#3927



# A Three-Dimensional Incompressible Navier-Stokes Flow Solver Using Primitive Variables

Dochan Kwak*
*NASA Ames Research Center, Moffett Field, California*
James L. C. Chang† and Samuel P. Shanks†
*Rockwell International, Canoga Park, California*
and
Sukumar R. Chakravarthy‡
*Stanford University, Stanford, California*

**An implicit, finite difference computer code has been developed to solve the incompressible Navier-Stokes equations in a three-dimensional, curvilinear coordinate system. The pressure field solution is based on the pseudocompressibility approach in which a time derivative pressure term is introduced into the mass conservation equation. The solution procedure employs an implicit, approximate factorization scheme. The Reynolds stresses, which are uncoupled from the implicit scheme, are lagged by one time step to facilitate implementing various levels of the turbulence model. Test problems for external and internal flows are computed and the results are compared with existing experimental data. The application of this technique for general three-dimensional problems is then demonstrated.**

## I. Introduction

THE development of new solution methodologies is one of the primary pacing items in computational fluid dynamics today.[1] With the current rate of progress in this discipline, as well as with grid generation techniques and the enhancements in computer capability, it is now practical to simulate complicated fluid dynamic phenomena associated with realistic geometries. To date, large computing times and memory requirements have been major difficulties in producing successful results from full Navier-Stokes codes. Even though the economic aspect should still be a primary concern in developing a fully three-dimensional production code, the time has come to utilize the available algorithms and high-speed computers to develop a useful tool for analysts and designers. The present paper presents the development of a three-dimensional, incompressible Navier-Stokes solver cast in generalized curvilinear coordinates using primitive variables.

Incompressible flow phenomena are frequently encountered in many engineering applications, especially in hydrodynamics and certain classes of aerodynamic problems such as dynamic stall and low-speed wind tunnel test problems. For most two-dimensional flow simulations, computer time and memory requirements are not the major limiting factors and various numerical techniques have been implemented quite successfully. For example, a stream-function vorticity formulation is frequently used for solving two-dimensional, viscous, incompressible flow problems (e.g., see Refs. 2-5). The three-dimensional extension of this method is not straightforward. Various three-dimensional Navier-Stokes codes have been developed, mainly for compressible flow. A few examples follow: Shang et al.[6] utilized MacCormack's explicit scheme, Hung and Kordulla[7] developed a code based on MacCormack's implicit scheme, Pulliam and Steger[8] implemented the Beam-Warming algorithm for a fully implicit code, and Briley and McDonald[9] independently developed a similar alternating direction implicit (ADI) scheme. Since the speed of sound approaches infinity at incompressible limit, implementing these codes for simulating incompressible flows is not computationally efficient. Therefore, in the present work, an efficient three-dimensional Navier-Stokes solver using primitive variables is developed for incompressible flow problems.

One of the major problems to be addressed in solving incompressible flows that use primitive variables is making the decision about which pressure solution method should be used to guarantee a divergence-free velocity field. The method of solving Poisson's equation for pressure was developed by Harlow and Welch[10] and has been used frequently for obtaining the pressure field, mostly using explicit methods. The usual computational procedure is to choose the pressure field such that the continuity is satisfied at the next time level, so that the new flowfield will be divergence free. This procedure normally requires a relaxation scheme iterating on pressure until the divergence-free condition is reasonably satisfied. This approach can be very time consuming and, thus, the computing time required for simulating three-dimensional flows has been prohibitively large. To accelerate the pressure field solution and alleviate the drawback associated with the Poisson's equation approach, Chorin[11] proposed to use artificial compressibility in solving the continuity equation. A similar method was adopted by Steger and Kutler[12] using an implicit approximate factorization scheme by Beam and Warming.[13] To implement the implicit time differencing, they fabricated a hyperbolic time-dependent system of equations by adding a time derivative of the pressure term to the mass conservation equation. These hyperbolic equations possess characteristics that are not present in the usual Poisson's equation for pressure (see Ref. 14 for a comprehensive analysis). This approach has been applied to simulate laminar, incompressible flow within liquid-filled shells.[15] Presently, this procedure has been extended to a three-dimensional flow solver cast in generalized curvilinear coordinates.

In Sec. II of this paper, the governing equations and turbulence model are presented. The finite difference algorithm is described in Sec. III and in Sec. IV results are presented

Presented as Paper 84-0253 at the AIAA 22nd Aerospace Sciences Meeting, Reno, NV, Jan. 9-12, 1984; received Feb. 17, 1984; revision received March 21, 1985. This paper is declared a work of the U.S. Government and therefore is in the public domain.

*Research Scientist. Member AIAA.

†Senior Member of Technical Staff, Rocketdyne Division.

‡Senior Research Associate (presently with Rockwell Science Center, Thousand Oaks, CA). Member AIAA.

for internal and external flow test problems that verify the accuracy of the code. Additional examples of a more practical nature are included to show the versatility of the current solver.

## II. Governing Equations and Turbulence Model

### Governing Equations

Unsteady, three-dimensional, viscous, incompressible flow with constant density is governed by the following Navier-Stokes equations, written in tensor notation and dimensionless form:

$$\frac{\partial u_i}{\partial x_i}=0 \tag{1a}$$

$$\frac{\partial u_i}{\partial t}+\frac{\partial u_i u_j}{\partial x_j}=-\frac{\partial p}{\partial x_i}+\frac{\partial \tau_{ij}}{\partial x_j} \tag{1b}$$

where $t$ is the time, $x_i$ the Cartesian coordinates, $u_i$ the corresponding velocity components, $p$ the pressure, and $\tau_{ij}$ the viscous stress tensor. The viscous stress tensor can be written in the following form:

$$\tau_{ij}=2\nu S_{ij}-R_{ij} \tag{1c}$$

The strain rate tensor is defined by

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

where $R_{ij}$ is the Reynolds stresses and $\nu$ the coefficient of kinematic viscosity.

