AIAA'84

AIAA-84-0253

An Incompressible Navier-Stokes Flow Solver in Three-Dimensional Curvilinear Coordinate System Using Primitive Variables

D. Kwak, NASA Ames Research Center, Moffett Field, CA; and J.L.C. Chang and S.P. Shanks, Rockwell International, Canoga Park, CA; and S.R. Chakravarthy, Stanford Univ., Stanford, CA

AIAA 22nd Aerospace Sciences Meeting

January 9-12, 1984/Reno, Nevada

AN INCOMPRESSIBLE NAVIER-STOKES FLOW SOLVER IN THREE-DIMENSIONAL CURVILINEAR COORDINATE SYSTEMS USING PRIMITIVE VARIABLES

Dochan Kwak*
NASA Ames Research Center, Moffett Field, California

James L.C. Chang† and Samuel P. Shanks†
Rocketdyne Division, Rockwell International, Canoga Park, California

Sukumar R. Chakravarthy‡
Stanford University, Stanford, California

Abstract

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 pseudo compressibility approach in which the time derivative pressure term is introduced into the mass conservation equation to form a set of hyperbolic equations. The solution procedure employs an implicit, approximate factorization scheme. The Reynolds stresses, that 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 with 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 in 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 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 (for example, 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 ADI scheme. Implementing these codes for simulating incompressible flows is not efficient and is generally not recommended. 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 continuity is satisfied at the next time-level, so that the new flow field will be divergence-free. This procedure normally requires a relaxation scheme iterating on pressure until the divergence-free

*Research Scientist.
†Senior Member of Technical Staff.
‡Senior Research Associate. Member AIAA. Present address, Rockwell Science Center, Thousand Oaks, California.

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 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, incompressible flow with constant density is governed by the following Navier-Stokes equations, written in Cartesian coordinates:

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}=0 \tag{1a}$$

$$\frac{\partial \hat{q}}{\partial t}+\frac{\partial(\hat{e}-\hat{e}_v)}{\partial x}+\frac{\partial(\hat{f}-\hat{f}_v)}{\partial y}+\frac{\partial(\hat{g}-\hat{g}_v)}{\partial z}=0 \tag{1b}$$

where

$$\hat{q}=\begin{bmatrix}u\\v\\w\end{bmatrix},$$

$$\hat{e}=\begin{bmatrix}u^2+p\\uv\\uw\end{bmatrix},\quad \hat{f}=\begin{bmatrix}vu\\v^2+p\\vw\end{bmatrix},\quad \hat{g}=\begin{bmatrix}wu\\wv\\w^2+p\end{bmatrix}, \tag{1c}$$

$$\hat{e}_v=\begin{bmatrix}\tau_{xx}\\\tau_{xy}\\\tau_{xz}\end{bmatrix},\quad \hat{f}_v=\begin{bmatrix}\tau_{yx}\\\tau_{yy}\\\tau_{yz}\end{bmatrix},\quad \hat{g}_v=\begin{bmatrix}\tau_{zx}\\\tau_{zy}\\\tau_{zz}\end{bmatrix}$$

To implement an implicit, approximate factorization scheme[13] to the above set of equations, the continuity equation is modified according to the procedures of Refs. 11 and 12 as follows:

$$\frac{\partial p}{\partial t}+\beta\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}\right)=\frac{\partial p^*}{\partial t} \tag{1d}$$

Here, t is time; x, y, and z are Cartesian coordinates; u,v, and w are corresponding Cartesian velocity components; p is the pressure; and τ_{ij} is the viscous stress tensor. The parameter $1/\beta$ is the pseudo compressibility, and p^* is the value of p at the previous iteration. The β and p^* terms, and the numerical algorithm are chosen to satisfy the continuity equation (1a); they will be discussed in more detail later.

The equations are written in dimensionless form with

$$\begin{aligned}
&\tilde{u}=\frac{u}{u_{ref}},\quad \tilde{v}=\frac{v}{u_{ref}},\quad \tilde{w}=\frac{w}{u_{ref}},\\
&\tilde{x}=\frac{x}{x_{ref}},\quad \tilde{y}=\frac{y}{x_{ref}},\quad \tilde{z}=\frac{z}{x_{ref}},\\
&\tilde{t}=\frac{tu_{ref}}{x_{ref}},\quad \tilde{p}=\frac{p-p_{ref}}{\rho u_{ref}^2},\ \tilde{\tau}_{ij}=\frac{\tau_{ij}}{\rho u_{ref}^2},\\
&\tilde{\nu}=Re^{-1}=\frac{\nu}{x_{ref}u_{ref}},\quad \tilde{\beta}=\frac{\beta}{u_{ref}^2}
\end{aligned} \tag{1e}$$

The subscript ref denotes reference quantities and for convenience the tildes (~) are dropped from the equations.

