AIAA JOURNAL
Vol. 39, No. 7, July 2001

Higher-Order Characteristics-Based Method for Incompressible Flow Computation on Unstructured Grids

Yong Zhao* and Chin Hoe Tai†
Nanyang Technological University, Singapore 639798, Republic of Singapore

An algorithm for numerical simulation of incompressible flows on three-dimensional unstructured grids is presented. It is an upwind finite volume method based on the method of characteristics, which is made possible with the introduction of Chorin's artificial compressibility formulation (Chorin, A., "A Numerical Method for Solving Incompressible Viscous Flow Problems," *Journal of Computational Physics*, Vol. 2, No. 1, 1967, pp. 12–26). Flow variables are calculated along characteristics, and their initial values are interpolated based on the signs of the corresponding characteristics. In addition, an upwind-biased interpolation method of third-order accuracy is used for interpolating flow variables on unstructured grids. With these inherent upwinding techniques for evaluating convection fluxes at control volume surfaces, no artificial viscosity is required. The discretized equations are solved by an explicit multistage Runge–Kutta time-stepping scheme, which is found to be efficient in terms of CPU and memory overheads. A computer code has been developed using the numerical methods presented. A number of test cases, including two-dimensional/three-dimensional inviscid and viscous flows, have been calculated to validate the code and to evaluate the performance of the numerical algorithm. Numerical results obtained are in good agreement with exact solutions and other published experimental/numerical results. The convergence rate of numerical simulation is generally found to be satisfactory. The third-order characteristics-based scheme is found to be more accuarate than the second-order central scheme.

I. Introduction

OVER the past decade, both computational fluid dynamics (CFD) algorithm and computer hardware technologies have been developing at a fast pace. Structured-grid solvers have reached a certain degree of maturity, and dramatic improvement in algorithm efficiency is unlikely. In recent years, considerable advancement has been achieved in the development of numerical methods to predict flows around complex geometries on unstructured grids. Compared with its structured counterpart, the unstructured-grid method offers two main advantages: 1) Unstructured grids can be generated around highly complex geometries automatically with much less worker hours compared with the structured grid method. 2) Solution adaption of grids can be easily implemented into the solver (see Ref. 1, for example). These are important features that are very useful when applying CFD to industrial flow problems because generation of structured grids over a complex geometry can take several months, whereas running the solver takes just a small fraction of that time. However, the unstructured-grid method also has its own drawbacks, which include relatively low accuracy, low efficiency, and excessive memory requirements due to the random nature of its data structure.

Traditional finite volume or finite difference incompressible flow solvers are typically based on the pressure correction scheme on structured grids.[2] The artificial compressibility method of Chorin[3] can be considered as an alternative, among others. In this method, a pseudotime derivative of pressure is added to the continuity equation, thus turning the system of governing equations into hyperbolic ones (viscous terms are excluded). Therefore, the pressure field is closely coupled with the velocity field at the same time level. With time marching, the artificial waves created are used for propagating information throughout the solution domain and driving the divergence of velocity to zero. This approach has been used successfully by some researchers.[1,4–9] Among them, Farmer et al.[8] and Chen and Kallinderis[1] have used a central scheme with artificial viscosity to prevent the pressure–velocity decoupling. Although this approach is relatively simple, one has to face the task of adjusting coefficients in the artificial viscosity terms to obtain stable results while minimizing the contamination of unphysical artificial viscosity. Rogers and Kwak[5] used a Roe-type upwinding scheme with a simple averaging of the left and right variables on a body-fitted grid system. The Jacobian matrix and left and right eigenvectors, as well as the associated metric coefficients, are quite complex. Recently, this has been extended to three-dimensional unstructured-grid solvers by Anderson et al.[10]

In this study, a different upwind method for calculating incompressible flows is developed based on the method of characteristics. Traditionally, the method of characteristics is used primarily for compressible flow calculations.[11] However, with the introduction of the pseudotime derivative in Chorin's formulation,[3] it is possible that the incompressible equations can also be solved by a similar method of characteristics. This idea was tested in Ref. 12 in the solution of the two-dimensional incompressible flow equations cast in a curvilinear coordinate system for structured grids. Based on this idea, a high-order upwind method has been developed to calculate three-dimensional incompressible flows on unstructured tetrahedral grids. It is shown that this method is mathematically simpler than the Roe-type method and can be readily applied to arbitrary three-dimensional unstructured grids in the finite volume framework, together with high-order schemes such as the upwind-biased interpolation. A computer code has been developed using the numerical methods presented. A number of test cases, including two-dimensional and three-dimensional inviscid and viscous flows, have been calculated to validate the code and evaluate the performance of the numerical algorithm. Numerical results obtained are in good agreement with exact solutions and other published results and measurements. Convergence is found to be satisfactory for all cases studied. The third-order characteristics-based scheme is also found to be less diffusive that the second-order central scheme with artificial viscosity with a very small coefficient.

II. Governing Equations

The Navier–Stokes equations for incompressible flows are derived from the conservation laws of mass and momentum. The Navier–Stokes (NS) equations modified by the artificial compressibility method can be expressed as

$$\frac{\partial \boldsymbol{W}}{\partial t} + \nabla \cdot \boldsymbol{F}_c = \nabla \cdot \boldsymbol{F}_v \tag{1}$$

Received 4 February 1998; revision received 10 March 2000; accepted for publication 20 October 2000.

*Associate Professor, School of Mechanical and Production Engineering, Nanyang Avenue. Member AIAA.

†Graduate Student, School of Mechanical and Production Engineering, Nanyang Avenue.

where

$$\boldsymbol{W} = \begin{bmatrix} p \\ \boldsymbol{U} \end{bmatrix}, \qquad \boldsymbol{F}^c = \begin{bmatrix} \beta \boldsymbol{U} \\ \boldsymbol{U}\boldsymbol{U} + p\delta_{ij} \end{bmatrix}, \qquad \boldsymbol{F}^v = \begin{bmatrix} 0 \\ \tau_{ij} \end{bmatrix}$$

where $\boldsymbol{W}$ is the vector of dependent variables, $\boldsymbol{U}$ is the velocity vector, and i and $j = 1, 2$, and 3. $\boldsymbol{F}^c$ and $\boldsymbol{F}^v$ are the convective flux and viscous flux vectors, β is a constant called artificial compressibility whose value affects the solution convergence to steady state, and p denotes pressure in the fluid. The viscous stress tensor is given in the following:

$$\tau_{ij} = \frac{1}{Re}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \tag{2}$$

where Re is the Reynolds number.

The preceding equations are actually written in nondimensional form based on the following scaling:

$$(x, y, z) = \left(\frac{x^*}{l^*}, \frac{y^*}{l^*}, \frac{z^*}{l^*}\right), \qquad t = \frac{t^*}{l^* \big/ U_\infty^*}$$

$$(u, v, w) = \left(\frac{u^*}{U_\infty^*}, \frac{v^*}{U_\infty^*}, \frac{w^*}{U_\infty^*}\right), \qquad p = \frac{p^* - p_\infty^*}{\rho_\infty \left(U_\infty^*\right)^2} \tag{3}$$

where l^* denotes a reference length, superscript $*$ indicates dimensional quantities, and subscript ∞ indicates freestream conditions.