The system of equations given above is not as readily solvable by a fast ADI scheme as the unsteady compressible Navier-Stokes equations, which are hyperbolic. This difficulty can be alleviated by modifying the continuity equation as follows (see Refs. 11 and 12):

$$\frac{1}{\beta}\frac{\partial p}{\partial t}+\frac{\partial u_i}{\partial x_i}=\frac{1}{\beta}\frac{\partial p^*}{\partial t} \tag{1e}$$

The parameter $1/\beta$ is the pseudocompressibility. As the solution converges to a steady state, the pseudocompressibility effect approaches zero, yielding the incompressible form of the equations. The pressure iteration term on the right-hand side of Eq. (1e) is mainly for time-dependent problems. Here, $p^*$ is the value of $p$ at the previous iteration and the pressure must be iterated until the divergence-free velocity field is obtained within the desired numerical accuracy. In the present paper, however, the primary interest is in obtaining steady-state solutions efficiently.

### Turbulence Model

Various levels of modeling are available (see Ref. 16 for a review), most of which require considerable experimental inputs. In the present code, however, turbulence is simulated by an algebraic eddy viscosity model, using a constitutive equation involving a "mixing length" that is a measure of the turbulence length scale. A generalization of this approach is given by the following equation:

$$R_{ij}=\tfrac{1}{3}R_{kk}\delta_{ij}-2\nu_t S_{ij} \tag{2}$$

where $\nu_t$ is the turbulent eddy viscosity. By including the normal stress $R_{kk}$ in the pressure, $\nu$ in Eq. (1c) can be replaced by $(\nu+\nu_t)$. For the turbulent viscosity, the algebraic model of Baldwin and Lomax[17] is implemented in the present study.

### Lower Bound of $\beta$

In solving the incompressible Navier-Stokes equations, exact mass conservation is of crucial importance in order to obtain a stable solution. In the present study, the continuity equation is modified to a hyperbolic type, thus introducing pressure waves of finite speed; the wave speed is infinity for truly incompressible flow.

Equation (1e) shows that $\partial u_i/\partial x_i$ approaches zero when $\beta \gg 1$. However, since magnitude of $\beta$ controls the speed of the pressure wave, it plays a very important role in determining convergence speed, accuracy, and stability. By simplifying the governing equation to a one-dimensional linearized form, propagation characteristics of the pressure wave have been studied.[14]

In the present paper, the lower bound of $\beta$ for laminar flow is set by using the following criterion recommended in Ref. 14:

$$\beta \gg \left[1+\frac{4}{Re}\left(\frac{x_{\text{ref}}}{x_\delta}\right)^2\left(\frac{x_L}{x_{\text{ref}}}\right)\right]^2-1 \tag{3a}$$

where $Re$ is the Reynolds number and $x_\delta$ and $x_L$ are the characteristic lengths that the vorticity and the pressure waves have to propagate during a given time span. For a duct flow simulation, $x_L$ is equal to the total length of the duct and $x_\delta$ is half the distance between the two walls of the duct. Similarly, for turbulent flows, the following criterion is used:

$$\beta \gg \left[1+\frac{1}{Re_t}\left(\frac{x_{\text{ref}}}{x_\delta}\right)\left(\frac{x_L}{x_{\text{ref}}}\right)\right]^2-1 \tag{3b}$$

where $Re_t$ is the Reynolds number based on the turbulent eddy viscosity. The upper bound of $\beta$ will be discussed in a later section.

## III. Numerical Algorithm

### Difference Equations

To accommodate fully three-dimensional geometries, the following generalized independent variables are introduced to transform the physical coordinates into general curvilinear coordinates:

$$\tau=t, \quad \xi_i=\xi_i(x_i,t) \tag{4}$$

where $\xi_i=\xi$, $\eta$, or $\zeta$ for $i=1$, 2, or 3, respectively.

Applying this transformation to the governing equations (1b) and (1e) and combining trapezoidal rule time differencing and the difference form of the transformed governing equations, the following equation in delta form is obtained:

$$\begin{aligned}&\{I+(h/2)J^{n+1}[\delta_\xi(\hat{A}_1^n-\Gamma_1)+\delta_\eta(\hat{A}_2^n-\Gamma_2)\\&+\delta_\zeta(\hat{A}_3^n-\Gamma_3)]\}(\hat{Q}^{n+1}-\hat{Q}^n)=(I-I_m)(p^n-p^{n-1})\\&-[I-(J^{n+1}/J^n)]\hat{Q}^n-hJ^{n+1}(\delta_\xi\hat{E}_1^n+\delta_\eta\hat{E}_2^n+\delta_\zeta\hat{E}_3^n)\\&+hJ^{n+1}(\delta_\xi\Gamma_1+\delta_\eta\Gamma_2+\delta_\zeta\Gamma_3)\hat{Q}^n\end{aligned} \tag{5}$$

where

$J=$ Jacobian of the transformation

$$\hat{Q}=\begin{bmatrix}p\\u\\v\\w\end{bmatrix}, \quad I_m=\begin{bmatrix}0&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$$

$$\hat{E}_i^n = \frac{1}{J}\begin{bmatrix} \beta U_i + L_0(p-\beta) \\ uU_i + L_1 p \\ vU_i + L_2 p \\ wU_i + L_3 p \end{bmatrix}$$

$$\hat{A}_i^n = \frac{1}{J}\begin{bmatrix} L_0 & (L_1\beta) & (L_2\beta) & (L_3\beta) \\ L_1 & (U_i + L_1 u) & L_2 u & L_3 u \\ L_2 & L_1 v & (U_i + L_2 v) & L_3 v \\ L_3 & L_1 w & L_2 w & (U_i + L_3 w) \end{bmatrix} \tag{6}$$