The viscous stress tensor can be written in the following form:

$$\tau_{ij}=\nu\frac{\partial u_j}{\partial x_j}-\frac{\partial R_{ij}}{\partial x_j} \tag{1f}$$

Here, R_{ij} is the Reynolds stress, and ν is the coefficient of viscosity. Combining Eqs. (1b) and (1d), the governing equations are written as

$$\begin{aligned}
&\frac{\partial D}{\partial t}+\frac{\partial(E-E_v)}{\partial x}+\frac{\partial(F-F_v)}{\partial y}+\frac{\partial(G-G_v)}{\partial z}\\
&\qquad=\left(\frac{\partial p^*}{\partial t},0,0,0\right)^T
\end{aligned} \tag{2a}$$

where

$$D=\begin{bmatrix}p\\\hat{q}\end{bmatrix},\quad E=\begin{bmatrix}\beta u\\\hat{e}\end{bmatrix},\quad F=\begin{bmatrix}\beta v\\\hat{f}\end{bmatrix},\quad G=\begin{bmatrix}\beta w\\\hat{g}\end{bmatrix},$$

$$E_v=\begin{bmatrix}0\\\hat{e}_v\end{bmatrix},\quad F_v=\begin{bmatrix}0\\\hat{f}_v\end{bmatrix},\quad G_v=\begin{bmatrix}0\\\hat{g}_v\end{bmatrix} \tag{2b}$$

Coordinate Transformation

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

$$\begin{aligned} \tau &= t \\ \xi &= \xi(x, y, z, t) \\ \eta &= \eta(x, y, z, t) \\ \varsigma &= \varsigma(x, y, z, t) \end{aligned} \tag{3}$$

Applying this transformation to Eq (2a), the following form of the governing equations is obtained

$$\frac{\partial \hat{D}}{\partial \tau} + \frac{\partial(\hat{E} - \hat{E}_v)}{\partial \xi} + \frac{\partial(\hat{F} - \hat{F}_v)}{\partial \eta} + \frac{\partial(\hat{G} - \hat{G}_v)}{\partial \varsigma} = \frac{1}{J}\left(\frac{\partial p^*}{\partial \tau}, 0, 0, 0\right)^T \tag{4}$$

where

$J = Jacobian\ \ of\ \ the\ \ transformation,$

$$\hat{D} = D/J,$$

$$\hat{E} = \frac{1}{J}\begin{bmatrix} \beta u + \xi_t(p-\beta) \\ uU + \xi_x p \\ vU + \xi_y p \\ wU + \xi_z p \end{bmatrix}, \quad \hat{F} = \frac{1}{J}\begin{bmatrix} \beta v + \eta_t(p-\beta) \\ uV + \eta_x p \\ vV + \eta_y p \\ wV + \eta_z p \end{bmatrix},$$

$$\hat{G} = \frac{1}{J}\begin{bmatrix} \beta w + \varsigma_t(p-\beta) \\ uW + \varsigma_x p \\ vW + \varsigma_y p \\ wW + \varsigma_z p \end{bmatrix} \tag{5}$$

Here, the contravariant velocities, U, V, and W without metric normalization, are defined as

$$\begin{aligned} U &= \xi_t + \xi_x u + \xi_y v + \xi_z w \\ V &= \eta_t + \eta_x u + \eta_y v + \eta_z w \\ W &= \varsigma_t + \varsigma_x u + \varsigma_y v + \varsigma_z w \end{aligned} \tag{6}$$

The viscous terms are

$$\begin{aligned} \hat{E}_v &= \frac{\nu}{J}(\nabla\xi\nabla\xi Im\frac{\partial D}{\partial \xi} + \nabla\xi\nabla\eta Im\frac{\partial D}{\partial \eta} + \nabla\xi\nabla\varsigma Im\frac{\partial D}{\partial \varsigma}) \\ &\quad + (R_{xx}...terms) \\ \hat{F}_v &= \frac{\nu}{J}(\nabla\eta\nabla\xi Im\frac{\partial D}{\partial \xi} + \nabla\eta\nabla\eta Im\frac{\partial D}{\partial \eta} + \nabla\eta\nabla\varsigma Im\frac{\partial D}{\partial \varsigma}) \\ &\quad + (R_{yz}...terms) \\ \hat{G}_v &= \frac{\nu}{J}(\nabla\varsigma\nabla\xi Im\frac{\partial D}{\partial \xi} + \nabla\varsigma\nabla\eta Im\frac{\partial D}{\partial \eta} + \nabla\varsigma\nabla\varsigma Im\frac{\partial D}{\partial \varsigma}) \\ &\quad + (R_{xz}...terms) \end{aligned} \tag{7}$$

where

$$I_m = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad I_m\frac{\partial D}{\partial \xi} = \frac{\partial \hat{q}}{\partial \xi}, etc \tag{8}$$

For orthogonal coordinates, Eq. (7) can be further simplified to read

$$\begin{aligned} \hat{E}_v &= \frac{\nu}{J}(\xi_x{}^2 + \xi_y{}^2 + \xi_z{}^2)I_m\frac{\partial D}{\partial \xi} + (R_{xx}...terms) \\ \hat{F}_v &= \frac{\nu}{J}(\eta_x{}^2 + \eta_y{}^2 + \eta_z{}^2)I_m\frac{\partial D}{\partial \eta} + (R_{yz}...terms) \\ \hat{G}_v &= \frac{\nu}{J}(\varsigma_x{}^2 + \varsigma_y{}^2 + \varsigma_z{}^2)I_m\frac{\partial D}{\partial \varsigma} + (R_{xz}...terms) \end{aligned} \tag{9}$$

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, the turbulence model is confined to an algebraic model to maximize the cost effectiveness of the flow solver. Even though the accuracy of the result could be limited by the algebraic model, economy is still an overriding factor in many engineering applications.

In the present code, 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:

$$R_{ij} = \frac{1}{3}R_{kk}\delta_{ij} - 2\nu_t S_{ij} \tag{10}$$

Here, ν_t is the turbulent eddy viscosity, and R_{kk} is the nomal component of the Reynolds stress. The strain rate tensor is defined by

$$S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \tag{11}$$

By including the normal stress, R_{kk}, in the pressure, ν can be replaced by $(\nu + \nu_t)$. For the turbulent viscosity, the algebraic model of Baldwin and Lomax[17] is implemented in the present code. Following their formulation, the turbulent viscosity for incompressible flow can be written as

$$\begin{aligned} \nu_t &= (\nu_t)_{inner}, \quad y \le y_c \\ &\quad (\nu_t)_{outer}, \quad y > y_c \end{aligned} \tag{12}$$

where y is the normal distance from the wall, and y_c is the smallest value of y at which values from the inner and outer formula are equal.