III. Solution Algorithm

A. Finite Volume Method

Equations (1) are discretized on an unstructured tetrahedral grid. A cell-vertex scheme is adopted here, that is, all computed variables in vector $\boldsymbol{W}$ are stored at vertices of the tetrahedral cells. For every vertex, a control volume is constructed using the median dual of the tetrahedral grid, as shown in Fig. 1.

Spatial discretization is performed by using the integral form of the NS equations over the control volume surrounding node or vertex p:

$$\frac{\partial}{\partial t}\iiint_{cv} W_p \,\mathrm{d}v + \iiint_{cv} \nabla \cdot \boldsymbol{F}_c \,\mathrm{d}v = \iiint_{cv} \nabla \cdot \boldsymbol{F}_v \,\mathrm{d}v \tag{4}$$

The convection term is transformed to introduce the upwind scheme using an edge-based procedure:

$$\iiint_{cv} \nabla \cdot \boldsymbol{F}_c \,\mathrm{d}v = \oint_{S_{cv}} \boldsymbol{F}_c \cdot \boldsymbol{n} \,\mathrm{d}s = \sum_{n=1}^{n\,\mathrm{edge}} [(\boldsymbol{F}_c)_{ij} \cdot \boldsymbol{n} \Delta S]_n$$

where ΔS_n is the part of control volume surface associated with edge n (similar to edge PA as shown in Fig. 2).

The viscous term is calculated in a cell-based method:

$$\iiint_{cv} \nabla \boldsymbol{F}_v \,\mathrm{d}v = \oint_{S_{cv}} \boldsymbol{F}_v \cdot \,\mathrm{d}\boldsymbol{s} = \sum_{i}^{n\,\mathrm{cell}} (\boldsymbol{F}_v \cdot \Delta \boldsymbol{S}_c)_i$$

ΔS_{ci} is the part of control volume surface in cell i (represented by C_{ij} in Fig. 1). By the use of the relation

$$\oint_s \mathrm{d}\boldsymbol{S} = 0 \tag{5}$$

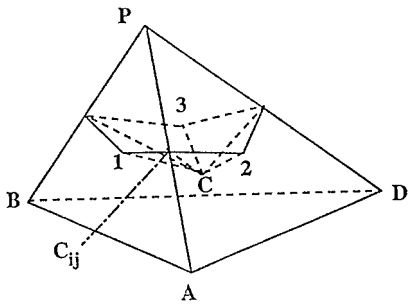


Fig. 1 Portion of a control volume within a tetrahedron for a node *P*.

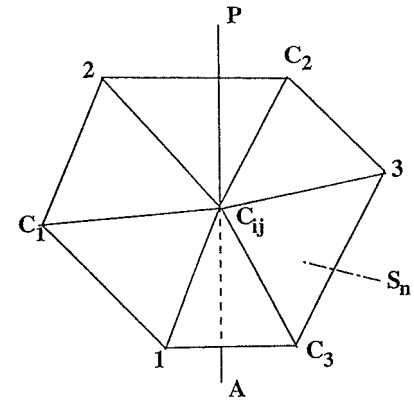


Fig. 2 Control volume surface and the associated edge.

the total vector surface of the control volume in a cell i becomes

$$\Delta \boldsymbol{S}_{ci} = \tfrac{1}{3}\Delta \boldsymbol{S}_{pi}$$

Thus, the calculation of viscous terms can be simplified as

$$\oint_{S_{cv}} \boldsymbol{F}_v \,\mathrm{d}\boldsymbol{s} = \sum_{i}^{n\,\mathrm{cell}} (\boldsymbol{F}_v \cdot \Delta \boldsymbol{S}_c)_i = \frac{1}{3}\sum_{i}^{nb\,\mathrm{cell}} (\boldsymbol{F}_v \cdot \Delta \boldsymbol{S}_p)_i$$

ΔS_{pi} is the surface vector of the face opposite node p of the tetrahedron under consideration. Here the $(\boldsymbol{F}_v)_i$ is calculated at the center of the tetrahedron with a vertex p, which can be obtained by using Green's theorem based on the variables at the four vertices of the tetrahedron. Similar to the Galerkin type of formulation, the gradient of a flow variable ϕ, at the center of a tetrahedron is

$$\operatorname{grad} \phi_c = -\left(\sum_{i=1}^{4} \phi_i 9 \boldsymbol{S}_i \Big/ 27V\right) = -\frac{1}{3}\left(\sum_{i=1}^{4} \phi_i \boldsymbol{S}_i \Big/ V\right)$$

where ϕ_i is the flow variable at a vertex i of the tetrahedron and $\boldsymbol{S}_i$ is the surface vector that is opposite to node i. V is the volume of the tetrahedron. Gradients at vertices are obtained by a volume averaging of the gradients at centers of cells associated with the vertex under consideration, as shown in Ref. 13.

B. Upwind-Biased Interpolation

Here, an edge-based method of calculating the total inviscid flux is adopted by calculating and storing the flux integrals based on the edges. Such a treatment leads to higher efficiency in computation and reduced data storage requirements.

The left and right state vectors $\boldsymbol{W}_L$ and $\boldsymbol{W}_R$ at a control volume surface are evaluated using a nominally third-order upwind-biased interpolation scheme. If the left and right state vectors are set to $\boldsymbol{W}_i$ and $\boldsymbol{W}_j$ (i and j corresponding to the two ends of an edge, for example, P and A as shown in Fig. 2), it is a first-order upwind scheme:

$$\boldsymbol{W}_L = \boldsymbol{W}_i + \tfrac{1}{4}\left[(1-\kappa)\Delta_i^- + (1+\kappa)\Delta_i^+\right]$$

$$\boldsymbol{W}_R = \boldsymbol{W}_j - \tfrac{1}{4}\left[(1-\kappa)\Delta_j^+ + (1+\kappa)\Delta_j^-\right]$$

Here

$$\begin{aligned}
\Delta_i^+ &= \boldsymbol{W}_j - \boldsymbol{W}_i \\
\Delta_i^- &= \boldsymbol{W}_i - \boldsymbol{W}_{i-1} \\
&= 2\boldsymbol{ij} \cdot \nabla \boldsymbol{W}_i - (\boldsymbol{W}_j - \boldsymbol{W}_i) \\
&= 2\boldsymbol{ij} \cdot \nabla \boldsymbol{W}_i - \Delta_i^+ \\
\Delta_j^- &= \boldsymbol{W}_j - \boldsymbol{W}_i = \Delta_i^+ \\
\Delta_j^+ &= \boldsymbol{W}_{j+1} - \boldsymbol{W}_j \\
&= 2\boldsymbol{ij} \cdot \nabla \boldsymbol{W}_j - (\boldsymbol{W}_j - \boldsymbol{W}_i) \\
&= 2\boldsymbol{ij} \cdot \nabla \boldsymbol{W}_j - \Delta_j^-
\end{aligned}$$

Therefore,

$$\boldsymbol{W}_L = \boldsymbol{W}_i + \tfrac{1}{2}\left[(1-\kappa)\boldsymbol{ij}\cdot\nabla\boldsymbol{W}_i + \kappa\Delta_i^+\right]$$

$$\boldsymbol{W}_R = \boldsymbol{W}_j - \tfrac{1}{2}\left[(1-\kappa)\boldsymbol{ij}\cdot\nabla\boldsymbol{W}_j + \kappa\Delta_j^-\right]$$

where κ is set to $\frac{1}{3}$, which corresponds to a nominally third-order accurate scheme.

