operation of transfer of function from fine to coarse grids is called *restriction*, and that from coarse to fine grids is called *prolongation*. The restriction operator $R_{k, k-1}$ transfers a fine-grid function Q_k to a coarse-grid function Q_{k-1} and the prolongation operator $P_{k-1, k}$ transfers a coarse-grid function Q_{k-1} to a fine-grid function Q_k . The definition of $R_{k, k-1}$ is (see Figure 2)

$$Q_{k-1}^{(0)} = R_{k, k-1} Q_{k-1} = \frac{A_{8120} Q_{k, a} + A_{2340} Q_{k, b} + A_{4560} Q_{k, c} + A_{6780} Q_{k, d}}{A_{1357}}. \tag{22}$$

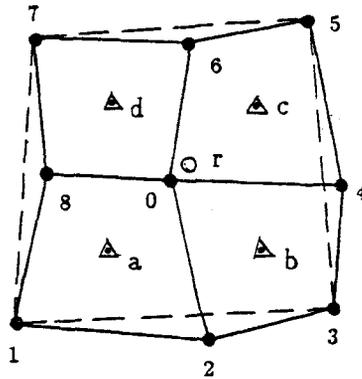


Figure 2. Restriction with area-weighted averages

It is important that the solution on the coarser grid be driven by the residual computed on the fine grid, so a residual forcing function is defined as

$$S_{k-1} = \sum \text{Res}_k - \text{Res}_{k-1}(Q_{k-1}^{(0)}),$$

where Res_k is the residual on the k th grid level and

$$\sum \text{Res}_k = A_{8120} \text{Res}_{k,a} + A_{2340} \text{Res}_{k,b} + A_{4560} \text{Res}_{k,c} + A_{6780} \text{Res}_{k,d}.$$

The residual used to drive the corrections on the coarser grid is then given by

$$\overline{\text{Res}}_{k-1}^{(m)} = \text{Res}_{k-1}(Q_{k-1}^{(m)}) + S_{k-1}.$$

Then the two-step TVD Runge-Kutta time integration scheme on one of the coarser grids must be reformulated as

$$\begin{aligned} Q_{k-1}^{(0)} &= R_{k,k-1} Q_k^{(n)}, \\ Q_{k-1}^{(1)} &= Q_{k-1}^{(0)} + \Delta t_{i,j} \overline{\text{Res}}_{k-1}^{(0)}, \\ Q_{k-1}^{(2)} &= Q_{k-1}^{(0)} + \Delta t_{i,j} \overline{\text{Res}}_{k-1}^{(1)}, \\ Q_{k-1}^{n+1} &= 0.5(Q_{k-1}^{(0)} + Q_{k-1}^{(2)}). \end{aligned} \tag{23}$$

After corrections have been computed on the $(k-1)$ th coarser grid, the process is continued to the $(k-2)$ th coarser grids in a similar manner.

After corrections have been computed on the coarser grid, they are prolonged back to successively finer grids. We used bilinear interpolation to define the operator $P_{k-2,k-1}$. Let Q_{k-1}^+ be the final value of Q_{k-1} resulting from both the correction calculated in the time step on the $(k-1)$ th grid and the correction transformed from the $(k-2)$ th grid; then

$$Q_{k-1}^+ = Q_{k-1}^{n+1} + P_{k-2,k-1}(Q_{k-2}^+ - Q_{k-2}^{(0)}). \tag{24}$$

For the operator $P_{k-2,k-1}$, we transform the conservative variable on the physical domain into the generalized co-ordinates (ξ, η) ; then we use bilinear interpolation in the computational co-ordinates.

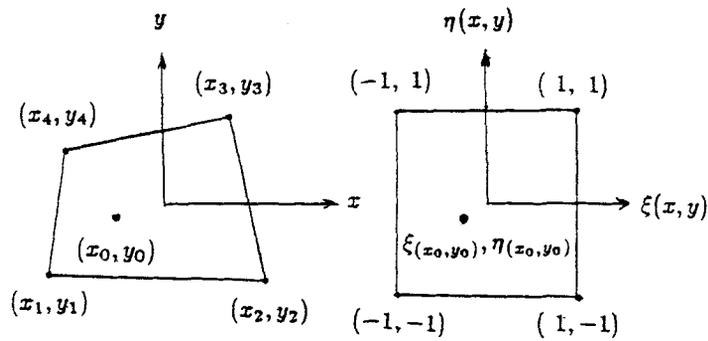


Figure 3. Bilinear interpolation of (x_0, y_0) from the four surrounding points

Bilinear interpolation of (x_0, y_0) from the four surrounding points (see Figure 3) gives

$$\begin{aligned}\xi(x, y) &= a_1 + a_2x + a_3y + a_4xy, \\ \eta(x, y) &= b_1 + b_2x + b_3y + b_4xy.\end{aligned}$$

In the ξ direction,

$$\begin{aligned}\xi(x_1, y_1) &= a_1 + a_2x_1 + a_3y_1 + a_4x_1y_1 = -1, \\ \xi(x_2, y_2) &= a_1 + a_2x_2 + a_3y_2 + a_4x_2y_2 = 1, \\ \xi(x_3, y_3) &= a_1 + a_2x_3 + a_3y_3 + a_4x_3y_3 = 1, \\ \xi(x_4, y_4) &= a_1 + a_2x_4 + a_3y_4 + a_4x_4y_4 = -1.\end{aligned}$$

In the η direction,

$$\begin{aligned}\eta(x_1, y_1) &= b_1 + b_2x_1 + b_3y_1 + b_4x_1y_1 = -1, \\ \eta(x_2, y_2) &= b_1 + b_2x_2 + b_3y_2 + b_4x_2y_2 = -1, \\ \eta(x_3, y_3) &= b_1 + b_2x_3 + b_3y_3 + b_4x_3y_3 = 1, \\ \eta(x_4, y_4) &= b_1 + b_2x_4 + b_3y_4 + b_4x_4y_4 = 1.\end{aligned}$$

Then we define

$$\begin{aligned}P_{k-2, k-1} Q(x_0, y_0) &= 0.25 \{ [1 - \xi(x_0, y_0)] [1 - \eta(x_0, y_0)] Q_{k-2}(x_1, y_1) \\ &\quad + [1 + \xi(x_0, y_0)] [1 - \eta(x_0, y_0)] Q_{k-2}(x_2, y_2) \\ &\quad + [1 + \xi(x_0, y_0)] [1 + \eta(x_0, y_0)] Q_{k-2}(x_3, y_3) \\ &\quad + [1 - \xi(x_0, y_0)] [1 + \eta(x_0, y_0)] Q_{k-2}(x_4, y_4) \}. \quad (25)\end{aligned}$$

In this paper, we used a fixed V -cycle (see Figure 4) in which one time step of the Runge-Kutta time integration is performed on each grid level up to the coarsest grid, and then the results are prolonged back to the finest grid.