In the inner region,

$$(\nu_t)_{inner} = l^2|\omega| \tag{13}$$

where $|\omega|$ is the magnitude of local vorticity. The turbulence length scale l and the nondimensional distance from wall y^+ are defined as

$$l = ky[1 - exp(-\frac{y^+}{A})]$$
$$y^+ = (\tau_w/\rho_w)^{1/2} y/\nu_w \tag{14}$$

In the outer region,

$$(\nu_t)_{outer} = KC_{cp}F_{wake}F_{kleb}(y) \tag{15}$$

where

$$F_{wake} = minimum[(y_{max}F_{max}), (C_{wk}y_{max}U^2_{dif}/F_{max}]$$

Here, F_{max} is the first peak value in a profile given by

$$F(y) = y|\omega|[1 - exp(-y^+/A^+)]$$

and y_{max} is the value of y at that point. The Klebanoff intermittency factor is written as

$$F_{kleb}(y) = [1 + 5.5(C_{kleb}y/y_{max})^6]^{-1} \tag{16}$$

and

$$U_{dif} = (\sqrt{u^2 + v^2 + w^2})_{max} - (\sqrt{u^2 + v^2 + w^2})_{min} \tag{17}$$

In wakes, the exponential term in $F(y)$ and the second term in Eq. (17) are set equal to zero.

The constants appearing in the model were determined by requiring agreement with the Cebeci formulation for constant-pressure boundary layers[18]. These values are

$$A = 26, \quad C_{cp} = 1.6, \quad C_{kleb} = 0.3, \quad C_{wk} = 0.25,$$
$$k = 0.4, \quad K = 0.0168$$

III. Numerical Algorithm

Time Advancing

The numerical algorithm used to advance Eq. (4) in time is an implicit, approximately factored, finite-difference scheme by Beam and Warming[13]. By combining trapezoidal-rule time-differencing and the difference form of Eq. (4), the following governing equation in delta-form is obtained:

$$\left\{I + \frac{h}{2}J^{n+1}[\delta_\xi(\hat{A}^n - \Gamma_1) + \delta_\eta(\hat{B}^n - \Gamma_2) + \delta_\varsigma(\hat{C}^n - \Gamma_3)]\right\}(D^{n+1} - D^n)$$
$$= (I - I_m)(p^* - p^n) - (I - \frac{J^{n+1}}{J^n})D^n - hJ^{n+1}(\delta_\xi\hat{E}^n + \delta_\eta\hat{F}^n + \delta_\varsigma\hat{G}^n) + hJ^{n+1}(\delta_\xi\Gamma_1 + \delta_\eta\Gamma_2 + \delta_\varsigma\Gamma_3)D^n \tag{18a}$$

where

$$\hat{A}^n, \hat{B}^n, \hat{C}^n = \begin{bmatrix} L_0 & (L_1\beta) & (L_2\beta) & (L_3\beta) \\ L_1 & (Q + L_1u) & L_2u & L_3u \\ L_2 & L_1v & (Q + L_2v) & L_3v \\ L_3 & L_1w & L_2w & (Q + L_3w) \end{bmatrix}$$

$$Q = L_0 + L_1u + L_2v + L_3w$$
$$L_0 = (\xi_i)_t, \quad L_1 = (\xi_i)_x, \quad L_2 = (\xi_i)_y, \quad L_3 = (\xi_i)_z$$
$$\xi_i = \xi, \eta, \quad or \quad \varsigma \quad for \quad A, B, \quad or \quad C, \quad respectively$$
$$\delta_\xi = finite \quad difference \quad form \quad of \quad \frac{\partial}{\partial\xi}, etc. \tag{18b}$$

The superscript n denotes n^{th} time step, and the viscous terms are given from Eq. (9) as

$$\Gamma_1 = \frac{\nu}{J}\nabla\xi\cdot(\nabla\xi_i Im\delta_{\xi_i}) + (R_{xx}...terms)$$
$$\Gamma_2 = \frac{\nu}{J}\nabla\eta\cdot(\nabla\xi_i Im\delta_{\xi_i}) + (R_{yx}...terms) \tag{18c}$$
$$\Gamma_3 = \frac{\nu}{J}\nabla\varsigma\cdot(\nabla\xi_i Im\delta_{\xi_i}) + (R_{zx}...terms)$$

Approximate Factorization

The full viscous terms $\hat{E}_v$, $\hat{F}_v$ and $\hat{G}_v$ in Eq. (7) produce non-tridiagonal elements in the left-hand side of Eq. (18a). Therefore, to implement an approximate factorization scheme, 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, the full viscous terms may be included. In the present version, a nearly orthogonal grid is used, and the viscous terms in Eq. (7) are simplified to

$$\hat{E}_v = \frac{\nu}{J}(\nabla\xi\cdot\nabla\xi_i)Im\delta_\xi D = \gamma_1 I_m\delta_\xi D$$
$$\hat{F}_v = \frac{\nu}{J}(\nabla\eta\cdot\nabla\eta_i)Im\delta_\eta D = \gamma_2 I_m\delta_\eta D \tag{19}$$
$$\hat{G}_v = \frac{\nu}{J}(\nabla\varsigma\cdot\nabla\varsigma_i)Im\delta_\varsigma D = \gamma_3 I_m\delta_\varsigma D$$