C. Upwind Characteristics Method

The Euler equations modified by the Chorin's method[3] are rewritten in partial differential form in a Cartesian coordinate system for the derivation of the method of characteristics:

$$\frac{\partial p}{\partial t} + \beta\frac{\partial u_j}{\partial x_j} = 0, \qquad \frac{\partial u_i}{\partial t} + u_j\frac{\partial u_i}{\partial x_j} + u_i\frac{\partial u_j}{\partial x_j} + \frac{\partial p}{\partial x_i} = 0$$

Here, subscripts i and j equal 1, 2, and 3, representing the three coordinates. Suppose that ξ is a new coordinate normal to the surface of a control volume that surrounds a particular edge (outward normal direction). To extend the method of characteristics to the three-dimensional unstructured-grid solver, it is assumed that flow in ξ direction is approximately one dimensional and that the preceding equations can then be transformed into

$$\frac{\partial p}{\partial t} + \beta\frac{\partial u_j}{\partial \xi}\xi_{x_j} = 0 \tag{6}$$

$$\frac{\partial u_i}{\partial t} + u_j\frac{\partial u_i}{\partial \xi}\xi_{x_j} + u_i\frac{\partial u_j}{\partial \xi}\xi_{x_j} + \frac{\partial p}{\partial \xi}\xi_{x_i} = 0 \tag{7}$$

where

$$\xi_{x_i} = \frac{\partial \xi}{\partial x_i}, \qquad \xi_{x_j} = \frac{\partial \xi}{\partial x_j}$$

In the t–ξ space as shown in Fig. 3, flow variables $\boldsymbol{W}$ at time level $n+1$ can be calculated along a characteristics k using a Taylor series expansion and the initial value at time level n ($\boldsymbol{W}^k$):

$$\boldsymbol{W} = \boldsymbol{W}^k + \boldsymbol{W}_\xi\xi_t\Delta t + \boldsymbol{W}_t\Delta t, \qquad \boldsymbol{W}_t = [(\boldsymbol{W}-\boldsymbol{W}^k)/\Delta t] - \boldsymbol{W}_\xi\xi_t$$

A wave speed λ^k is introduced:

$$\xi_t = \lambda^k\sqrt{\xi_{x_i}\xi_{x_i}}$$

and the normal vector components are

$$n_{x_j} = \xi_{x_j}\Big/\sqrt{\xi_{x_i}\xi_{x_i}}$$

Substitute components of $\boldsymbol{W}_t$ into Eqs. (6) and (7), and we have

$$\frac{1}{\sqrt{\xi_{x_i}\xi_{x_i}}}\frac{(p-p^k)}{\Delta t} - p_\xi\lambda^k + \beta(u_\xi n_x + v_\xi n_y + w_\xi n_z) = 0 \tag{8}$$

$$\frac{1}{\sqrt{\xi_{x_i}\xi_{x_i}}}\frac{(u-u^k)}{\Delta t} - u_\xi(\lambda^0-\lambda^k) + u(u_\xi n_x + v_\xi n_y + w_\xi n_z) + p_\xi n_x = 0 \tag{9}$$

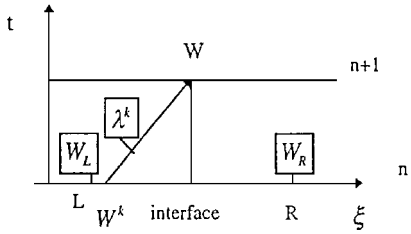


Fig. 3 Coordinates for t–ξ.

$$\frac{1}{\sqrt{\xi_{xj}\xi_{xj}}}\frac{(v-v^k)}{\Delta t} - v_\xi(\lambda^0-\lambda^k) + v(u_\xi n_x + v_\xi n_y + w_\xi n_z) + p_\xi n_y = 0 \tag{10}$$

$$\frac{1}{\sqrt{\xi_{xj}\xi_{xj}}}\frac{(w-w^k)}{\Delta t} - w_\xi(\lambda_0-\lambda^k) + w(u_\xi n_x + v_\xi n_y + w_\xi n_z) + p_\xi n_z = 0 \tag{11}$$

where λ^0 is the contravariant velocity

$$\lambda^0 = un_x + vn_y + wn_z$$

To derive the compatibility equations, the spatial derivatives, such as u_ξ, v_ξ, w_ξ, and p_ξ, have to be eliminated from the preceding equations. With the use of the approach of Eberle[11] for compressible flow equations, each of the preceding four equations is multiplied by an arbitrary variable, and all of the resulting equations are summed to form a new equation:

$$\frac{1}{\Delta t\sqrt{\xi_{xj}\xi_{xj}}}A + p_\xi B + u_\xi C + v_\xi D + w_\xi E = 0 \tag{12}$$

where

$$A = a(p-p^k) + b(u-u^k) + c(v-v^k) + d(w-w^k)$$

$$B = -a\lambda^k + bn_x + cn_y + dn_z$$

$$C = an_x\beta + b\left(\lambda^0 - \lambda^k + un_x\right) + cvn_x + dwn_x$$

$$D = an_y\beta + bun_y + c\left(\lambda^0 - \lambda^k + vn_y\right) + dwn_y$$

$$E = an_z\beta + bun_z + cvn_z + d\left(\lambda^0 - \lambda^k + wn_z\right)$$

and a, b, c, and d are the arbitrary variables used to multiply the equations. We define the coefficients of the partial space derivatives to be zero, that is, A, B, C, D, and E are zero. The following equations are, thus, obtained:

$$A = a(p-p^k) + b(u-u^k) + c(v-v^k) + d(w-w^k) = 0 \tag{13}$$

$$B = 0 \tag{14}$$

$$C = 0 \tag{15}$$

$$D = 0 \tag{16}$$

$$E = 0 \tag{17}$$

Equations (13–17) can be represented as a linear system $\Phi X = 0$ with $X = \{a, b, c, d\}$. Variables a, b, c, and d are generally nonzero. Thus, the system of equations has a nontrivial solution. This means that $\det(\Phi) = 0$, and we can derive the following eigenvalues based on this condition:

$$\lambda_1 = \lambda_2 = \lambda^0, \qquad \lambda_3 = \lambda^1 = \lambda^0 + \sqrt{(\lambda^0)^2 + \beta} = \lambda^0 + c$$

$$\lambda_4 = \lambda^2 = \lambda^0 - \sqrt{(\lambda^0)^2 + \beta} = \lambda^0 - c$$

For each eigenvalue or characteristics speed, characteristic equations can be derived from Eqs. (13–17). For example, for $\lambda^k = \lambda^0$, we have

$$a = (bn_x + cn_y + dn_z)\Big/\lambda^0$$

Substituting this into Eq. (13), we obtain

$$(bn_x + cn_y + dn_z)/\lambda_0(p-p^0) + b(u-u^0) + c(v-v^0) + d(w-w^0) = 0$$

that is,

$$b[n_x(p-p^0)+\lambda^0(u-u^0)]+c[n_y(p-p^0)+\lambda^0(v-v^0)]$$
$$+d[n_z(p-p^0)+\lambda^0(w-w^0)]=0$$