Here, the contravariant velocities, $U_i$ without metric normalization, are defined as

$$U_i = L_0 + L_1 u + L_2 v + L_3 w \tag{7}$$

where $L_0 = (\xi_i)_t$, $L_i = (\xi_i)_x$, $L_2 = (\xi_i)_y$, $L_3 = (\xi_i)_z$; $\delta_\xi$ is the finite difference form of $\partial/\partial\xi$, etc; and $h = \Delta\tau$ = time step. The superscript $n$ denotes the $n$th time step and the viscous terms are given as

$$\Gamma_i = \frac{\nu}{J}\left(\nabla\xi_i \cdot \nabla\xi_j I_m \frac{\partial}{\partial\xi_j}\right) \tag{8}$$

### Approximate Factorization

The full viscous terms in Eq. (8) produce nontridiagonal elements in the left-hand side of Eq. (5). Therefore, to implement an approximate factorization scheme,[9,13] only orthogonal terms are kept on the left-hand side. For steady-state solutions, this can be done since the left-hand side approaches zero as a steady state is approached. For a time-accurate solution, this approximation procedure needs to be further investigated when a nonorthogonal grid is used. For the right-hand side (RHS), the full viscous terms may be included. Presently, a nearly orthogonal grid is used and the viscous terms are further simplified to

$$\Gamma_i = \frac{\nu}{J}(\nabla\xi_i \cdot \nabla\xi_i) I_m \frac{\partial}{\partial\xi_i} = \gamma_i I_m \frac{\partial}{\partial\xi_i} \tag{9}$$

After adding smoothing terms to stabilize the computation, the approximate factored form of the governing equation becomes

$$\begin{aligned} &[I + (h/2)J^{n+1}\delta_\xi(\hat{A}_1^n - \gamma_1 I_m \delta_\xi) - \epsilon_i \nabla_\xi \Delta_\xi] \\ &\cdot [I + (h/2)J^{n+1}\delta_\eta(\hat{A}_2^n - \gamma_2 I_m \delta_\eta) - \epsilon_i \nabla_\eta \Delta_\eta] \\ &\cdot [I + (h/2)J^{n+1}\delta_\zeta(\hat{A}_3^n - \gamma_3 I_m \delta_\zeta) - \epsilon_i \nabla_\zeta \Delta_\zeta](\hat{Q}^{n+1} - \hat{Q}^n) \\ &= [\mathrm{RHS}(8)] - \epsilon_e[(\nabla_\xi\Delta_\xi)^2 + (\nabla_\eta\Delta_\eta)^2 + (\nabla_\zeta\Delta_\zeta)^2]\hat{Q}^n \end{aligned} \tag{10}$$

where $\epsilon_i$ and $\epsilon_e$ are implicit and explicit smoothing terms, and

$$\nabla_\xi\hat{Q} = \hat{Q}_j - \hat{Q}_{j-1}, \qquad \Delta_\eta\hat{Q} = \hat{Q}_{k+1} - \hat{Q}_k$$

$$\delta_\zeta\hat{Q} = (\hat{Q}_{l+1} - \hat{Q}_{l-1})/(2\Delta\zeta)$$

$$\begin{aligned} \delta_\eta\gamma\delta_\eta\hat{Q} = [&(\gamma_{k+1} + \gamma_k)(\hat{Q}_{k+1} - \hat{Q}_k) \\ &- (\gamma_k + \gamma_{k-1})(\hat{Q}_k - \hat{Q}_{k-1})]/[2(\Delta\eta)^2] \end{aligned}$$

Analogous terms in other directions are defined similarly.

### Upper Bound of $\beta$

The criteria given by Eqs. (3a) and (3b) fix the lower bound of $\beta$. However, a large value of $\beta$ makes the added pressure term in the continuity equation very small, which in turn causes the pseudocompressible formulation to become very stiff.[12] Therefore, the upper bound of $\beta$ depends on the particular numerical algorithm chosen. For example, if an explicit scheme is used, the stability condition associated with the pseudosound speed must be considered just as in the case of compressible flows.

The approximate factorization algorithm used in the present study has a different feature to be considered. In factoring the finite difference form of the governing equations to obtain Eq. (10), higher-order cross-differencing terms are added to the left-hand side of Eq. (5). These added terms contaminate the momentum equations as well as the continuity equation and, therefore, must be kept smaller than the original terms everywhere in the computational domain; that is,

$$\mathcal{O}\left[\left(\frac{h}{2}J\right)^2 \delta_\xi\hat{A}_1\delta_\eta\hat{A}_2\right] < \mathcal{O}\left[\frac{h}{2}J\delta_\xi\hat{A}_1\right] \tag{11}$$

This leads to the following criterion for the upper bound of $\beta$:

$$\beta h < \mathcal{O}(1) \tag{12}$$

If $\beta$ is chosen to be larger than the above range, the accuracy of the numerical solution will deteriorate significantly and can even induce numerical instability.

### Higher-Order Smoothing Terms

It has been found that a higher-order smoothing term is required to make the present algorithm stable (see Refs. 12 and 13). In this section, the smoothing terms used in the code are described in relation to the usual upwinding scheme as well as their behavior near computational boundaries.