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

$$[I + \frac{h}{2}J^{n+1}\delta_\xi(\hat{A}^n - \gamma_1 I_m\delta_\xi) + \epsilon_i\nabla_\xi\Delta_\xi]\cdot$$
$$[I + \frac{h}{2}J^{n+1}\delta_\eta(\hat{B}^n - \gamma_2 I_m\delta_\eta) + \epsilon_i\nabla_\eta\Delta_\eta]\cdot$$
$$[I + \frac{h}{2}J^{n+1}\delta_\varsigma(\hat{C}^n - \gamma_3 I_m\delta_\varsigma) + \epsilon_i\nabla_\varsigma\Delta_\varsigma](D^{n+1} - D^n)$$
$$= [RHS \quad (18a)] - \epsilon_e[(\nabla_\xi\Delta_\xi)^2 + (\nabla_\eta\Delta_\eta)^2 + (\nabla_\varsigma\Delta_\varsigma)^2]D^n \tag{20}$$

where ϵ_i and ϵ_e are implicit and explicit smoothing terms (which are explained in the next section), and

$$h = \Delta\tau = time - step,$$

$$\nabla_\xi D = D_j - D_{j-1}, \quad \Delta_\eta D = D_{K+1} - D_k,$$

$$\delta_\zeta D = (D_{l+1} - D_{l-1})/(2\Delta\zeta)$$

$$\delta_\eta \gamma \delta_\eta D = [(\gamma_{k+1} + \gamma_k)(D_{k+1} - D_k) - (\gamma_k + \gamma_{k-1})(D_k - D_{k-1})]/[2(\Delta\eta)^2]$$

Analogous terms in the η and ζ directions are defined similarly.

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, 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:

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0 \tag{21}$$

The usual upwind differencing of the convective term results in

$$c\delta_x u = \begin{cases} c(3u_j - 4u_{j-1} + u_{j-2})/(2\Delta x), & c > 0 \\ c(-3u_j + 4u_{j+1} - u_{j+2})/(2\Delta x), & c < 0 \end{cases} \tag{22a}$$

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

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

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

$$c\delta_x u = \frac{c}{2\Delta x}(u_{j+1} - u_{j-1}) + \frac{|c|}{4\Delta x}(u_{j-2} - 4u_{j-1} + 6u_j - 4u_{j+1} + u_{j+2}) \tag{22b}$$

Therefore, a finite-difference form of Eq. (21) with a fourth order explicit smoothing term can be written as

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + c\frac{u_{j+1}^{n+1} - u_{j-1}^{n+1}}{2\Delta x} = -\frac{\epsilon_e|c|}{4\Delta x}(u_{j-2} - 4u_{j-1} + 6u_j - 4u_{j+1} + u_{j+2}), \quad 0 < \epsilon_e \leq 1 \tag{23a}$$

and, in delta form, it can be written as

$$\Delta u + \frac{c\Delta t}{2\Delta x}(\Delta u_{j+1} - \Delta u_{j-1}) = -\frac{c\Delta t}{2\Delta x}(u_{j+1} - u_j)^n - \frac{\epsilon_e|c|\Delta t}{4\Delta x}(u_{j-2} - 4u_{j-1} + 6u_j - 4u_{j+1} + u_{j+2})^n \tag{23b}$$

where

$$\Delta u = u^{n+1} - u^n$$

$$\epsilon_e = explicit \quad smoothing \quad coefficient$$

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

$$c\delta_x u = \frac{c}{2\Delta x}(u_{j+1} - u_{j-1}) - \frac{|c|}{2\Delta x}(u_{j+1} - 2u_j + u_{j-1}) \tag{24a}$$

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

$$\Delta u + \frac{c\Delta t}{2\Delta x}(\Delta u_{j+1} - \Delta u_{j-1}) - \frac{\epsilon_i|c|\Delta t}{2\Delta x}(\Delta u_{j+1} - 2\Delta u_j + \Delta u_{j-1}) = -\frac{c\Delta t}{2\Delta x}(u_{j+1} - u_{j-1})^n - \frac{\epsilon_e|c|\Delta t}{4\Delta x}(u_{j-2} - 4u_{j-1} + 6u_j - 4u_{j+1} + u_{j+2})^n \tag{24b}$$

where

$$\epsilon_i = implicit \quad smoothing \quad coefficient$$

$$\epsilon_e = explicit \quad smoothing \quad coefficient$$

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

$$\frac{\partial u'}{\partial t} - \epsilon\frac{\partial^2 u'}{\partial^2 x} = 0 \tag{25a}$$

then

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

For the fourth-order smoothing term,

$$\frac{\partial u'}{\partial t} + \epsilon\frac{\partial^4 u'}{\partial^4 x} = 0 \tag{25b}$$

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, and the fourth-order extrapolated smoothing term is better on solid boundaries.

Pseudo Compressibility and the Pressure Field Solution

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 [Eq. (1d)], thus, introducing pressure waves of finite speed; the wave speed is infinity for truly incompressible flow. Wave propagation depends on the magnitude of the compressibility parameter, β, which, if chosen to be too large, would at the same time contaminate the accuracy of the approximate factorization. Therefore, a correct specification of β is of prime importance to the success of this approach. For instance, a certain value of the pseudo compressibility suitable for a particular problem can be totally inadequate for other geometries and flow speeds, and can even cause instability.