2.5. Adaptive grid generation

In this section, an adaptive grid method is used coupling the line-by-line equidistribution adaptive grid method of Dwyer and co-workers^{13, 14} with Jeng and Liou's averaging proced-

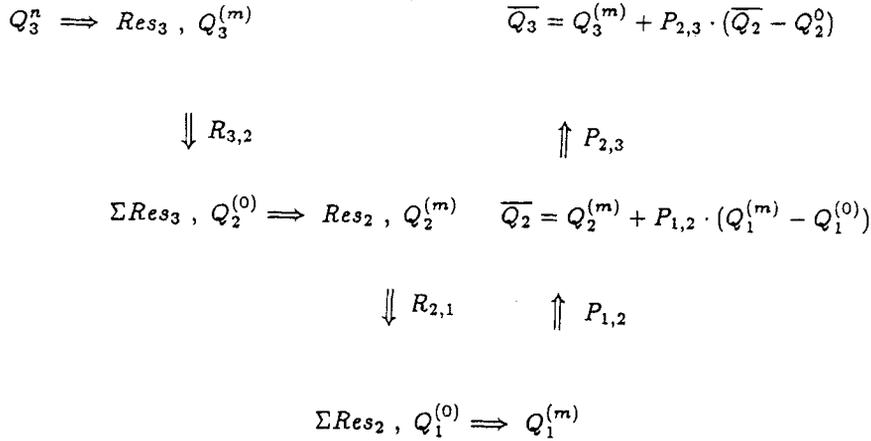


Figure 4. A multigrid procedure with fixed V -cycles of three grid levels

ure.¹⁵ To apply their adaptive grid method, one needs to compute weighting functions to adjust the spacing of the grid system. Let Q be the computed solution on an original grid system (ξ, η) . First, along a $\xi = \xi_i$ grid line in the original grid system, the line-by-line adaptive grid method is applied to decide the weighting function $w_i(s)$,

$$\begin{aligned}
 &w_i(s) s_\eta = \text{constant}, \\
 &w_i(s) = 1 + b \left[\frac{1}{6} \left(\left| \frac{\partial Q}{\partial s} \right|_{i-1} \right) + \left(\left| \frac{\partial Q}{\partial s} \right|_{i+1} \right) + \frac{2}{3} \left(\left| \frac{\partial Q}{\partial s} \right|_i \right) \right], \tag{26}
 \end{aligned}$$

where b is an adjustable parameter of grid adaptation and s and s_η are the arc length and the desired grid spacing along the grid line after adaptation, respectively. Similarly, along $\eta = \eta_j$, we can decide the weighting function $w_j(s)$. For details of the adaptive grid procedure, see References 13–15.

2.6. Boundary conditions

Upto now, the numerical algorithm has been developed, and the next most important aspect of solving a fluid dynamics problem is the proper implementation of boundary conditions. In this paper, we consider several different types of boundary conditions encountered in different physical problems.

2.6.1. Solid surface boundary conditions. The solid surface boundary condition for inviscid flow is the no-penetration condition, i.e. the velocity component normal to the wall boundary is identically zero. The mesh system is shown in Figure 5, where ‘1’ and ‘2’ represent the first and second point centres of the control volume above the wall, and ‘b’ is located at wall surface. Then the no-penetration condition is $V_{nb} = 0$. Let V_t be the tangential component of the velocity in the direction tangential to the wall surface. A second-order extrapolation is used to compute V_t at the boundary as follows:

$$V_{tb} = V_b t = V_1 t + (V_1 t - V_2 t) \frac{l_{1b}}{l_{12}}, \tag{27}$$

while ‘ l_{1b} ’ denotes the distance between points ‘b’ and ‘1’ and ‘ l_{12} ’ denotes the distance between points ‘1’ and ‘2’ as shown in Figure 5.

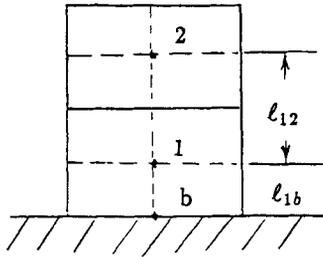


Figure 5. Extrapolation of tangent velocity at wall boundary

For steady inviscid flow, the pressure gradient in the normal direction at the impermeable solid surface can be written as

$$\frac{\partial p}{\partial n} = \frac{V_{1b}^2}{R}, \tag{28}$$

where R is the radius of curvature.

For viscous flow, the usual no-slip condition is applied, i.e. $u = 0, v = 0$. The pressure at the wall is obtained by setting the normal gradient of p equal to zero at the no-slip wall, i.e. $\partial p / \partial n = 0$.

2.6.2. *Inflow and outflow boundary conditions.* Characteristic boundary conditions are used in these boundary conditions.¹ For an inflow boundary, two eigenvalues of the A (see Appendix) are positive; then the inflow boundary condition is computed by two characteristic values from the free stream and one extrapolated from the interior. This can be written in terms of the left eigenvector L as

$$\begin{bmatrix} l_{11} & l_{12} & l_{13} \\ l_{21} & l_{22} & l_{23} \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} p \\ u \\ v \end{bmatrix}_{\text{boundary}} = \begin{bmatrix} (l_{11}p + l_{12}u + l_{13}v)_{\text{freestream}} \\ (l_{21}p + l_{22}u + l_{23}v)_{\text{freestream}} \\ (l_{31}p + l_{32}u + l_{33}v)_{\text{interior}} \end{bmatrix}. \tag{29}$$

For the outflow boundary, only one boundary condition can be imposed, and we have

$$\begin{bmatrix} l_{11} & l_{12} & l_{13} \\ l_{21} & l_{22} & l_{23} \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \begin{bmatrix} p \\ u \\ v \end{bmatrix}_{\text{boundary}} = \begin{bmatrix} (l_{11}p + l_{12}u + l_{13}v)_{\text{far-field}} \\ (l_{21}p + l_{22}u + l_{23}v)_{\text{interior}} \\ (l_{31}p + l_{32}u + l_{33}v)_{\text{interior}} \end{bmatrix}. \tag{30}$$

3. NUMERICAL RESULTS

In this section the results of the following laminar flow computations are presented:

- (1) 2D laminar flow inside a wall-driven cavity,
- (2) 2D laminar flow through a backward-facing step,
- (3) inviscid flow over a cylinder,
- (4) viscous flows with low Reynolds numbers over a cylinder,
- (5) unsteady test case.

In all the cases presented here, the results show that scheme 3 (equation (6)) is better than schemes 1 and 2 in terms of accuracy and stability. It was found that the accuracy and convergence was quite sensitive to the value of β for schemes 1 and 2 but not sensitive for scheme 3. Concerning the

divergence-free flow condition for both steady- and unsteady-state calculations, we use the following convergence criterion on the pressure field:

$$E(n) = \frac{1}{M} \sum |p^{n+1} - p^n| < \varepsilon,$$

where M is the total number of grid points and the summation is taken over the grid points. In our numerical computations, ε is taken as 10^{-5} or 10^{-6} .

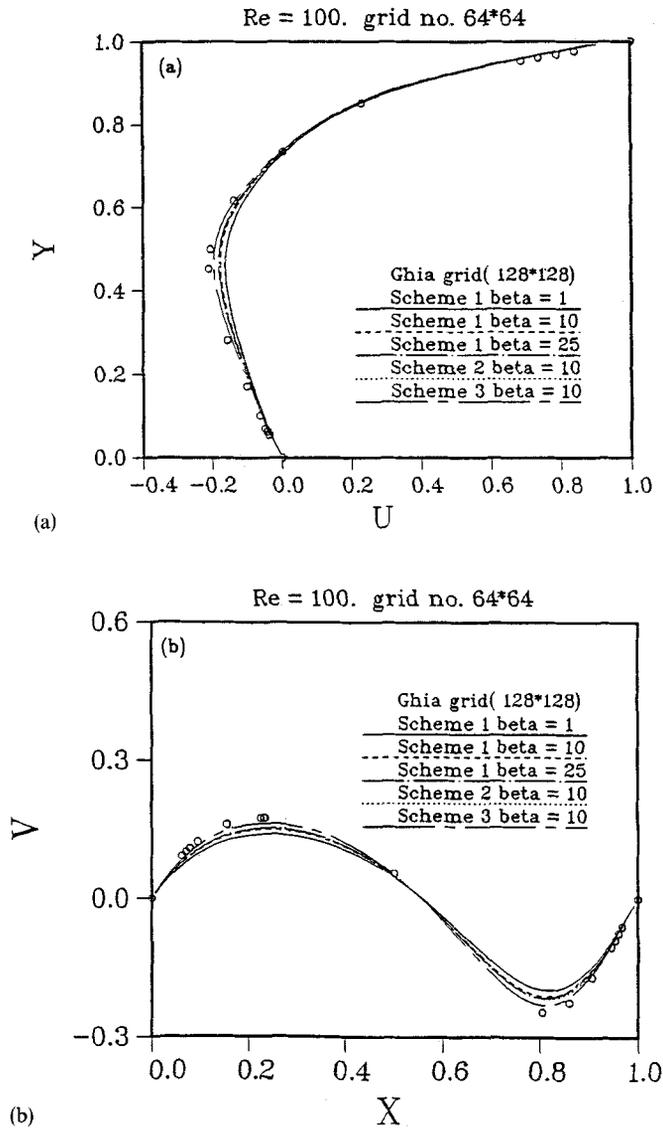
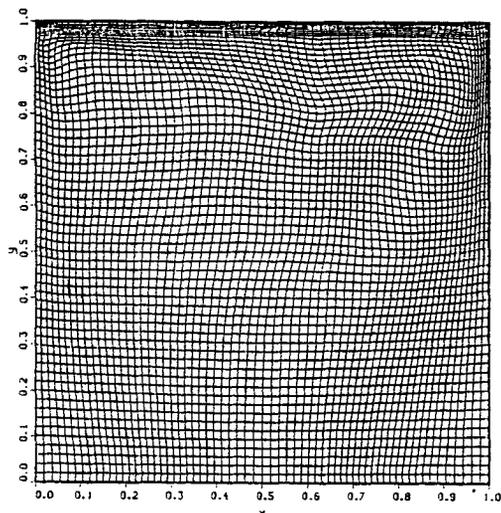