For simplicity, the following equation is considered:

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0 \tag{13}$$

The usual upwind differencing of the convective term results in

$$\begin{aligned} c\delta_x u = c(3u_j - 4u_{j-1} + u_{j-2})/(2\Delta x), \qquad c>0 \\ c(-3u_j + 4u_{j+1} - u_{j+2})/(2\Delta x), \qquad c<0 \end{aligned} \tag{14a}$$

where $\delta_x$ is a difference form of $\partial u/\partial x$. This equation can be written, for all $c$, as

$$\begin{aligned} c\delta_x u = &(c/4\Delta x)[(3u_j - 4u_{j-1} + u_{j-2}) + (-3u_j + 4u_{j+1} - u_{j+2})] \\ &+ (|c|/4\Delta x)[(3u_j - 4u_j + u_{j-2}) - (-3u_j + 4u_{j+1} - u_{j+2})] \end{aligned}$$

Replacing the first [ ] term of the above by the usual central difference formula results in

$$\begin{aligned} c\delta_x u = &(c/2\Delta x)(u_{j+1} - u_{j-1}) + (|c|/4\Delta x) \\ &\times (u_{j-2} - 4u_{j-1} + 6u_j - 4u_{j+1} + u_{j+2}) \end{aligned} \tag{14b}$$

Therefore, a finite difference form of Eq. (13) with a fourth-order explicit smoothing term can be written, in delta form, as

$$\begin{aligned} \Delta u_j + \frac{c\Delta t}{2\Delta x}(\Delta u_{j+1} - \Delta u_{j-1}) = -\frac{c\Delta t}{2\Delta x}(u_{j+1} - u_{j-1})^n \\ - \frac{\epsilon_e |c| \Delta t}{4\Delta x}(u_{j-2} - 4u_{j-1} + 6u_j - 4u_{j+1} + u_{j+2})^n \\ 0 < \epsilon_e \le 1 \end{aligned} \tag{15}$$

where $\Delta u = u^{n+1} - u^n$ and $\epsilon_e$ is an explicit smoothing coefficient.

In a similar manner, the implicit second-order smoothing term can be obtained by starting from

$$c\delta_x u = \frac{c}{2\Delta x}(u_{j+1} - u_{j-1}) - \frac{|c|}{2\Delta x}(u_{j+1} - 2u_j + u_{j-1}) \quad (16a)$$

This can be put into a numerical scheme, in delta form, as

$$\Delta u_j + \frac{c\Delta t}{2\Delta x}(\Delta u_{j+1} - \Delta u_{j-1}) - \frac{\epsilon_i |c| \Delta t}{2\Delta x}(\Delta u_{j+1} - 2\Delta u_j + \Delta u_{j-1})$$
$$= -\frac{c\Delta t}{2\Delta x}(u_{j+1} - u_{j-1})^n - \frac{\epsilon_e |c| \Delta t}{4\Delta x}$$
$$\times (u_{j-2} - 4u_{j-1} + 6u_j - 4u_{j+1} + u_{j+2})^2 \quad (16b)$$

where $\epsilon_i$ is an implicit smoothing coefficient. It is interesting to observe how a small perturbation $u'$ propagates under various smoothing terms. Following the fluid, the second-order smoothing acting on the perturbed quantity is

$$\frac{\partial u'}{\partial t} - \epsilon \frac{\partial^2 u'}{\partial^2 x} = 0 \quad (17a)$$

then

$$u' = e^{-\epsilon\alpha^2 t}(ae^{i\alpha x} + be^{-i\alpha x})$$

For the fourth-order smoothing term,

$$\frac{\partial u'}{\partial t} + \epsilon \frac{\partial^4 u'}{\partial^4 x} = 0 \quad (17b)$$

then

$$u' = e^{-\epsilon\alpha^4 t}(ae^{i\alpha x} + be^{-i\alpha x})$$

On the boundary, the flow variables are often extrapolated to maintain the same order of differencing. However, a lower order of differencing could be used by backward or forward differencing. And it should be noted that the sign changes when a lower-order smoothing is incorporated. In the present work, the second-order smoothing behaves well on inflow and outflow boundaries, while the fourth-order extrapolation smoothing term is better on solid boundaries.

### Boundary Conditions

Once the numerical algorithm has been developed, the next most important aspect of solving a fluid dynamics problem is the proper implementation of the boundary conditions. There are several different types of boundaries encountered in numerical simulations: solid surface, far-field, inflow and outflow, and symmetric or reflective boundaries. All of these are required in the present code.

On a solid surface, the usual no-slip condition is applied. The pressure at the wall is obtained by setting the gradient of $p$ equal to zero at the no-slip wall, i.e.,

$$\frac{\partial p}{\partial n} = 0 \quad (18)$$

The boundary condition can be implemented either explicitly or implicitly. The latter enhances stability of the code.

## IV. Computed Results

The flow solver was verified by solving a few simple test problems. To test the code on an external flow simulation, flow over a circular cylinder was computed. To test this program for internal flows, a channel flow problem was computed. Various complicated three-dimensional flows were then computed; a couple of examples are described here to demonstrate the capability of the present code.

### Flow over a Circular Cylinder

Flow over a circular cylinder has been a rich source of various fluid dynamic phenomena (see Ref. 18). The flow over an impulsively started circular cylinder at a Reynolds number of 40 based on the diameter of the cylinder is chosen as a verification case. If the viscous region is taken to be approximately two diameters away from the cylinder, Eqs. (3a) and (12) lead to the following range for $\beta$ using $\Delta\tau = 0.1$:

$$0.1 \ll \beta < 10$$

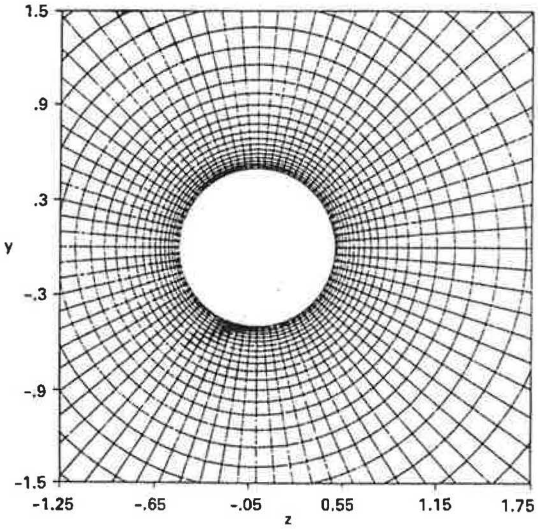


**Fig. 1 Grid for flow over a circular cylinder.**

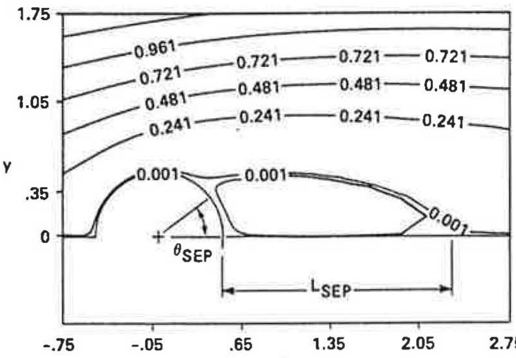


**Fig. 2 Steady-state stream function contours for flow over a circular cylinder at $Re = 40$.**

**Table 1 Comparison of steady-state flowfield (flow over a circular cylinder at $Re = 40$ based on diameter)**

| | $C_{dp}$ | $C_{pf}$ | $C_{pr}$ | $\theta_{sep}$, deg | $L_{sep}$ |
|---|---|---|---|---|---|
| Literature data summary | 0.93-1.05 | 1.14-1.23 | −0.47-(−0.55) | 50-53.9 | 1.8-2.5 |
| Present result | 1.03 | 1.15 | −0.51 | 52 | 1.9 |

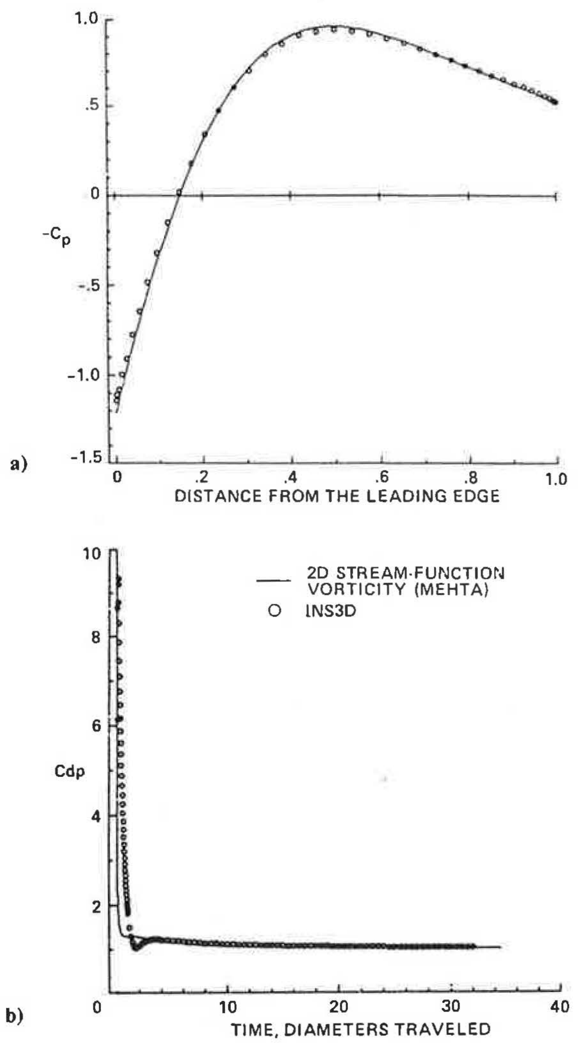


**Fig. 3 Flow over an impulsively started circular cylinder at $Re=40$: a) steady-state pressure coefficient on the surface; b) pressure drag history.**

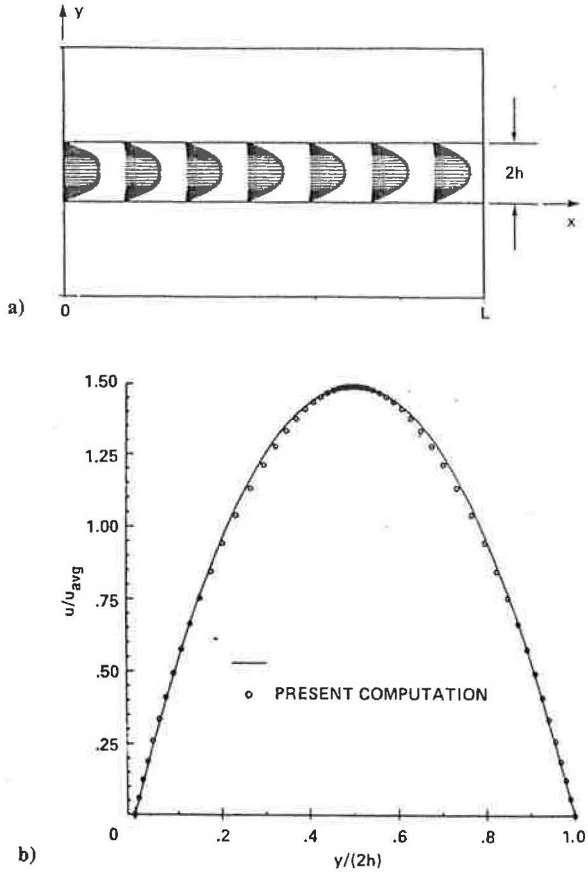


**Fig. 4 Developing laminar channel flow at $Re=1000$ ($Re$ based on channel width and average velocity): a) velocity vector; b) fully developed velocity profile.**