In Ref. 14, wave-propagation characteristics are analyzed using a one-dimensional form of the governing equations. By comparing the velocity of pressure-wave propagation and the rate of vorticity spreading, the following criterion for the artificial compressibility was derived:

$$\beta >> [1 + \frac{4}{Re}\left(\frac{x_{ref}}{x_\delta}\right)^2\left(\frac{x_L}{x_{ref}}\right)]^2 - 1 \qquad (26)$$

where x_δ and x_L are the characteristic lengths that the vorticity and the pressure waves have to propagate during a given time span. During a duct-flow simulation, x_L is equal to the total length of the duct, and x_δ is half the distance between the two walls of the duct. The computational experiments showed this dependency of the pseudo compressibility on the Reynolds number, as well as on the characteristic lengths of the geometry.

For the near-field in external flows where pressure waves propagate out to infinity the choice of β is less restrictive. The pressure buildup in the near-field region is only temporary, even if β is not optimum. However, for internal flows, local pressure buildup can be serious when β is not properly chosen (see Ref. 14 for details).

Boundary Condition

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: 1) solid surface, 2) far-field, 3) in-flow and out-flow, and 4) symmetric or reflective boundaries. All of these are required in the present code.

On a solid surface, the ususal no-slip condition is applied. By taking advantage of the viscous sublayer assumption where the pressure is constant normal to the surface, one obtains

$$p_n = 0 \qquad (27)$$

The boundary condition can be implemented either explicitly or implicitly. The latter enhances stability of the code. For this discussion, a $\varsigma = const$ surface is considered next. On this surface, Eq. (27) can be written as

$$\nabla\varsigma \cdot \nabla p = 0 \qquad (28a)$$

Expanding this,

$$\varsigma_x(\xi_x p_\xi + \eta_x p_\eta + \varsigma_x p_\varsigma) + \varsigma_y(\xi_y p_\xi + \eta_y p_\eta + \varsigma_y p_\varsigma)$$
$$+\varsigma_z(\xi_z p_\xi + \eta_z p_\eta + \varsigma_z p_\varsigma) = 0 \qquad (28b)$$

Since a nearly orthogonal grid is assumed near the surface, this can be further simplified to

$$\frac{\partial p}{\partial \varsigma} = 0 \qquad (28c)$$

This condition is implemented in the ς-directional sweep. From Eq. (20), the matrix equation on a solid surface (at L=1) is written as follows:

$$\hat{a}\Delta D_{(L=0)} + \hat{b}\Delta D_{(L=1)} + \hat{c}\Delta D_{(L=2)} = \hat{f} \qquad (29)$$

where $\Delta D = D^{n+1} - D^n$. To implement Eq. (28c) explicitly, one simply sets

$$\hat{a} = \hat{c} = \hat{f} = 0, \quad \hat{b} = I \qquad (30a)$$

then the pressure is updated at the end of each step. The same procedure is used for fixed boundaries such as a free stream boundary. Equation (28c) is applied in an implicit manner by setting

$$\hat{a} = [0], \quad \hat{b} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \hat{c} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$
$$\hat{f} = \begin{bmatrix} p_{L=2} - p_{L=1} \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad (30b)$$

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. 19). 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. The three-dimensional coordinate system chosen is shown in Figs. 1a and 1b. Figs. 2a and 2b show the steady-state velocity vectors and stream function

contours. The steady-state pressure coefficient on the cylinder surface is then compared in Figs. 3a with that of Mehta (private communication), who used a stream function and vorticity formulation in two dimensions. Various experimental and computational studies on this bench-mark case have been reported. The quantities, that are presently used to verify the physical phenomena are the wake length, L_{wake}; the separation angle, θ_{sep} (see Fig. 2b); pressure drag, C_{dp}; and the pressure coefficients at the forward and rear stagnation points, C_{pf} and C_{pr}, respectively. The flow field 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 an efficient 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. Figures 4a and 4b show the developing laminar channel flow. To reduce the channel length for obtaining fully developed flow, a partially developed boundary layer profile is used as an inflow condition. Using a nondimensional time-step of 0.1, fully developed flow is obtained after about 100 steps for a Reynolds number of 1,000, based on channel width. The Baldwin-Lomax turbulence model is tested by increasing the Reynolds number to 100,000. The velocity vectors for this case are shown in Fig. 5a. The mean velocity defect is compared, in Fig. 5b, with Prandtl's universal law, Comte-Bellot's experiment for channel flow,[20] and Klebanoff's well-known boundary layer profile. The convergence histories for both laminar and turbulent cases are shown in Fig. 6.

Three-Dimensional Flow Examples

One of the prime objectives in developing the present code is to analyze and verify the flow field in the Space Shuttle Main Engine (SSME) power head. The examples shown in Figs. 7, 8 and 9 exhibit typical flow characteristics locally encountered in the SSME simulation.

The rectangular duct shown in Fig. 7a is similar to part of the new 80 x 120 low-speed wind tunnel at Ames Research Center (guide vanes are not shown). The flow, in the form of velocity vectors, is shown at one station in Fig. 7b. This clean-duct simulation shows the nature and magnitude of the secondary flow, separated region, and substantial variation in mean velocity vectors (Figs. 8a and 8b) from top or bottom wall to the center of the duct.

Figures 9a and 9b show the geometry of an annular duct with a 180° bend. This configuration represents the turn-around duct in the hot-gas manifold of the Shuttle engine. The laminar-flow (Re=1,000) solution in Fig. 9c shows a considerable nonuniformity after the 180° 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 x 21 x 16 mesh was used with a non-dimensional 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 1.44E-04 sec per mesh point on the Cray 1S 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, will be presented in a future paper.

V. Concluding Remarks

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 a 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, Reno, Nev., 1983.