Figure 6. (a) Comparison of the velocity u along a vertical centreline. (b) Comparison of the velocity v along a horizontal centreline

3.1. 2D laminar flow inside a wall-driven cavity

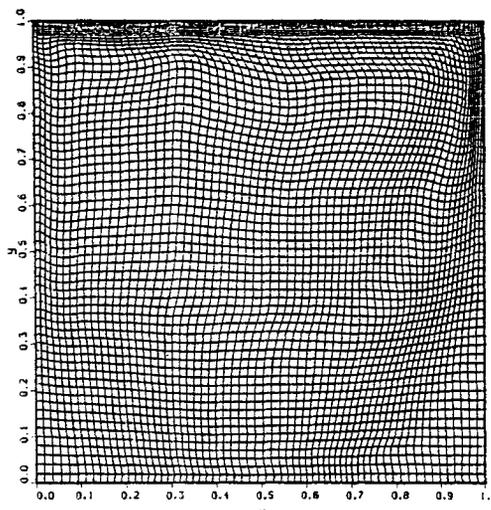
The laminar incompressible flow in a square cavity whose top wall moves with a uniform velocity in its own plane has served over and over again as a model problem for testing and evaluating the numerical scheme. The boundary conditions for (u, v) are $u=v=0$ everywhere except on the top lid, where $u=1, v=0$. We specify the Neumann boundary condition $\partial p/\partial n=0$ at

RE = 100. ADAPTIVE GRID (64*64)



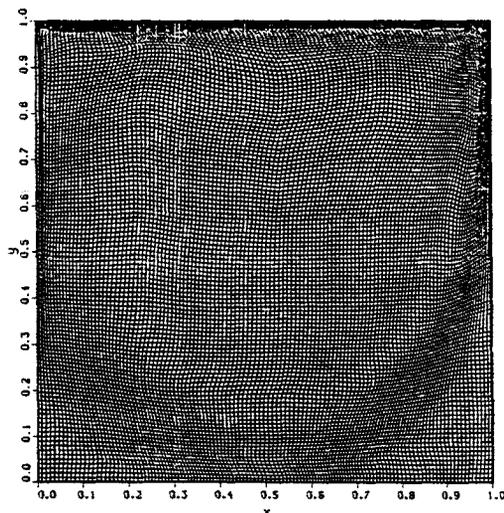
(a)

RE = 400. ADAPTIVE GRID (64*64)



(b)

RE = 1000. ADAPTIVE GRID (128*128)



(c)

Figure 7. Adaptive grids for $Re=100, 400$ and 1000

the wall. The Reynolds numbers considered are 100, 400 and 1000. In each case $u = 1$, $v = 0$ is taken as the initial guess. The computational results of the present study are compared with those of Ghia *et al.*,¹⁶ who used a multigrid finite-difference method for the streamline-vorticity formulations with uniform fine (129×129) grid points for $Re < 3200$ and 257×257 grid points for $Re \geq 5000$.

First, we illustrate the effects of the artificial compressibility coefficient β . Figures 6(a) and 6(b) show respectively, the velocity u along a vertical centreline and the velocity v along a horizontal centreline through the geometric centre of the cavity. The numerical results by scheme 1, with $\beta = 1, 10$ and 25 , show that the performance of this scheme is sensitive to the value of β . The overall results computed by scheme 3 are not sensitive to the value of β . We conclude that scheme 3 is the most accurate of the three schemes.

Second, we consider the effects of the proposed adaptive grid method. Figures 7(a)–7(c) show the adaptive grids for different Reynolds numbers. They vary smoothly and stretch well with the structures of flows. For $Re = 400$, Figures 8(a)–8(d) indicate that the numerical results with the

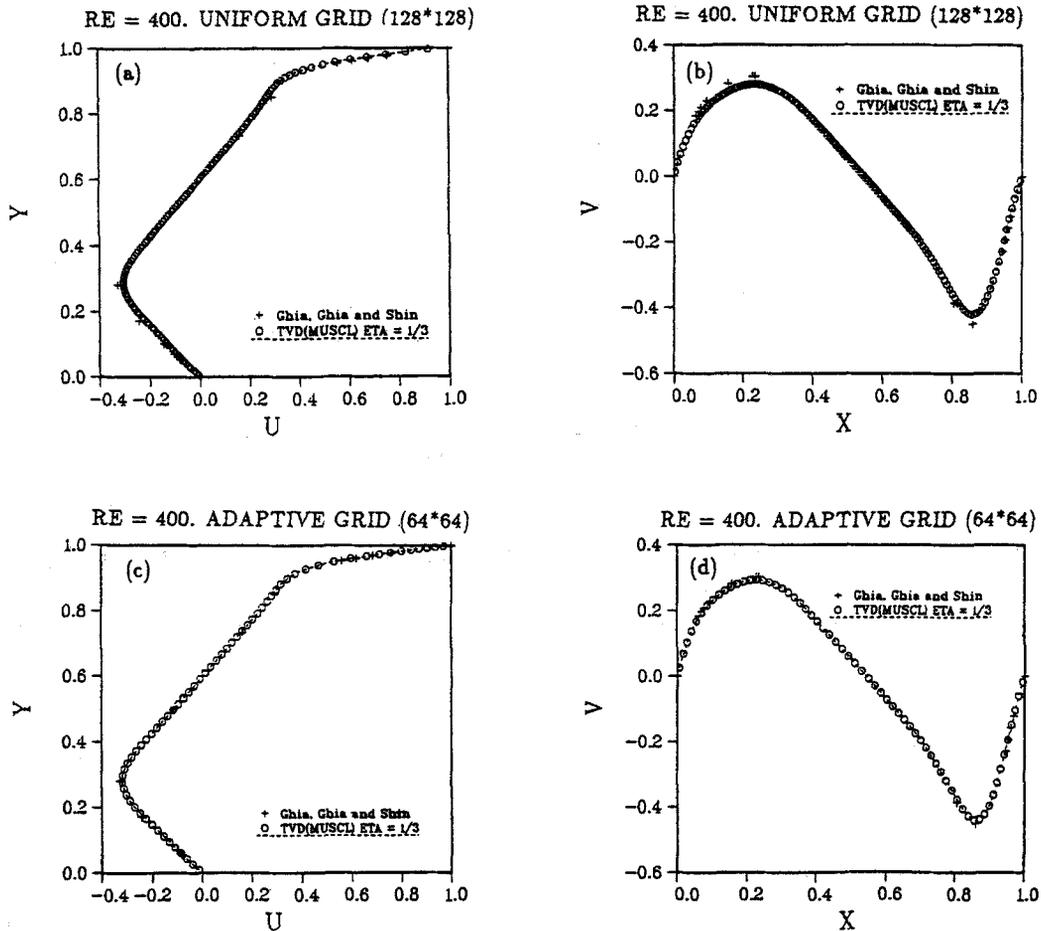


Figure 8. (a) and (c) Comparison of the velocity u along a vertical centreline. (b) and (d) Comparison of the velocity v along a horizontal centreline