The grid system chosen is shown in Fig. 1 using $80 \times 41$ mesh points in circumferential and vertical directions of the cylinder, respectively. Figure 2 shows the steady-state stream function contours for the upper half space. The steady-state pressure coefficient on the cylinder surface is then compared in Fig. 3a with that of Mehta (private communication, NASA Ames), who used a stream function and vorticity formulation in two dimensions. Various experimental and computational studies on this benchmark case have been reported. The quantities that are presently used to verify the physical phenomena are the separated bubble length $L_{sep}$, the separation angle $\theta_{sep}$ (see Fig. 2), pressure drag $C_{dp}$, and the pressure coefficients at the forward and rear stagnation points $C_{pf}$ and $C_{pr}$, respectively. The flowfield computed by the present code, INS3D, compares quite well with those reported in the literature (see Table 1). For this computation, a nondimensional time step of 0.1 was used and the steady state was reached in 300 iterations. To simulate the impulsive start, the number of pressure iterations was increased to five at each time step; the pressure-drag history is compared in Fig. 3b with the time-accurate computation of Mehta. Even though the present algorithm is geared for a steady-state solution, this example demonstrates the potential for a time-accurate procedure.

### Channel Flow

Just as the circular cylinder problem is the simplest representation for external flows, channel flow is perhaps the simplest representation for internal flows. However, the channel flow problem still provides the essential features for testing the present algorithm. The coordinate system chosen for this problem is shown in Fig. 4a. To reduce the channel length for obtaining fully developed flow, a partially developed boundary-layer profile is used as an inflow condition. In Fig. 4b, a converged laminar velocity profile is compared with an analytic solution. A turbulent case is tested by increasing the Reynolds number to 100,000. The mean velocity defect is compared in Fig. 5 with Prandtl's universal law, Comte-Bellot's experiment for channel flow,[19] and Klebanoff's well-known boundary-layer profile.

The influence of the $\beta$ term on convergence rate and accuracy is shown in Fig. 6. As a measure of convergence, rms values of $\Delta Q$ (denoted by RMSDQ) are monitored. The accuracy of the solution can be tested by checking how accurately the divergence-free velocity field is attained. The first term of $(\hat{Q}^{n+1} - \hat{Q}^n)$ in Eq. (10) can be regarded as a parameter showing how well the continuity equation is satisfied. The rms values of this quantity are denoted by RMSCO. For $L=20$, $Re=1000$, and $\Delta\tau=0.1$, the recommended range for $\beta$ is estimated to be

$$0.75 \ll \beta < 10$$

For various values of $\beta$, RMSDQ and RMSCO are plotted in Figs. 6a and 6b, respectively. This test shows that, if $\beta$ is chosen outside this range, the accuracy of the solution becomes poor or the iteration is not converging as in the case of $\beta=0.1$.

### Three-Dimensional Flow Examples

The geometry of a rectangular duct with 45 deg bend, which is similar to part of the new low-speed wind tunnel at Ames Research Center, is shown in Fig. 7a. The laminar flow solution at $Re = 1000$ is shown in the form of velocity vectors in Figs. 7b and 8. This qualitative simulation shows the nature of the secondary flow and the separated region. Substantial variation in mean velocity vectors can be observed from the top or bottom wall to the center of the duct due to the secondary flow.

One of the prime objectives in developing the present code is to analyze and verify the flowfield in the Space Shuttle main engine (SSME) power head. The example shown in Fig. 9 exhibits typical flow characteristics locally encountered in the SSME simulation.

Figure 9a shows the geometry of an annular duct with a 180 deg bend. This configuration represents the turnaround duct in the hot-gas manifold of the Shuttle engine. The laminar flow ($Re = 1000$) solution in Fig. 9b shows a considerable nonuniformity after the 180 deg bend. In this case, an adverse pressure gradient is developed after the bend and subsequently a large area of separation is formed along the inner wall. Further numerical simulation of the SSME will be used to redesign and to optimize the hot-gas manifold. For this computation, a $50 \times 21 \times 16$ mesh was used in flow, radial, and circumferential directions, respectively, with a

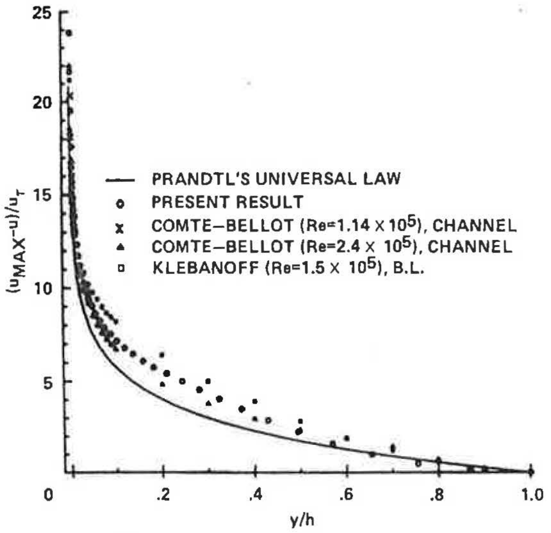


Fig. 5 Mean velocity defect comparison for turbulent channel flow at $Re = 10^5$.