[2]Mehta, U. B., " Dynamic Stall of an Oscillating Airfoil", Paper No.23, AGARD Conference Proceedings No. 227, Ottawa, Canada, Sept. 1977.

[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 Method in Fluid Dynamics, Aachen, Germany, June 28-July 2, 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 Method in Fluid Dynamics, Aachen, Germany, June 28-July 2, 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 Method in Fluid Dynamics, Aachen, Germany, June 28-July 2, 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, No. 9, 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, Danver, Mass., 1983.

[8]Pulliam, T. H. and Steger, J. L., " Implicit Finite-Difference Simulations of Three-Dimensional Compressible Flow," AIAA Journal, Vol. 18, No. 2, 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 Note 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, No. 12, 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 Jounal, Vol. 15, No. 4, Apr. 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, Reno, Nev., 1984.

[15]Chakravarthy, S. R., " Numerical Simulation of Laminar Incompressible Flow within Liquid Filled Shells," Report ARBRL-CR-00491, U.S. Army Ballistics Research Laboratory, Aberdeen Proving Ground, Md., 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, Huntsville, Ala., 1978.

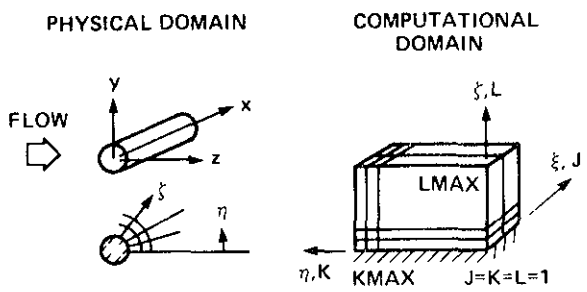
[18]Cebeci, T., " Calculation of Compressible Turbulent Boundary Layers with Heat and Mass Transfer," AIAA Paper 70-741, Los Angeles, Calif., 1970.

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

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

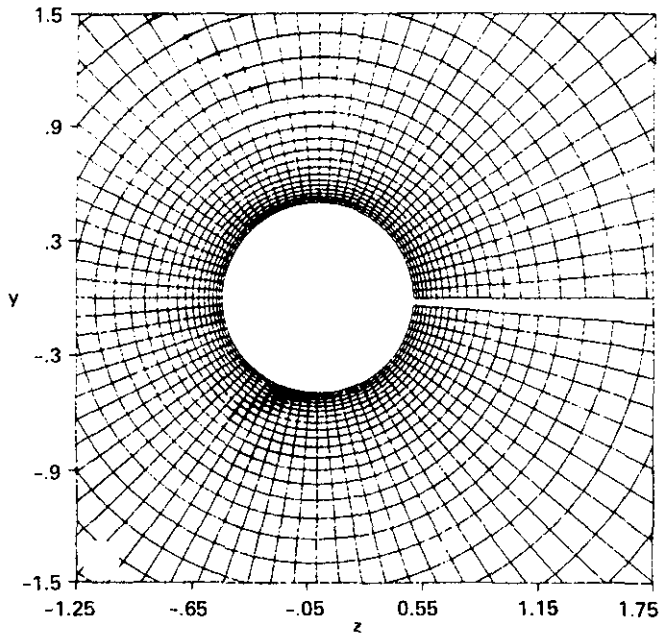
Table 1 Comparison of steady-state flow field
(flow over a circular cylinder at Re = 40 based on diameter)

	C_{dp}	C_{pf}	C_{pr}	θ_{sep}	L_{wake}
Summary of literature data	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



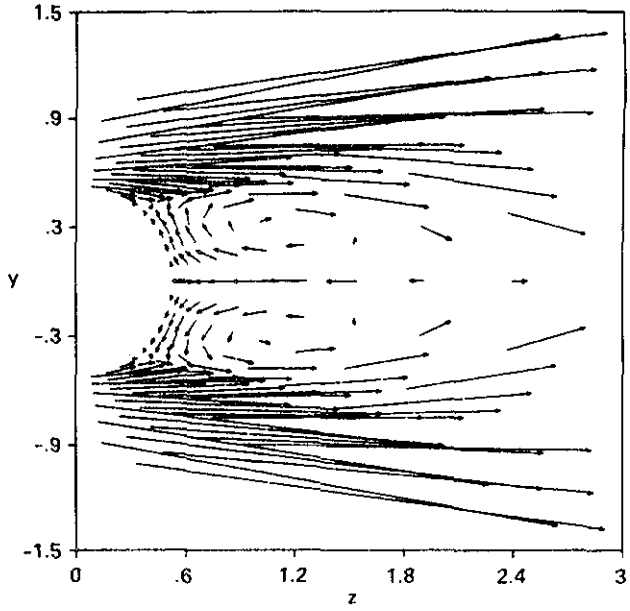
(JMAX,KMAX,IMAX) = (3,80,41)
STRETCHING RATIO = 1.15

a) *coordinate system.*

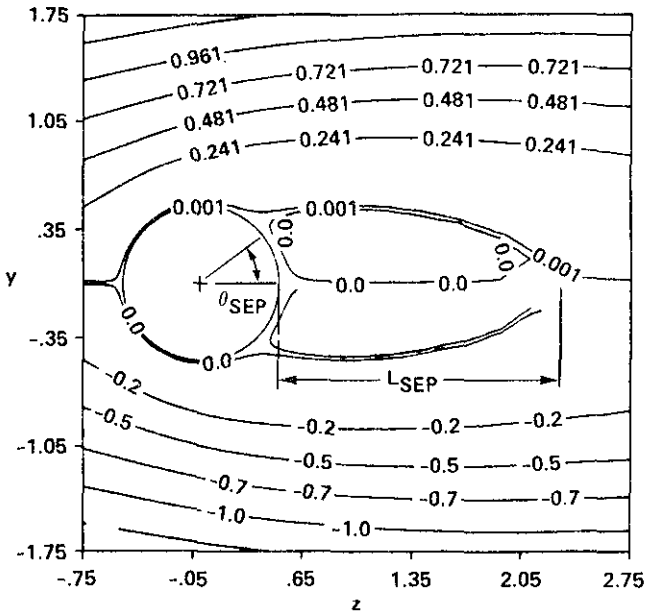


b)grid.