For any b, c, and d, the preceding equations are always satisfied. Therefore, all of the terms in square brackets are zero. As a result, we have

$$(u-u^0)n_y-(v-v^0)n_x=0 \tag{18}$$

$$(v-v^0)n_z-(w-w^0)n_y=0 \tag{19}$$

$$(u-u^0)n_z-(w-w^0)n_x=0 \tag{20}$$

For $\lambda=\lambda^1$

$$p-p^1=-\lambda^1\left[(u-u^1)n_x+(v-v^1)n_y+(w-w^1)n_z\right] \tag{21}$$

For $\lambda=\lambda^2$

$$p-p^2=-\lambda^2\left[(u-u^2)n_x+(v-v^2)n_y+(w-w^2)n_z\right] \tag{22}$$

Finally, u, v, w, and p are determined using the preceding characteristics equations (18–22):

$$u=fn_x+u^0\left(n_y^2+n_z^2\right)-v^0n_xn_y-w^0n_xn_z$$

$$v=fn_y+v^0\left(n_x^2+n_z^2\right)-w^0n_yn_z-u^0n_yn_x$$

$$w=fn_z+w^0\left(n_x^2+n_y^2\right)-u^0n_zn_x-v^0n_zn_y$$

$$p=p^1-\lambda^1\left[n_x(u-u^1)+n_y(v-v^1)+n_z(w-w^1)\right]$$

or

$$p=p^2-\lambda^2\left[n_x(u-u^2)+n_y(v-v^2)+n_z(w-w^2)\right]$$

where

$$c=\sqrt{(\lambda^0)^2+\beta}$$

$$f=1/2c\left[(p-p^0)+n_x(\lambda^1u^1-\lambda^2u^2)\right.$$
$$\left.+n_y(\lambda^1v^1-\lambda^2v^2)+n_z(\lambda^1w^1-\lambda^2w^2)\right]$$

Flow quantities at the $(n+1)$ time level obtained from the preceding equations on the characteristics are then used to calculate convection fluxes at the control volume interface, whereas those on different characteristics at the n time level are approximately evaluated by an upwind scheme using the signs of the characteristics:

$$\boldsymbol{W}^j=\tfrac{1}{2}\left\{[1+\text{sign}(\lambda^j)]\boldsymbol{W}_L+[1-\text{sign}(\lambda^j)]\boldsymbol{W}_R\right\}$$

and $\boldsymbol{W}_L$ and $\boldsymbol{W}_R$ are obtained by the upwind-biased interpolation.

D. Time Integration

Finally, for a certain node p, Eq. (4) is discretized as

$$\frac{\partial}{\partial t}W_p\Delta V_{cv}=R(W_p) \tag{23}$$

$$R(W_p)=-\sum_{n=1}^{n\,\text{edge}}[(\boldsymbol{F}_c)_{ij}\cdot\boldsymbol{n}\Delta S]_n+\frac{1}{3}\sum_{i=1}^{nb\,\text{cell}}(\boldsymbol{F}_v\boldsymbol{n}\Delta S_c)_i+(S_t)_p\Delta V_{cv} \tag{24}$$

The spatially discretized equations form a set of coupled ordinary differential equations that are integrated in time by means of an explicit multistage Runge–Kutta scheme. A general m-stage scheme to advance the solution from the nth time step to the next step can be written as

$$\boldsymbol{W}^{(0)}=\boldsymbol{W}^n$$

$$\boldsymbol{W}^{(1)}=\boldsymbol{W}^{(0)}+\alpha_1\frac{\Delta t}{\Delta V_{cv}}R^{(0)}(\boldsymbol{W})$$

$$\vdots$$

$$\boldsymbol{W}^{(m-1)}=\boldsymbol{W}^{(0)}+\alpha_{m-1}\frac{\Delta t}{\Delta V_{cv}}R^{(m-2)}(\boldsymbol{W})$$

$$\boldsymbol{W}^{(m)}=\boldsymbol{W}^{(0)}+\alpha_m\frac{\Delta t}{\Delta V_{cv}}R^{(m-1)}(\boldsymbol{W})$$

$$\boldsymbol{W}^{(n+1)}=\boldsymbol{W}^{(m)}$$

For a five-stage scheme, the stage coefficients are

$$\alpha_1=\tfrac{1}{4},\qquad \alpha_2=\tfrac{1}{6},\qquad \alpha_3=\tfrac{3}{8},\qquad \alpha_4=\tfrac{1}{2},\qquad \alpha_5=1$$

In the present code, the viscous fluxes are updated at every time step. As an explicit scheme, the maximum time step is determined by a stability criterion of the numerical method. Although it is impossible to derive an analytical stability condition for general three-dimensional problems, the Courant–Friedrichs–Lewy (CFL) condition for hyperbolic equations can be used to derive such a condition by analyzing a linearized model equation. This condition can then be extended approximately to general three-dimensional problems. For NS computations using an explicit multistage Runge–Kutta scheme, the local time step size is estimated as

$$\Delta t=\text{CFL}\frac{\Delta l}{\max(\lambda^j)},\qquad j=0,1,2$$

where the Δl length scale is associated with a node under consideration. Normally, it is taken as the smallest height of all of the cells sharing the node.

E. Implicit Residual Smoothing

The idea behind this is to replace the residual at one point of the flowfield with a smoothed or weighted average of the residuals at the neighboring points. The averaged residuals are calculated implicitly to increase the maximum CFL number, thus increasing the convergence rate. Normally this procedure allows the CFL number to be increased by a factor of 2 or 3. The smoothing equation for a node k can be expressed as

$$\bar{R}_k=R_k+\epsilon\delta^2\bar{R}_k \tag{25}$$

where $\bar{R}$ is the smoothed residual, R is the original residual, and ϵ is the smoothing coefficient:

$$\epsilon=\max\left\{\tfrac{1}{4}[(\text{CFL}/\text{CFL}^*)^2-1],0.0\right\}$$

Here CFL^* is the maximum CFL number of the basic scheme. The solution to the preceding equations can be obtained on an unstructured grid by the Jacobi iterative method:

$$\overline{R_k^{(m)}}=R_k^{(0)}+\epsilon\sum_{i=1}^{\text{numnod}\,(k)}\left[\bar{R}_i^{(m)}-\bar{R}_k^{(m)}\right]$$

that is,

$$\overline{R_k^{(m)}}=\frac{R_k^{(0)}+\epsilon\sum_{i=1}^{\text{numnod}\,(k)}\bar{R}_i^{(m-1,m)}}{1+\epsilon\cdot\text{numnod}(k)}$$

where $\text{numnod}(k)$ is the number of neighboring nodes of node k.

F. Boundary Conditions

At the solid wall, slip (for inviscid flows) or no-slip (for viscous flows) and no-injection boundary conditions are imposed, that is, the zero normal fluxes of mass, momentum, and energy are imposed. In addition, the solid surface is assumed to be adiabatic, and the

pressure gradient normal to the wall at the surface is considered to be zero.

For a far-field or upstream boundary, the flow velocity is given directly, and gradients of variables are assumed to be zero.

G. Numerical Grid Generation

The grid generation process can be divided into the following subprocesses: 1) geometry definition, 2) surface grid and initial volume grid generation, 3) point insertion, 4) clustering of grid points, and 5) smoothing of the grid.