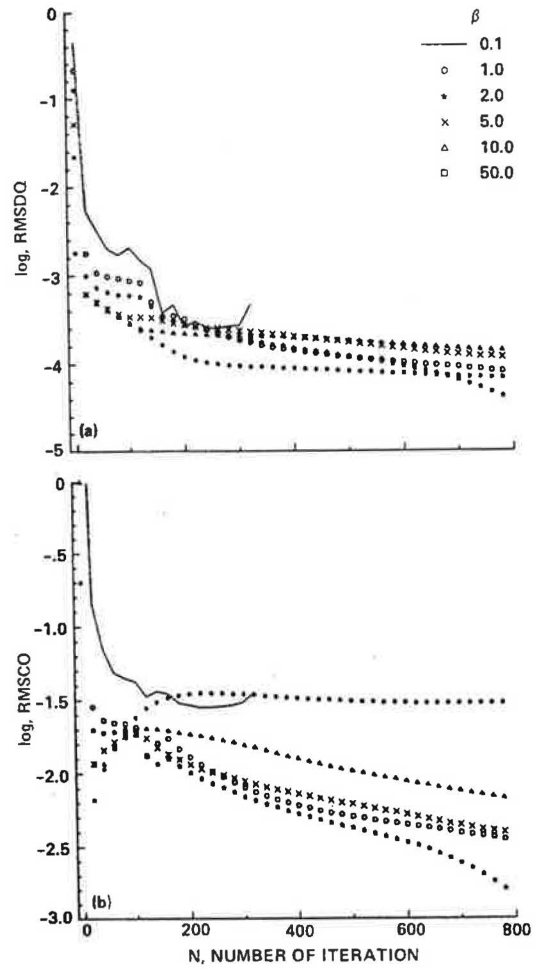


Fig. 6 Influence of $\beta$ on convergence and accuracy for channel flow with $L = 20$ at $Re = 1000$.

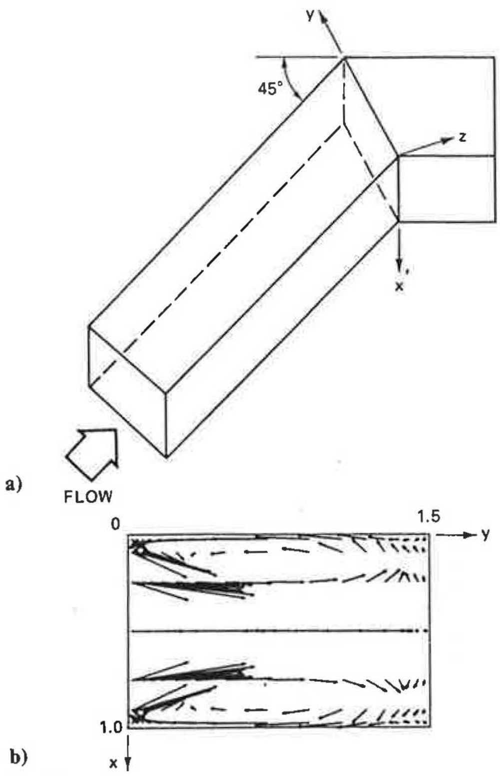


Fig. 7 Flow through a rectangular duct with a 45 deg bend at $Re = 1000$: a) duct geometry; b) secondary flow at $z = 0$ plane (view from downstream).

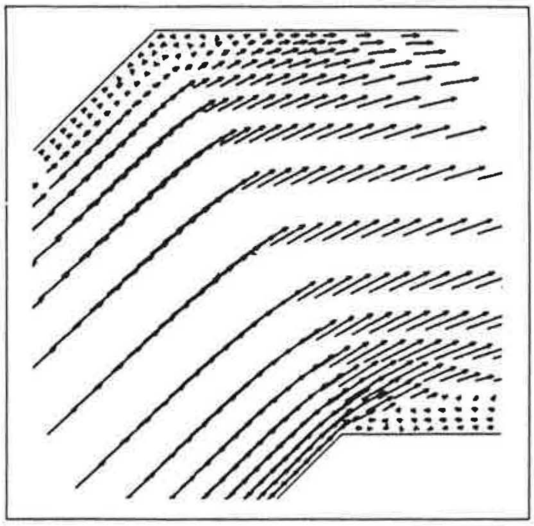

Fig. 8 Flow through a rectangular duct with a 45 deg bend at $Re = 1000$ (velocity vector at $x = 0.5$ plane).

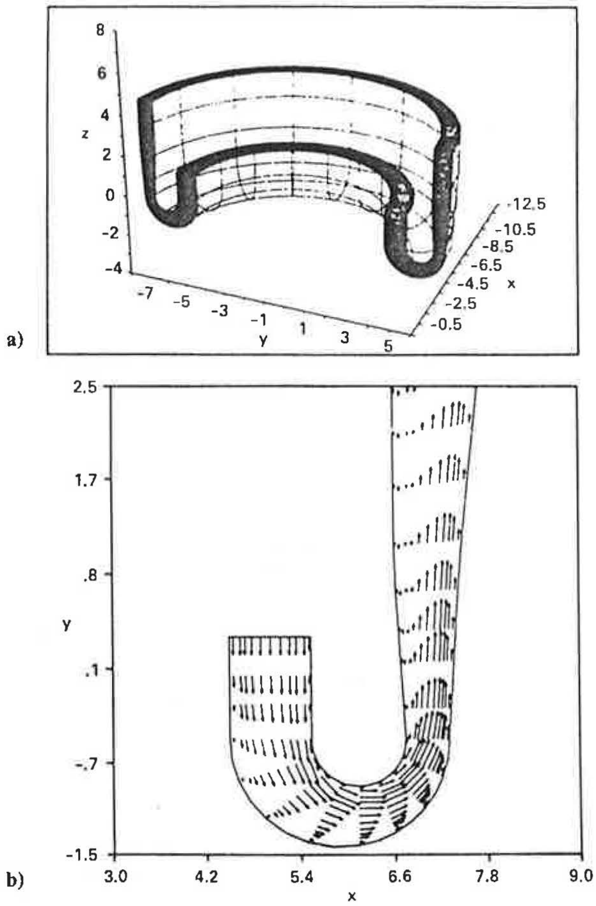


**Fig. 9 Flow through a turnaround duct: a) three-dimensional grid; b) typical flow pattern with separation.**

nondimensional time step of 0.1. A converged solution was obtained after 300 time steps, starting from a uniform velocity field. The computing time for each time step was 1.0E-04 s per mesh point on the Cray X-MP computer at Ames. Because of the axisymmetric nature of the present configuration, only 16 mesh points were used in the circumferential direction. However, for nonsymmetric flows, more mesh points and proper local clustering of grid lines will be required. The three-dimensional computation of the hot-gas manifold of the SSME has been performed using the present code. Details of this numerical simulation, as well as comparison with experimental results, have been presented in Ref. 20.