Fig. 1 Topology for flow over a circular cylinder.

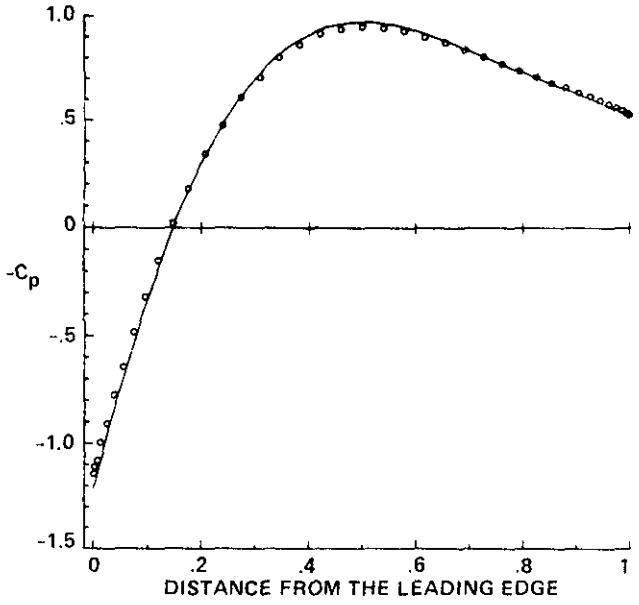


a) velocity vectors,

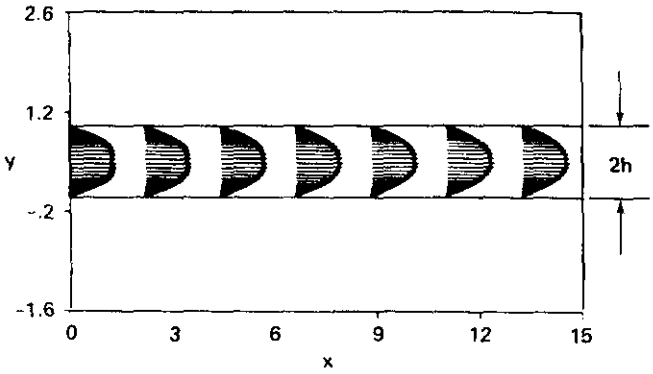


b)stream-function contours.

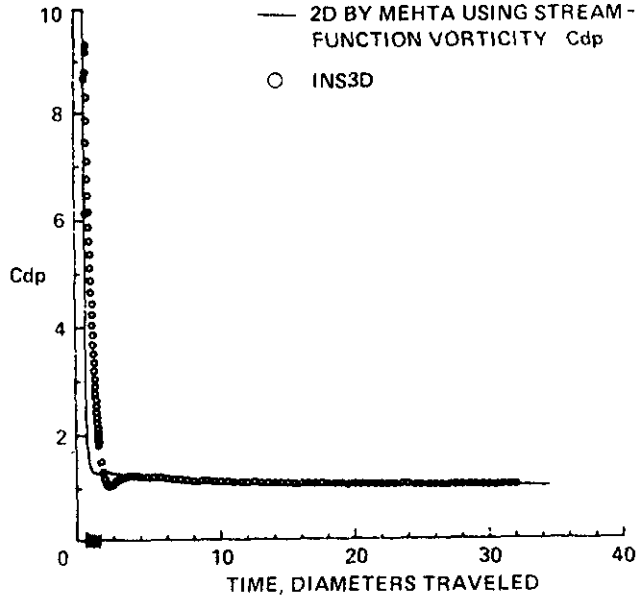
Fig. 2 Steady state solution for flow over a circular cylinder at Re=40.



a) coordinate system.

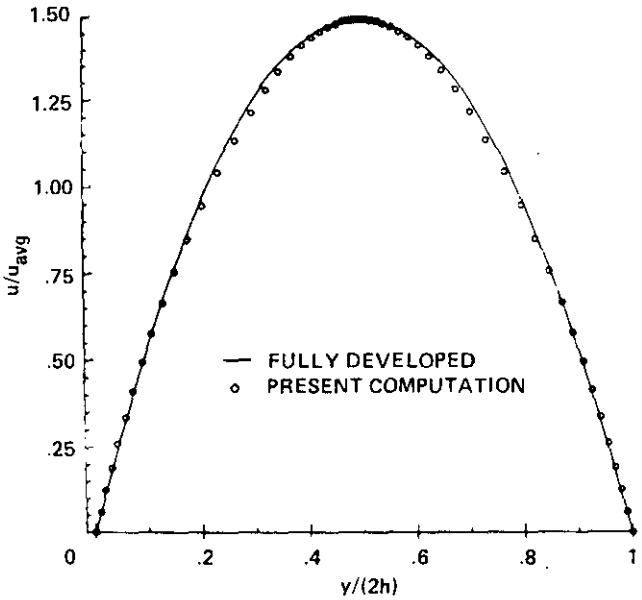


a) velocity vector.