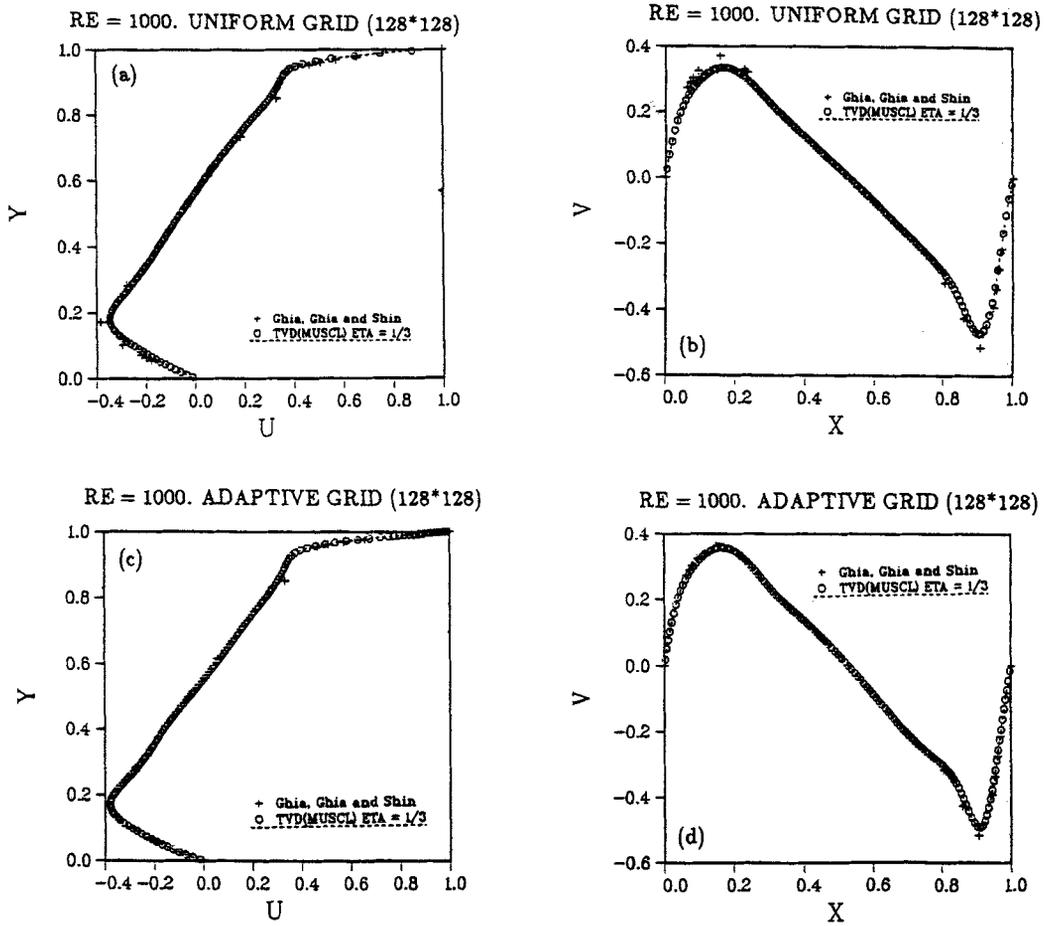


Figure 9. (a) and (c) Comparison of the velocity u along a vertical centreline. (b) and (d) Comparison of the velocity v along a horizontal centreline

adaptive grid 64×64 are better than the results with the uniform grid 128×128 . Figures 9(a)–9(d) also show the same trend for $Re=1000$.

Finally, we compare the convergence rates of scheme 3 with/without implicit residual smoothing in one-, two-, three-, four- and five-level grid systems. The test case chosen simulates the flow for $Re=100$ with 64×64 uniform grid and adaptive grid. The CPU times quoted here are for the HP-720 workstation. Summaries of the performance are given in Tables I–IV. For the single grid, the results show that the scheme with the implicit residual method gives an average saving of 39% on the uniform grid and 50% on the adaptive grid. For the multigrid method, the results show that the convergence rate is increased 16–20 times by using the four- or five-level grid.

3.2. 2D laminar flow through a backward-facing step

The geometry and boundary conditions are shown in Figure 10. The aspect ratio of the backward-facing step (h) to the overall cross-sectional width is 1:2, and the total length in the horizontal direction is $32h$. A fully developed parabolic velocity profile is prescribed at the inlet

Table I. The solutions converge to 10^{-6} , uniform grid (64×64) with the RK2 method

| M | Step | CPU (HP) | Ratio |
|---|--------|----------|-------|
| 1 | 14 325 | 3112.4 | 13.9 |
| 2 | 4625 | 1068.3 | 4.8 |
| 3 | 2050 | 492.9 | 2.2 |
| 4 | 875 | 224.5 | 1.0 |

Table II. The solutions converge to 10^{-6} , uniform grid (64×64) with RK2 and implicit residual smoothing method

| M | Step | CPU (HP) | Ratio |
|---|------|----------|-------|
| 1 | 6275 | 2054.8 | 14.9 |
| 2 | 2250 | 638.3 | 4.6 |
| 3 | 1050 | 323.9 | 2.3 |
| 4 | 425 | 138.8 | 1.0 |
| 5 | 450 | 138.3 | 1.0 |

Table III. The solutions converge to 10^{-5} , adaptive grid (64×64) with RK2

| M | Step | CPU (HP) | Ratio |
|---|--------|----------|-------|
| 1 | 90 400 | 18 461.6 | 24.26 |
| 2 | 26 900 | 7358.6 | 9.67 |
| 3 | 13 375 | 3978.7 | 5.23 |
| 4 | 11 150 | 3281.0 | 4.31 |
| 5 | 2575 | 760.9 | 1.0 |

Table IV. The solutions converge to 10^{-5} , adaptive grid (64×64) with RK2 and implicit residual smoothing method

| M | Step | CPU (HP) | Ratio |
|---|--------|----------|-------|
| 1 | 33 225 | 9166.3 | 20.52 |
| 2 | 11 850 | 4178.9 | 9.35 |
| 3 | 4675 | 1739.8 | 3.89 |
| 4 | 1200 | 454.3 | 1.02 |
| 5 | 1150 | 446.8 | 1.0 |

boundary. The results are compared with the experimental data of Armaly *et al.*¹⁷ and the numerical simulation of Mansour,¹⁸ Ghia¹⁹ and Shon.²⁰ We used the stretched 64×64 grid (see Figure 11) with fine mesh near the wall and near the inlet boundary. The Reynolds numbers selected were 100, 200, 300, 330, 400, 500, 600, 700 and 800. Here, the Reynolds number ($Re = V2h/\nu$) is based on the bulk velocity at the inlet boundary and the cross-section width of the

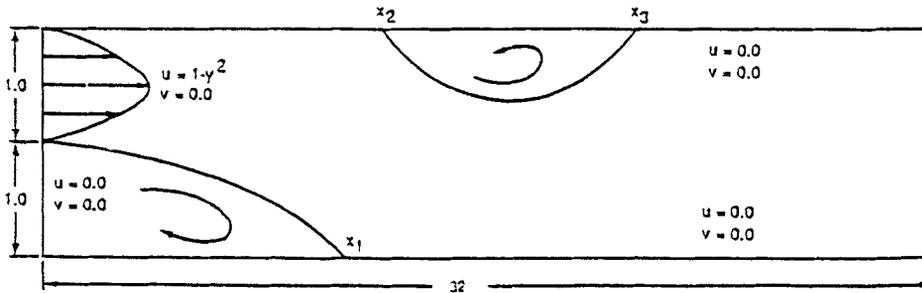


Figure 10. Geometry and boundary conditions of 2D laminar flow through a backward-facing step