Here, the geometry definition is done by a commercial CAD system, and the surface and initial volume grid are generated by using an edge swapping scheme.[14] It is necessary to further modify the initial grids. This will involve point insertion, clustering, and smoothing of grids, all based on the edge swapping method.

The clustering of grid points for viscous flow simulation is carried out based on the distance of the point to a solid wall, that is, the shortest distance to all of the wall nodes. A function f is defined as an exponential function of the distance d:

$$f(d) = 1/\exp^{\beta}[(d - dd)/dd]$$

where dd is a length scale of the flowfield and β is a constant less than unit one. We introduce another function for a cell with an area A (or a volume in three dimensions):

$$\Phi = Af(d)$$

Then the average of Φ over the field and its deviation can be calculated as follows:

$$\bar{\Phi} = \frac{1}{N}\sum_{i=1}^{N}\Phi_i, \qquad \sigma = \sqrt{\left[\sum_{i=1}^{N}(\bar{\Phi} - \Phi_i)^2\right] \Big/ N}$$

For any cells whose Φ is greater than $\bar{\Phi} + \sigma$, a point is inserted to the center of its circumcircle or circumsphere and connected to the vertices of the cell, and then the mesh is reconnected based on the edge swapping algorithm.[14]

Similar to the implicit smoothing of the residual, the Laplacian smoother is employed to redistribute the grid nodes $r(x, y, z)$:

$$\bar{\boldsymbol{r}}_{\boldsymbol{k}} = \boldsymbol{r}_k + \epsilon\delta^2\bar{\boldsymbol{r}}_{\boldsymbol{k}} \tag{26}$$

where $\bar{\boldsymbol{r}}$ is the smoothed node coordinate vector, $\boldsymbol{r}$ is the original vector, and ϵ is the smoothing coefficient. The solution to preceding equations can be obtained on an unstructured grid by the Jacobi iterative method:

$$\overline{\boldsymbol{r}_k^{(m)}} = \boldsymbol{r}_k^{(0)} + \epsilon \sum_{i=1}^{\text{numnod}(k)} \left[\bar{\boldsymbol{r}}_i^{(m)} - \bar{\boldsymbol{r}}_k^{(m)}\right]$$

that is,

$$\overline{\boldsymbol{r}_k^{(m)}} = \frac{\boldsymbol{r}_k^{(0)} + \epsilon \sum_{i=1}^{\text{numnod}(k)} \bar{\boldsymbol{r}}_i^{(m-1,\,m)}}{1 + \epsilon \cdot \text{numnod}(k)}$$

where numnod(k) is the number of neighboring nodes of node k.

IV. Results and Discussion

A. Flow Around a Cylinder

First, inviscid incompressible flow around a cylinder with a uniform freestream velocity is calculated using the code developed. Figure 4 shows the Euler grid used in the calculation, which has 2958 nodes and 5716 cells. The computed surface pressure is compared with the corresponding exact solution. As shown in Fig. 5, the agreement is satisfactory. Fig. 6 shows the computed pressure contours where symmetrical distribution is obvious. The convergence history is illustrated in Fig. 7. With a CFL number of 0.8, more than four order of magnitude of residual reduction is achieved in 8000 iterations. Note that the β value for all of the cases considered in this work is equal to unit one.

Second, viscous laminar flow around a cylinder is also calculated with a much denser grid to capture the basic flow features near the surface of the cylinder. The Reynolds number for this case is

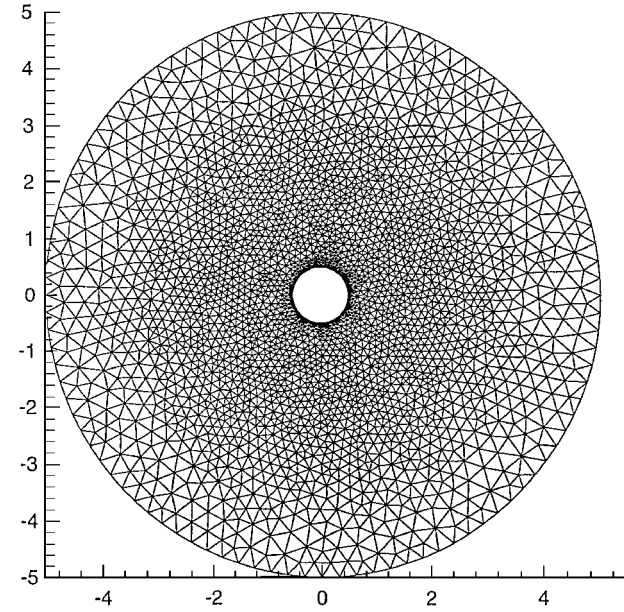


Fig. 4 Numerical grid around a cylinder for Euler computation.

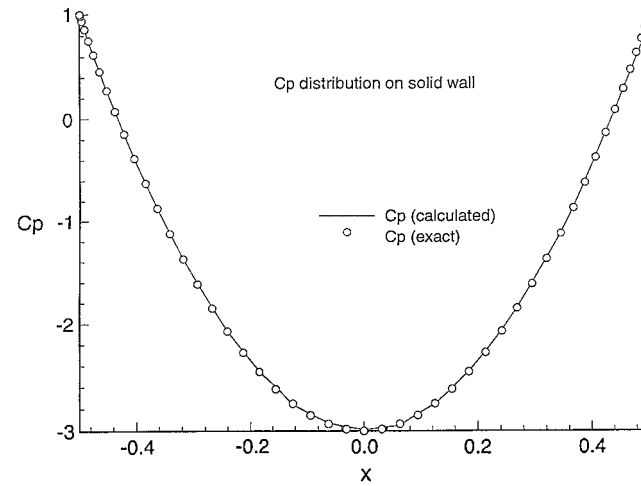


Fig. 5 Comparison of calculated and exact pressure distribution on cylinder wall (Euler solution).

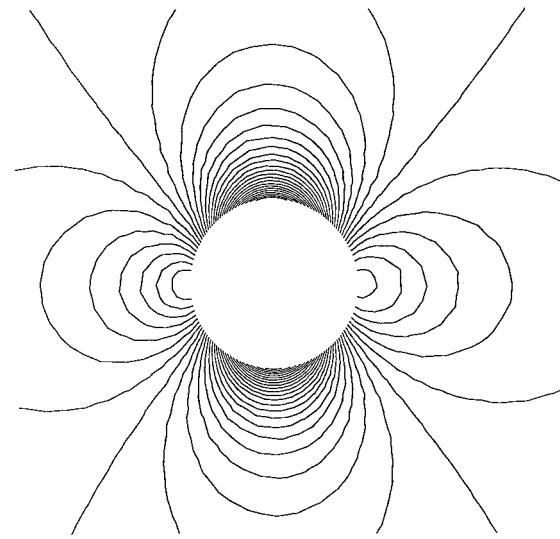


Fig. 6 Pressure contours (Euler solution).