## V. Conclusions

This paper presents the development of an efficient and robust computer code for incompressible Navier-Stokes flows (INS3D). In this work, the basic formulation and the algorithm are described. INS3D has been applied to various geometrically complex flows. Other aspects, such as multiple-zone computations, implementation of smoothing terms near intersections of solid walls, symmetry boundary conditions, and analysis of differencing errors, will be presented in a future paper. The algebraic turbulence model implemented in the present version of the code needs more development for solving the flow with massively separated regions, as in the case of the SSME power head simulation. Higher-level turbulence models are being investigated in an effort to achieve better estimates of the turbulence length scales.

## References

[1]Kutler, P., "A Perspective of Theoretical and Applied Computational Fluid Dynamics," AIAA Paper 83-0037, 1983.

[2]Mehta, U. B., "Dynamic Stall of an Oscillating Airfoil," AGARD CP 227, Sept. 1977, Paper 23.

[3]Lecointe, Y. and Piquet, J., "On the Use of Several Compact Methods for the Study of Unsteady Incompressible Viscous Flow for Outer Problems: II," *Eighth International Conference on Numerical Methods in Fluid Dynamics*, Springer-Verlag, Aachen, FRG, June 1982, pp. 323-328.

[4]Bramley, J. S. and Dennis, S. C. R., "A Numerical Treatment of Two-Dimensional Flow in a Branching Channel," *Eighth International Conference on Numerical Methods in Fluid Dynamics*, Springer-Verlag, Aachen, FRG, June 1982, pp. 155-160.

[5]Rosenfeld, M. and Wolfstein, M., "Numerical Solution of Viscous Flow around Arbitrary Airfoils in a Straight Cascade," *Eighth International Conference on Numerical Methods in Fluid Dynamics*, Springer-Verlag, Aachen, FRG, June 1982, pp. 433-439.

[6]Shang, J. S., Buning, P. G., Hankey, W. L., and Wirth, M. C., "Performance of a Vectorized Three-Dimensional Navier-Stokes Code on the Cray-1 Computer," *AIAA Journal*, Vol. 18, Sept. 1980, pp. 1073-1079.

[7]Hung, C. M. and Kordulla, W., "A Time-Split Finite-Volume Algorithm for Three-Dimensional Flow-Field Simulation," AIAA Paper 83-1957, 1983.

[8]Pulliam, T. H. and Steger, J. L., "Implicit Finite-Difference Simulations of Three-Dimensional Compressible Flow," *AIAA Journal*, Vol. 18, Feb. 1980, pp. 159-167.

[9]Briley, W. R. and McDonald, H., "Solution of the Three-Dimensional Compressible Navier-Stokes Equations by an Implicit Technique," *Proceedings of Fourth International Conference on Numerical Method in Fluid Dynamics, Lecture Notes in Physics*, Vol. 35, Springer-Verlag, New York, 1975, pp. 105-110.

[10]Harlow, F. H. and Welch, J. E., "Numerical Calculation of Time-Dependent Viscous Incompressible Flow with Free Surface," *Physics of Fluids*, Vol. 8, Dec. 1965, pp. 2182-2189.

[11]Chorin, A. J., "A Numerical Method for Solving Incompressible Viscous Flow Problems," *Journal of Computational Physics*, Vol. 2, 1967, pp. 12-26.

[12]Steger, J. L. and Kutler, P., "Implicit Finite-Difference Procedures for the Computation of Vortex Wakes," *AIAA Journal*, Vol. 15, April 1977, pp. 581-590.

[13]Beam, R. M. and Warming, R. F., "An Implicit Finite-Difference Algorithm for Hyperbolic Systems in Conservation-Law Form," *Journal of Computational Physics*, Vol. 22, Sept. 1976, pp. 87-110.

[14]Chang, J. L. C. and Kwak, D., "On the Method of Pseudo Compressibility for Numerically Solving Incompressible Flows," AIAA Paper 84-252, Jan. 1984.

[15]Chakravarthy, S. R., "Numerical Simulation of Laminar Incompressible Flow within Liquid Filled Shells," U. S. Army Ballistics Research Laboratory, Aberdeen Proving Ground, MD, Rept. ARBRL-CR-00491, Nov. 1982.

[16]Reynolds, W. C., "Computation of Turbulent Flows," *Annual Review of Fluid Mechanics*, Vol. 8, 1976, pp. 183-208.

[17]Baldwin, B. S. and Lomax, H., "Thin Layer Approximation and Algebraic Model for Separated Turbulent Flows," AIAA Paper 78-257, Jan. 1978.

[18]Morkovin, M. V., "Flow around Circular Cylinder—A Kaleidoscope of Challenging Fluid Phenomena," *Symposium on Fully Separated Flows*, edited by A. G. Hansen, ASME, New York, 1964, pp. 102-118.

[19]Comte-Bellow, G., "Contribution a l'Etude de la Turbulence de Conduite," Doctoral Thesis, University of Grenoble, Grenoble, France, 1963.

[20]Chang, J. L. C., Kwak, D., Dao, S. C., and Rosen, R., "A Three-Dimensional Incompressible Flow Simulation Method and Its Application to the Space Shuttle Main Engine, Part I—Laminar Flow," AIAA Paper 83-0175, Jan 1985.