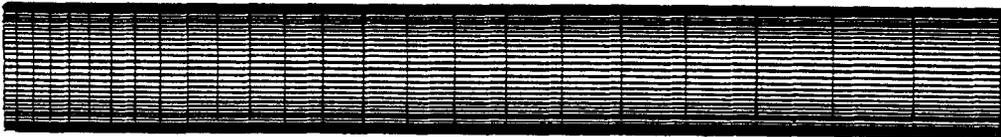
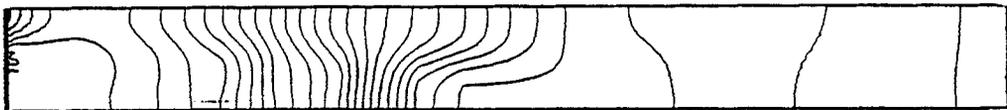


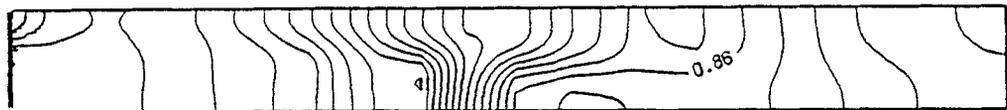
Figure 11. Stretched grids of 2D laminar flow through a backward-facing step



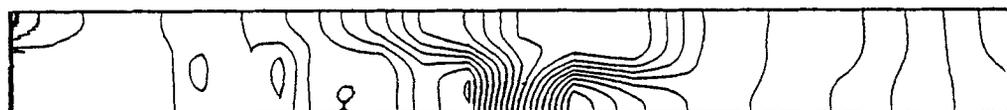
$Re = 200$



$Re = 400$



$Re = 600$



$Re = 800$

Figure 12. Pressure contours of 2D laminar flow through a backward-facing step

whole domain as defined in Armaly *et al.*¹⁷ Pressure contours are shown in Figure 12. As the Reynolds number increases, the position of the point with maximum pressure moves downstream. Figure 13 represents streamline contours for selected Reynolds numbers. The recirculating zone behind a backward-facing step becomes larger as the Reynolds number increases, and another recirculating zone near the upper wall is generated when the Reynolds number is greater than 330 according to our computations with this grid mesh. Comparisons with experimental data¹⁷ and other numerical results¹⁸⁻¹⁹ are shown in Figure 14, where x_1 is the reattachment location of the primary vortex, x_2 is the separation location of the secondary vortex at the top wall and x_3 is the reattachment location of the secondary vortex. The reattachment length is well predicted except at $Re \geq 600$, probably because fluid flow then becomes three-dimensional.

3.3. Inviscid cylinder flow

Flow past a circular cylinder has been a rich source of fluid dynamic problems. The grid system used is 96×96 meshes for both inviscid and viscous flow computations (Figure 15). The total computational region extended to 20 times the radius of the circular cylinder. The characteristic inflow and outflow boundary conditions used are based on the sign of local normal velocity. We used a pressure boundary condition on the inviscid wall of $\partial p / \partial n$. Figure 16 shows a comparison of the pressure coefficients on the cylinder surface with an ideal flow solution for $\beta = 10$ and $\beta = 100$. They agree well with the ideal flow. The pressure contours shown in Figures 17(a) and

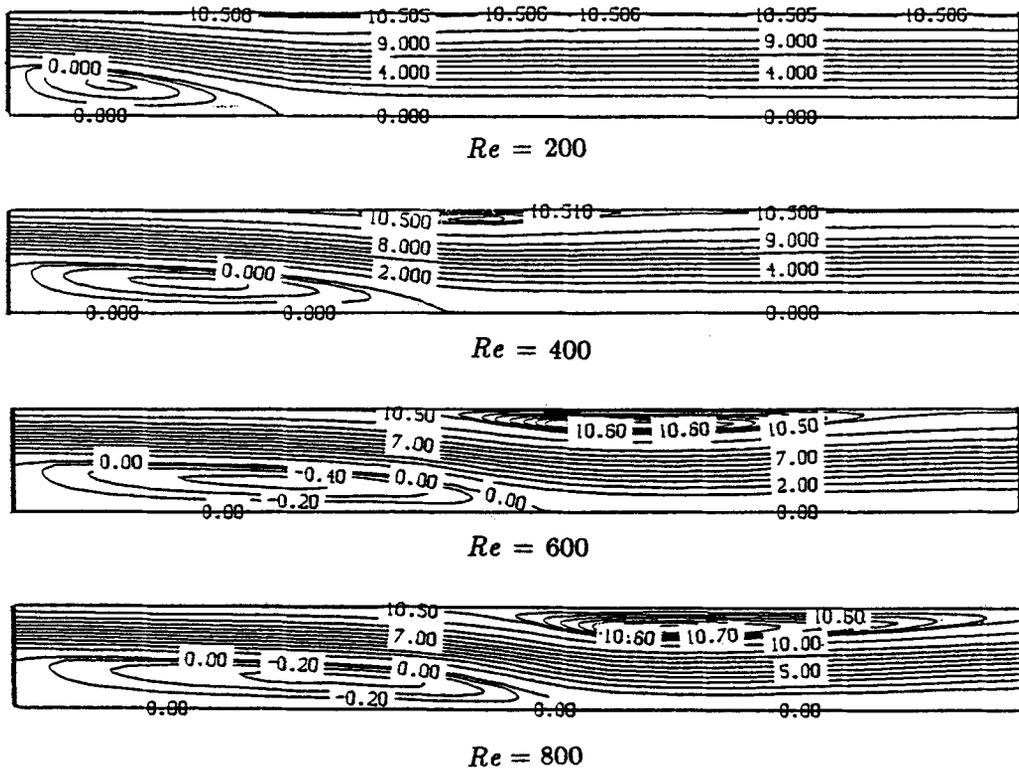


Figure 13. Streamline contours of 2D laminar flow through a backward-facing step

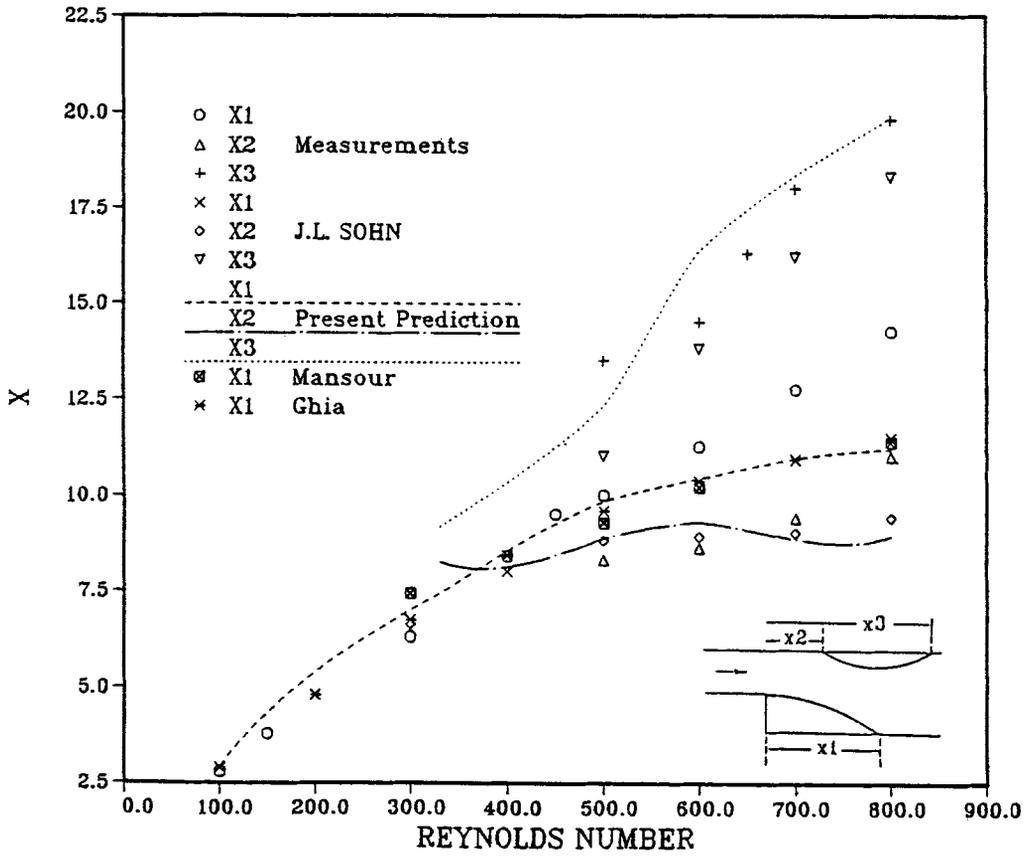


Figure 14. Comparison of reattachment lengths of 2D laminar flow through a backward-facing step