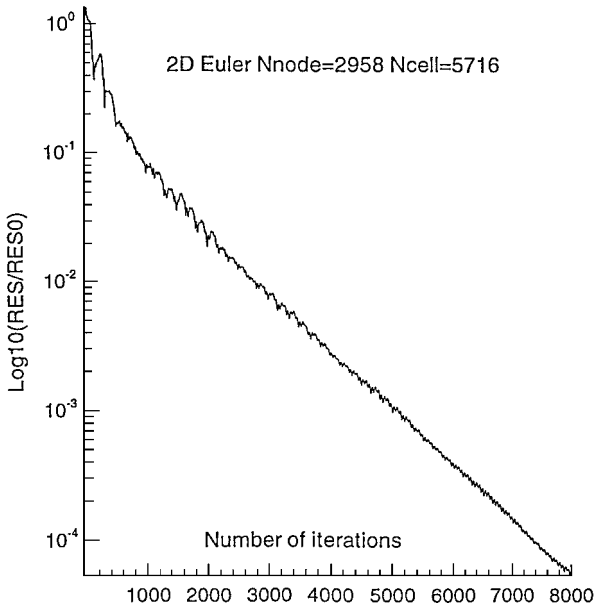


Fig. 7 Convergence history for inviscid flow around a cylinder.

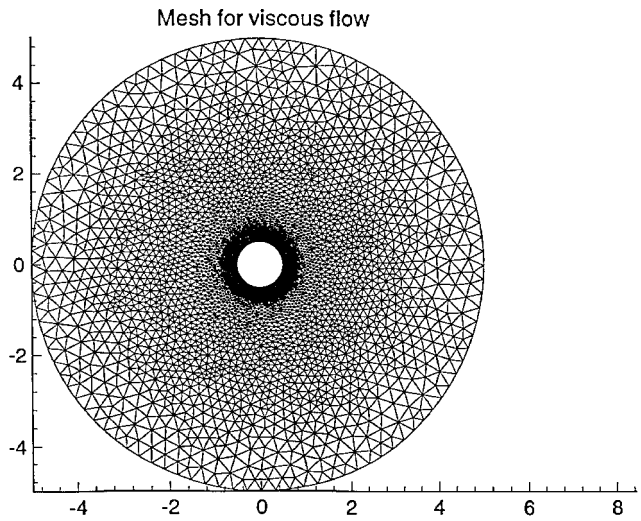


Fig. 8 Numerical grid around a cylinder for NS computation.

4×10^1 with the same flow conditions as in the study conducted by Fornberg.[15] The grid used is further refined using the grid stretching technique introduced earlier. As a result, the number of nodes and cells are increased to 12,699 and 25,198, respectively. The final grid is shown in Fig. 8. Again, the calculated pressure coefficient over the wall surface is compared with the corresponding result by Fornberg[15] in Fig. 9, which demonstrates a reasonable agreement. Velocity vectors in the vicinity of the wake region are plotted in Fig. 10a and compare favorably with results obtained by Connell and Braaten[16] using a compressible flow solver (Fig. 10b). Flow separation and the recirculation zone are well captured by the method. When the Reynolds number is increased to 4.1×10^1, the result obtained is then compared with a experimentally obtained visualization in Fig. 11 (see Ref. 17), which demonstrates a reasonable agreement in terms of separation point, shape, and size of the separation bubble behind the cylinder.

B. Flow Around a Sphere

Viscous incompressible flow around a sphere is calculated by the solver to validate the developed method for three-dimensional viscous flow simulation. The grid used has 240,503 nodes and 1,421,606 tetrahedral elements with mesh refinement in the near-wall region, as well as the rear of the sphere. Laminar flows at $Re = 5.65 \times 10^1$ and 1.18×10^2 are calculated, and they are found to be axisymmetric. Figure 12 shows a comparison of the

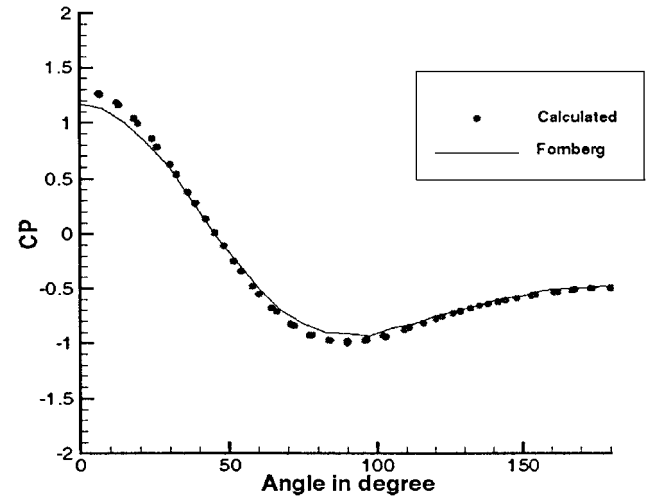


Fig. 9 Pressure distribution on cylinder wall (NS solution); dot represents pressure calculated by the present solver, and solid line denotes pressure by Fornberg.[15]

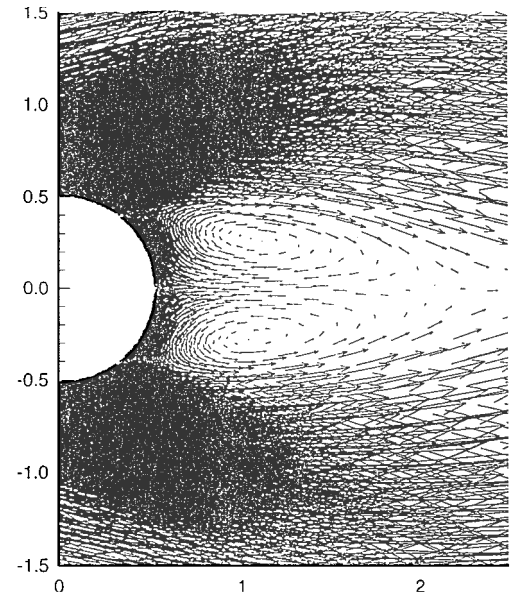


Fig. 10a Calculated velocity vectors.

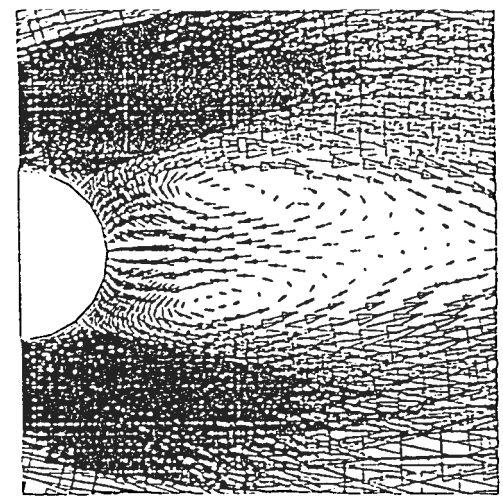


Fig. 10b Velocity vectors computed by Connell.[16]

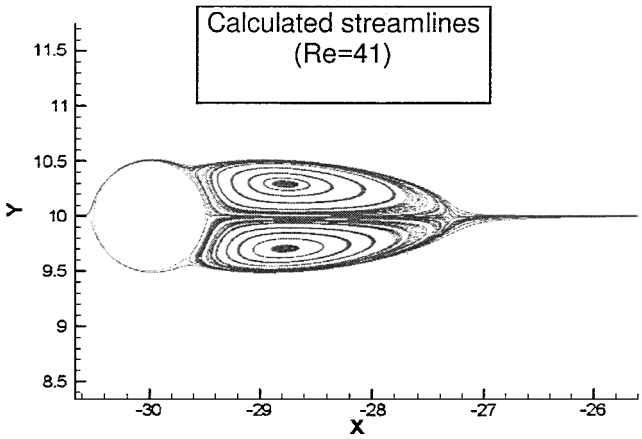


Fig. 11a Calculated streamline patterns behind cylinder.