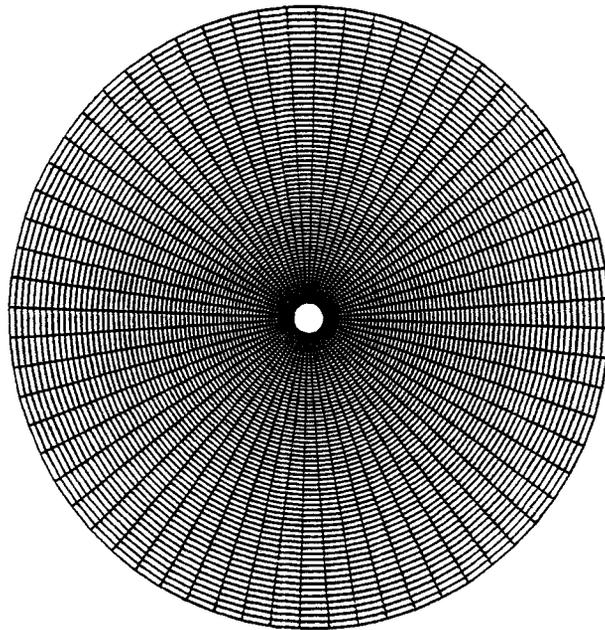


Figure 15. Grid configurations for a circular cylinder

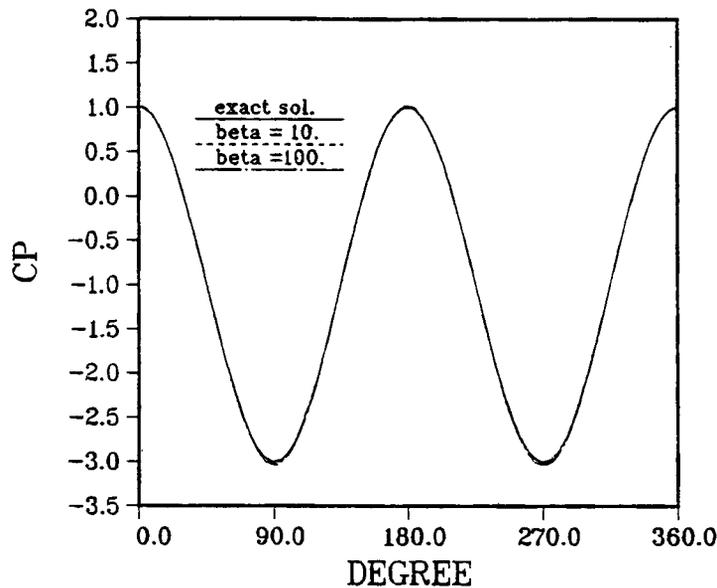


Figure 16. Surface pressure coefficient for inviscid flow over a cylinder

17(b) are symmetric. Again, all the results in this example indicate that the performance of scheme 3 is not sensitive to the value of β .

3.4. Viscous cylinder flow

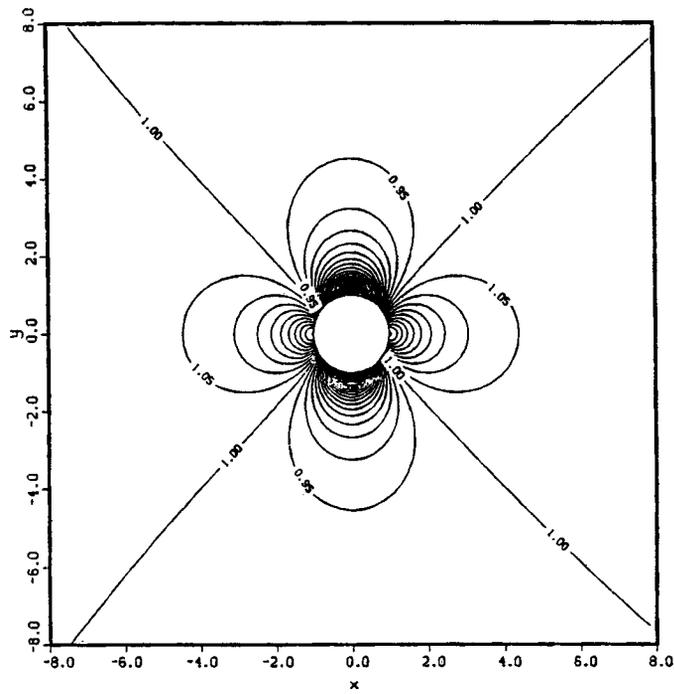
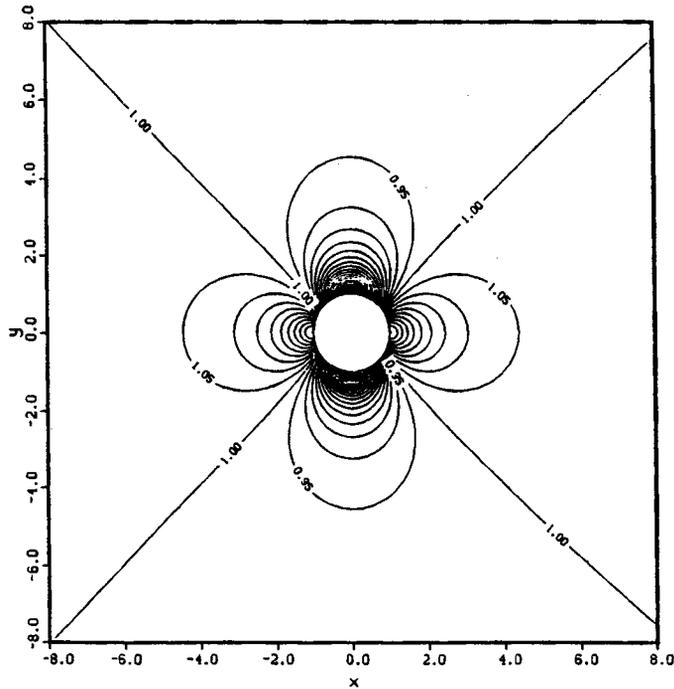
Viscous flow past a circular cylinder has been extensively studied in the last few decades. The flow will be separated somewhere near the cylinder surface. For Reynolds numbers less than 40, the separated flow is steady. In this work, we only simulate cylinder flows with $Re=20$ and $Re=40$. Figure 18 shows the distribution of the wall vorticity, which agrees well with the other numerical results.^{21, 22} In Figure 19 the distribution of the pressure coefficient agrees well the experimental data²³ and other numerical results.^{21, 22}

3.5. Time-dependent test case

The accuracy of the present method has been evaluated by solving a test problem for which an analytical solution is available as a function of time, space and Reynolds number. The Taylor problem was used,^{24, 25} the solution of which is given by the following equations:

$$\begin{aligned} u &= -\cos(2\pi x) \sin(2\pi y) e^{(-2/Re)(4\pi^2)t}, \\ v &= \sin(2\pi x) \cos(2\pi y) e^{(-2/Re)(4\pi^2)t}, \\ p &= -0.25 [\cos(4\pi x) + \cos(4\pi y)] e^{(-4/Re)(4\pi^2)t}. \end{aligned}$$

The Navier–Stokes equations are solved numerically by the present method in the unit square domain $([0, 1]^2)$ and in the square domain $([0.25, 0.75]^2)$. Figures 20(a) and 20(b) show the exact pressure and vorticity contours and Figures 21(a) and 21(b) show the computed pressure and vorticity contours. The evolution of the error as a function of the mesh space Δx , the Reynolds number and the time step Δt is given in Figures 22(a)–22(d).