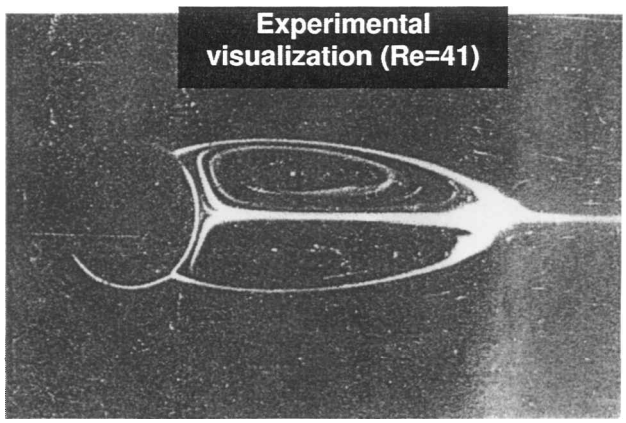
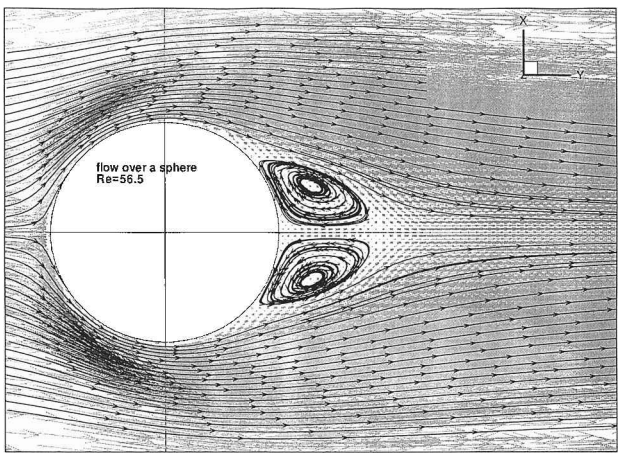


Fig. 11b Experimental visualization of the flow.[17]



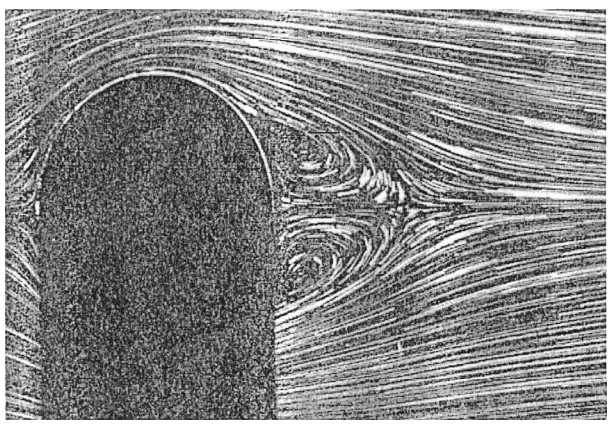
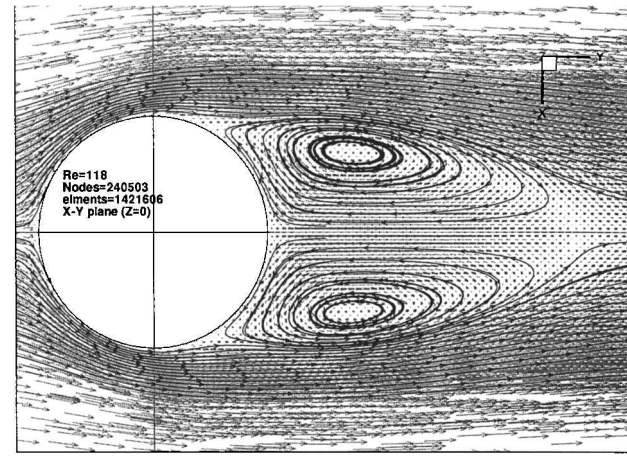


Fig. 12 Viscous flow over a sphere at $Re = 5.65 \times 10^2$: comparison of calculated streamlines with experimental visualization.[17]



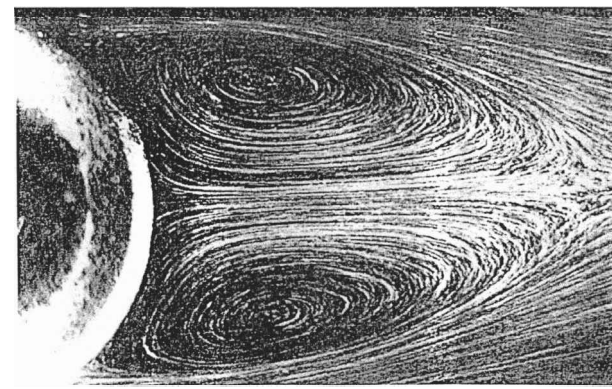


Fig. 13 Viscous flow over a sphere at $Re = 1.18 \times 10^2$: comparison of calculated streamlines with experimental visualization.[17]

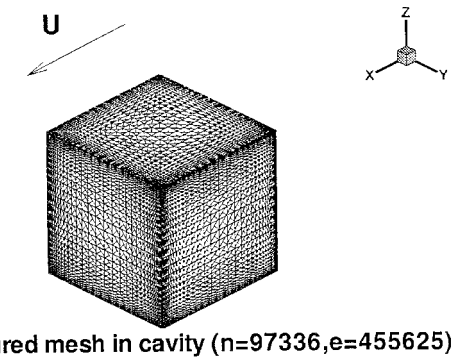


Fig. 14a Computational grid for the cavity.

calculated and measured streamline patterns in the x–y plane at $Re = 5.65 \times 10^1$, and Fig. 13 presents a similar comparison at $Re = 1.18 \times 10^2$. It is observed that the predicted streamline patterns for both cases have good agreement with the corresponding experimental visualizations.

C. Lid-Driven Flow in a Cavity

A lid-driven three-dimensional cavity flow with $Re = 4 \times 10^2$ is finally studied with the solver. Two schemes are used for comparison, one is the current third-order characteristics-basedscheme and the other is a second-order central diference scheme with fourth-order artificial viscosity similar to that in Ref. 8. In this study, the artificial viscosity coeffient also used is 0.0019. The computational results obtained are also compared with that of a least-squares finite element method (LSFEM) using structured grids.[18] A grid with 97,336 nodes and 455,625 tetrahedral elements is used with grid clustering near the five cavity walls, which is shown in Fig. 14a. A convergence history of the solution using the third-order characteristics-based scheme is shown in Fig. 14b, indicating four orders of residual

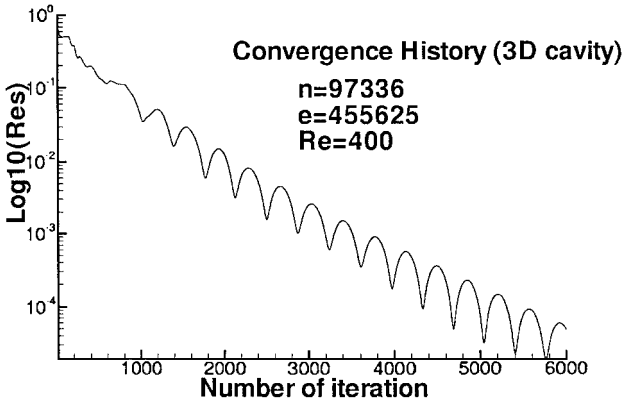


Fig. 14b Convergence history.