(a) $\beta = 10$ (b) $\beta = 100$ Figure 17. Pressure contours for inviscid flow over a circular cylinder. (a) $\beta = 10$, (b) $\beta = 100$

The space accuracy is found to be higher than second order, as shown in Figure 22(a). The error as a function of the Reynolds number remains small up to $Re = 10^9$ (Figure 22(b)). The temporal accuracy is found to be complicated. In the large computation domain $([0, 1]^2)$ the order of accuracy is one and a half (Figure 22(c)). On the other hand, in the small computation domain $([0.25, 0.75]^2)$ the overall order of accuracy is higher than two (Figure 22(d)).

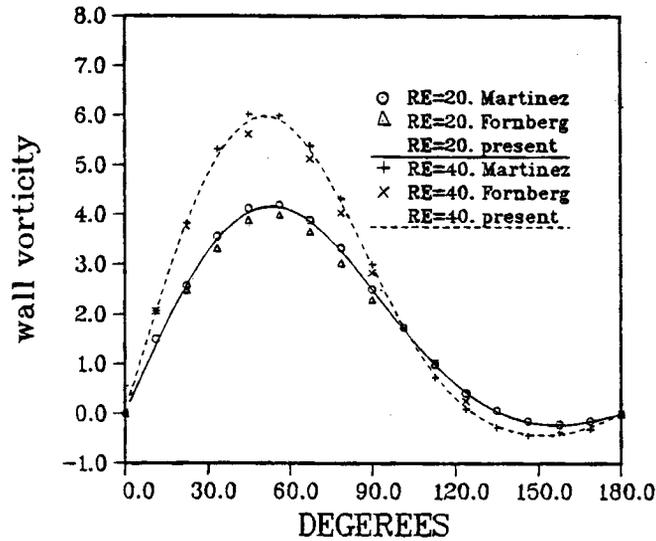


Figure 18. Wall vorticity for viscous flow over a cylinder

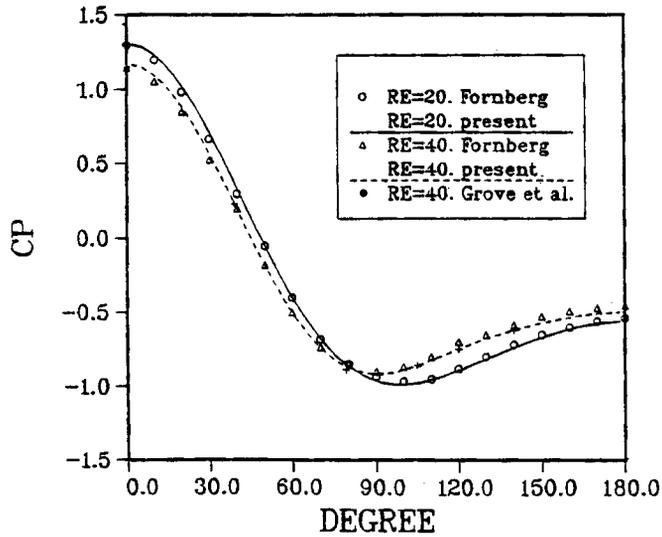


Figure 19. Surface pressure coefficient for viscous flow over a cylinder

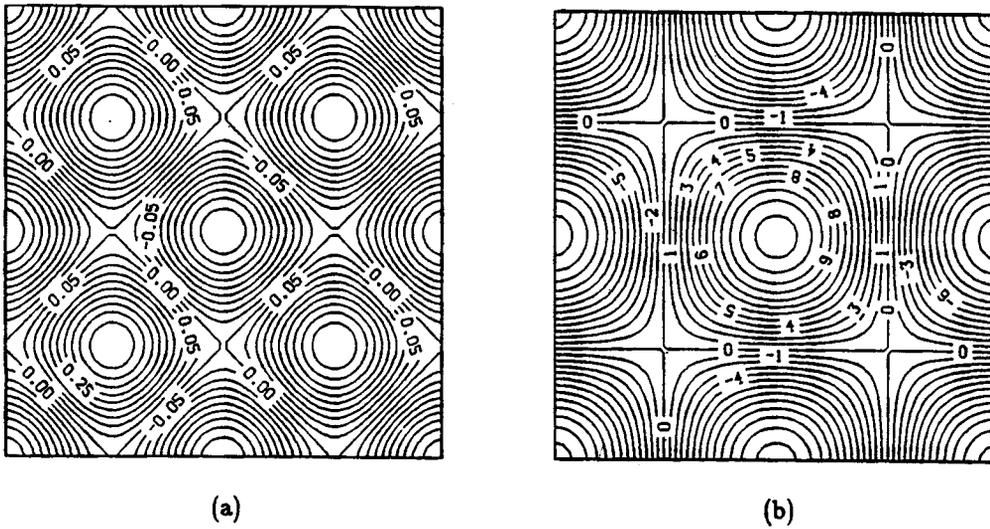


Figure 20. The exact solutions of the example in Section 3.5 with $Re=1000$, $t=1.0$: (a) pressure contours; (b) vorticity contours

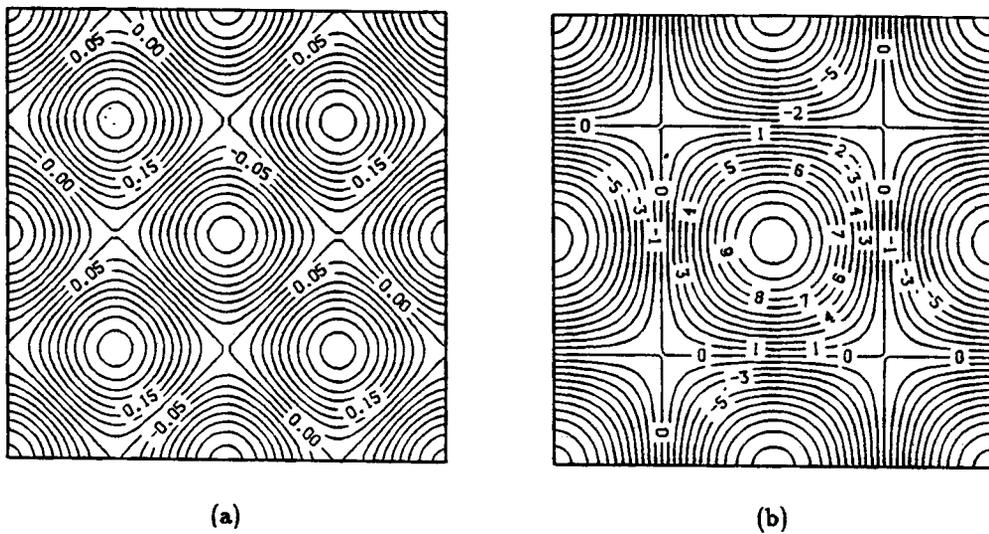


Figure 21. The numerical solutions of the example in Section 3.5 with $Re=1000$, $t=1.0$; $\Delta x = \frac{1}{64}$, $\Delta t = 0.01$: (a) pressure contours; (b) vorticity contours