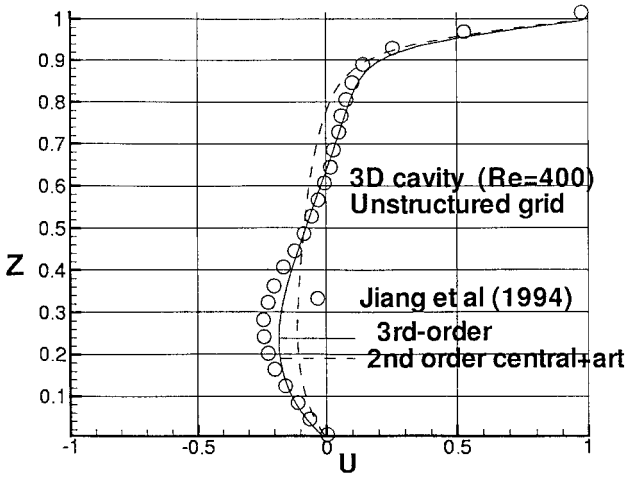


Fig. 14c Comparison of result of current third-order characteristics-based scheme with those of second-order central scheme and structured-grid least-square finite element solver.[18]

reduction in 6000 iterations. Note Fig. 14c, which shows a comparison of the two schemes as well as the comparison of unstructured-grid and structured-grid results for the same test case. A structured grid consisting of $50 \times 52 \times 50$ (130,000) elements and nodes was used in the LSFEM simulation. The u velocity profiles along the vertical centerline of the cavity, stretching from the bottom wall to the top of the cavity, are displayed. The velocity profile of the third-order characteristics-based scheme is found to agree well with that of the LSFEM given the lower number of nodes used than in the latter. The second-order scheme seems to smear the profile in regions of large gradients, thus indicating that even when a very small viscosity coefficient is used, the second-order central scheme with fourth-order artificial viscosity is more diffusive (less accurate) than the third-order scheme.

V. Conclusions

This paper presents an algorithm for numerical simulation of incompressible flows on three-dimensional unstructured grids. It is an upwind finite volume method based on the method of characteristics, which is made possible with the introduction of Chorin's[3] artificial compressibility formulation. Flow variables are calculated along characteristics, and their initial values are interpolated based on the signs of the corresponding characteristics. In addition, an upwind-biased interpolation method of third-order accuracy is used for interpolating flow variables on unstructured grids. With these inherent upwinding techniques for evaluating convection fluxes at control volume surfaces, no artificial viscosity is required. The discretized equations are solved by an explicit multistage Runge–Kutta time-stepping scheme, which is found to be efficient in terms of CPU and memory overheads. A computer code has been developed using the numerical methods presented. A number of test cases, including two-dimensional inviscid/viscous flows and three-dimensional inviscid/viscous flows, have been calculated to validate the code and to evaluate the performance of the numerical algorithm. Numerical results obtained are in good agreement with exact solutions and other published experimental and/or numerical results. The convergence rate of numerical simulation is generally found to be satisfactory, and the third-order characteristics-based scheme is found to be more accuarate than the second-order central scheme.

References

[1]Chen, A. J., and Kallinderis, Y., "Adaptive Hybrid (Prismatic-Tetrahedral) Grids for Incompressible Flows," *International Journal for Numerical Methods in Fluids*, Vol. 26, No. 9, 1998, pp. 1085–1105.

[2]Patankar, S. V., "Numerical Heat Transfer and Fluid Flow," Hemisphere, New York, 1980.

[3]Chorin, A., "A Numerical Method for Solving Incompressible Viscous Flow Problems," *Journal of Computational Physics*, Vol. 2, No. 1, 1967, pp. 12–26.

[4]Kwak, D., Chang, J. L. C., Shanks, S. P., and Chakravarthy, S. R., "A Three-Dimesional Incompressible Navier–Stokes Flow Solver Using Primitive Variables, *AIAA Journal*, Vol. 24, No. 3, 1986, pp. 390–396.

[5]Rogers, S. E., and Kwak, D., "Steady and Unsteady Solutions of the Incompressible Navier–Stokes Equations," *AIAA Journal*, Vol. 29, No. 4, 1991, pp. 603–610.

[6]Merckle, C., and Athavale, M. A., "Time-Accurate Unsteady Incompressible Flow Algorithms Based on Artificial Compressibility," AIAA Paper 87-1137, 1987.

[7]Rizzi, A., and Erikson, L., "Computation of Inviscid Incompressible Flow with Rotation," *Journal of Fluid Mechanics*, Vol. 153, April 1985, pp. 275–312.

[8]Farmer, J., Martinelli, L., and Jameson, A., "Fast Multigrid Method for Solving Incompressible Hydrodynamic Problems with Free Surface," *AIAA Journal*, Vol. 32, No. 6, 1994, pp. 1175–1182.

[9]Dick, E., and Linden, J., "A Multigrid Method for Steady Incompressible Navier–Stokes Equations Based on Flux Difference Splitting," *International Journal for Numerical Methods in Fluids*, Vol. 14, No. 11, 1992, pp. 1311–1323.

[10]Anderson, W. K., Rausch, R. D., and Bonhaus, D. L., "Implicit/Multigrid Algorithms for Incompressible Turbulent Flows on Unstructured Grids," *Journal of Computational Physics*, Vol. 128, No. 2, 1996, pp. 391–408.

[11]Eberle, A., "Three-Dimensional Euler Calculations Using Characteristic Flux Extrapolation," AIAA Paper, 85-0119, 1985.

[12]Drikakis, D., Govatsos, P. A., and Papantionis, D. E., "A Characteristic-Based Method for Incompressible Flows," *International Journal for Numerical Methods in Fluids*, Vol. 19, No. 8, 1994, pp. 667–685.

[13]Zhao, Y., and Winterbone, D. E., "The Finite Volume FLIC Method and Its Stability Analysis," *International Journal of Mechanical Sciences*, Vol. 37, No. 11, 1995, pp. 1147–1160.

[14]Barth, T. J., "Aspects of Unstructured Grids and Finite-Volume Solvers for Euler and Navier–Stokes Equations," AGARD–Von Kármán Inst. Lecture Series, R 787, AGARD, 1992.

[15]Fornberg, B., "A Numerical Study of Steady Viscous Flow Past a Circular Cylinder," *Journal of Fluids Mechanics*, Vol. 98, 1980, pp. 151–182.

[16]Connell, S. D., and Braaten, M. E., "Semistructured Mesh Generation for Three-Dimensional Navier–Stokes Calculations," *AIAA Journal*, Vol. 33, No. 6, 1995, pp. 1017–1024.

[17]Van Dyke, M., *An Album of Fluid Motion*, Parabolic, Stanford, CA, 1982, p. 118.

[18]Jiang, B. N., Lin, T. L., and Povinelli, L. A., "Large-Scale Computation of Incompressible Viscous Flow by Least-Squares Finite Element Method," *Computer Methods in Applied Mechanics and Engineering*, Vol. 114, No. 109, 1994.

J. Kallinderis
Associate Editor