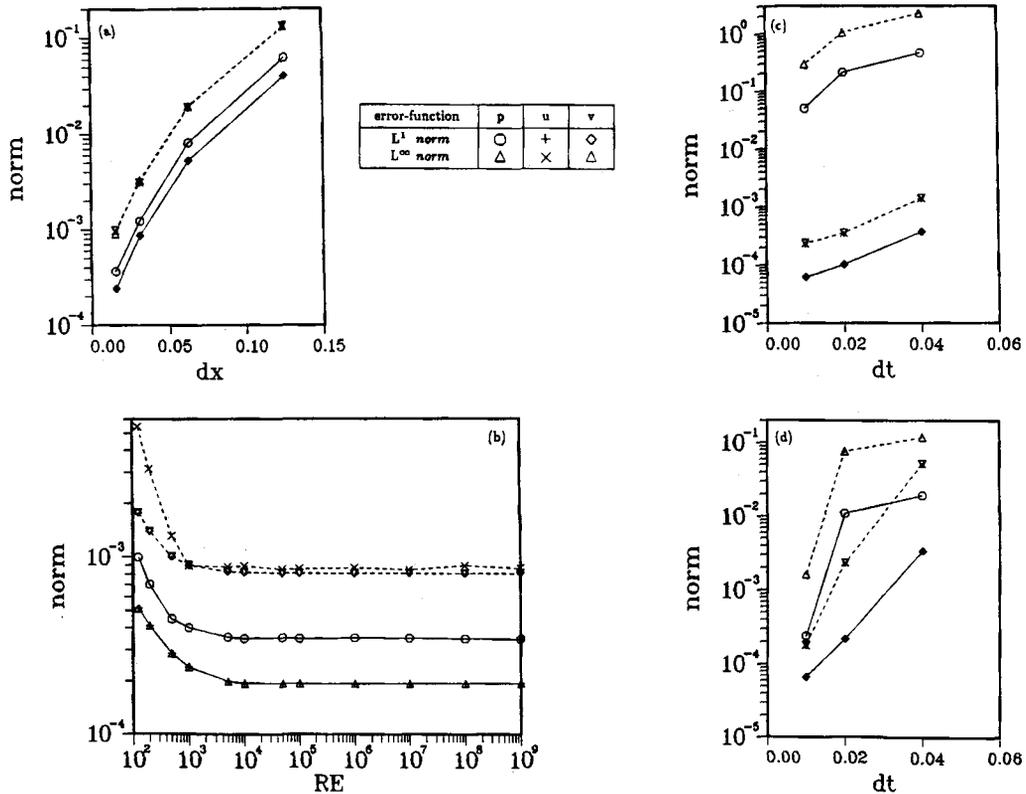


Figure 22. (Example in Section 3.5): (a) the error as a function of Δx ; $\Delta t = 0.01$, $Re = 1000$, $t = 1.0$; (b) the error as a function of Re ; $\Delta x = \frac{1}{64}$, $\Delta t = 0.01$, $t = 1.0$; (c) the error as a function of Δt in the large domain; $Re = 5$, $\Delta x = \frac{1}{32}$, $t = 0.6$; (d) the error as a function of Δt in the small domain; $Re = 5$, $\Delta x = \frac{1}{32}$, $t = 0.6$

4. CONCLUSIONS

An algorithm for computing steady- and unsteady-state solutions to the incompressible Navier–Stokes equation has been presented. The artificial compressibility method allows the equations to be solved as a hyperbolic-dominated system in pseudo-time. The use of upwind difference makes the scheme stable. With the use of two-step Runge–Kutta time integration coupling an implicit residual smoothing and a multigrid method, the code can be run with large time steps and convergence is very fast. Adaptive grid generation is included in the numerical procedure to achieve more accurate results with coarser grid systems. Comparisons of the computational results with some experimental data and other numerical results showed good agreement. The relationship between artificial compressibility and upwind difference was discussed. The overall results show that the performance of scheme 3 is not sensitive to the value of artificial compressibility. The authors are also investigating a series of unsteady flow simulations. Further advances in the convergence speed of the algorithm will still be very useful in increasing the usefulness of this code as a design tool.

ACKNOWLEDGEMENTS

We would like to thank the reviewers for some valuable suggestions. We also thank Professor Y. N. Jeng for providing an adaptive-grid-generation program. The authors would like to

acknowledge support from IAA. This work is partially supported by the National Science Council of R.O.C. under Contract No. NSC81-0401-E-006-606.

REFERENCES

1. A. J. Chorin, 'A numerical method for solving incompressible viscous flow problems', *J. Comput. Phys.*, **2**, 12–26 (1967).
2. D. Pan and S. R. Chakravarthy, 'Unified formulation for incompressible flows', *AIAA Paper 89-0122* 1989.
3. S. E. Rogers, D. Kwak and C. Kiris, 'Numerical solution of the incompressible Navier–Stokes equations for steady-state and time-dependent problems', *AIAA Paper 89-0463*, 1989.
4. B. Van Leer, 'Toward the ultimate conservative difference scheme V, a second-order sequel to Godunov's scheme', *J. Comput. Phys.*, **32**, 101–136 (1979).
5. S. Osher and S. Chakravarthy, 'Very high order accurate TVD schemes', *ICASE Report No. 84-44*, 1984; also *The IMA Volumes in Mathematics and its Applications, Vol. 2*, Springer, Berlin, 1985, pp. 229–274.
6. D. Y. Yeh and J. C. Cheng, private communication.
7. M. Vinokur, 'An analysis of finite-difference and finite-volume formulations of conservation laws', *J. Comput. Phys.*, **81**, 1–52 (1989).
8. Y. N. Jeng and J. L. Chen, 'Truncation error analysis of the finite volume method for a model steady convective equation', *J. Comput. Phys.*, **100**, 64–76 (1992).
9. A. Jameson, 'Multigrid algorithms for compressible flow calculations', in *Multigrid Method II*, Lecture Notes in Mathematics, Vol. 1228, 1985, pp. 166–201.
10. S. Y. Lin, T. M. Wu and Y. S. Chin, 'Upwind finite-volume method with a triangular mesh for conservation laws', *J. Comput. Phys.* (to appear).
11. P. Jorgenson and R. Chima, 'An unconditionally stable Runge–Kutta method for unsteady flows', *AIAA Paper 89-0205*, 1989.
12. M. G. Hall, 'Cell vortex multigrid schemes for solution of the two dimensional Euler equations', *IMA Conf. on Numerical Methods for Fluid Dynamics*, Reading, 1985.
13. H. A. Dwyer, R. J. Kee and B. R. Sanders, 'Adaptive grid method for problems in fluid mechanics and heat transfer', *AIAA J.*, **18**, 1205–1212 (1980).
14. H. A. Dwyer, 'Grid adaptive for problems in fluid dynamics', *AIAA J.*, **22**, 1705–1712 (1984).
15. Y. N. Jeng and S. C. Liou, 'Modified multiple one-dimensional adaptive grid method', *Numer. Heat Transfer*, **15**, 241–247 (1989).
16. U. Ghia, K. N. Ghia and C. T. Shin, 'High-Re solutions for incompressible flow using the Navier–Stokes equations and a multigrid method', *J. Comput. Phys.*, **48**, 387–411 (1982).
17. B. F. Armaly, F. Durst, J. C. F. Pereira and B. Schonung, 'Experiment and theoretical investigation of backward-facing step flow', *J. Fluid Mech.*, **127**, 473–496 (1983).
18. N. N. Mansour and P. Moin, 'Computation of turbulent flow over a backward-facing step', *4th Symp. on Turbulent Shear Flow*, Karlsruhe, Germany, 1983.
19. K. N. Ghia, G. A. Osswald and U. Ghia, 'Analysis of incompressible massively separated viscous flow unsteady Navier–Stokes equations', *Int. j. numer. methods fluids*, **9**, 1025–1050 (1989).
20. J. L. Sohn, 'Evaluation of FIDAP on some classical laminar and turbulent bench-marks', *Int. j. numer. methods fluids*, **8**, 1469–1490 (1988).
21. G. Martinez, *These Docteur-Ingenieur I. N. P. Toulouse* (in German), 1979.
22. B. Fornberg, 'A numerical study of steady viscous flow past a circular cylinder', *J. Fluid Mech.*, **98**, 819–855 (1980).
23. A. S. Grove, F. H. Shair, E. E. Pertersen and A. Acrivos, 'An experimental investigation of the steady separated flow past a circular cylinder', *J. Fluid Mech.*, **19**, 60–80 (1984).
24. G. K. Batchelor (Ed.), *The Collected works of G. I. Taylor, Vol. 2*, Cambridge University Press, Cambridge, 1960.
25. M. Braza, P. Chassaing and H. H. Minh, 'Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder', *J. Fluid Mech.*, **165**, 79–130 (1986).