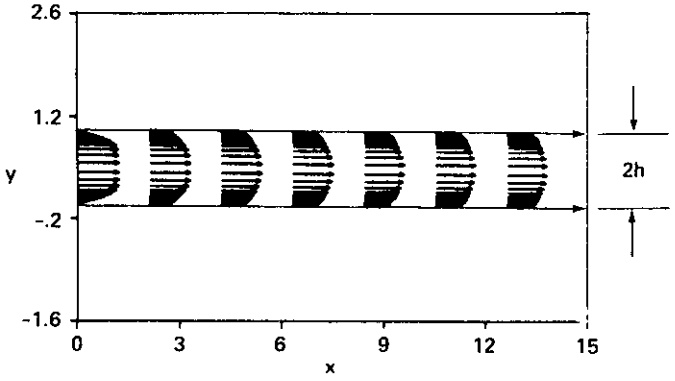
b)grid

Fig. 3 Flow over an impulsively started circular cylinder at Re=40.



b) fully developed velocity profile.

Fig. 4 Developing laminar channel flow at Re = 1000. (Re based on channel width and average velocity.)



a) velocity vectors.

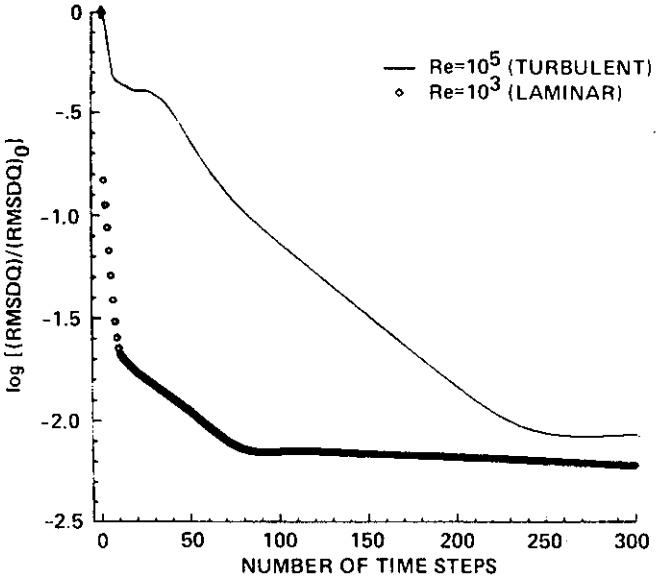
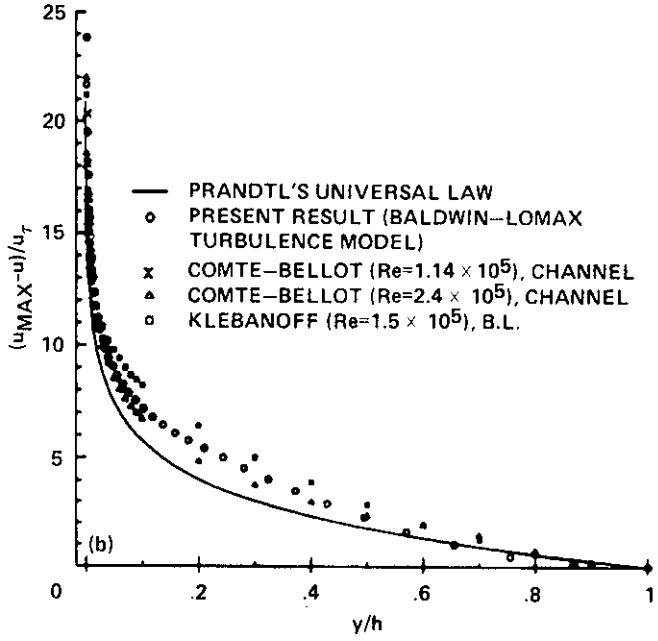
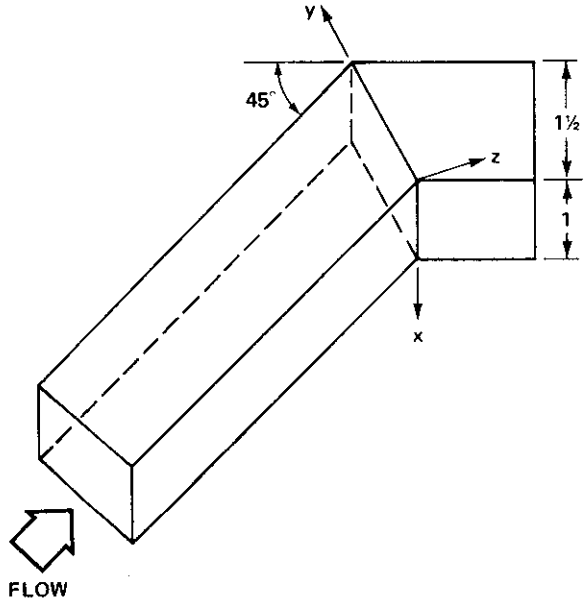


Fig. 6 Convergence history comparison for laminar and turbulent channel flow.

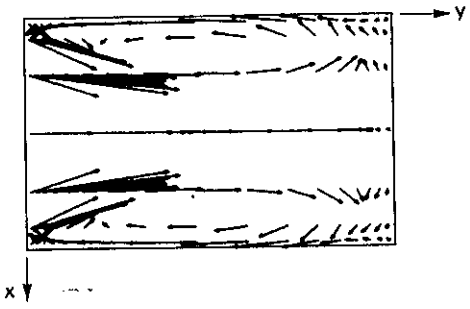


b) mean velocity defect comparison.

Fig. 5 Developing turbulent channel flow at Re $= 10^5$.

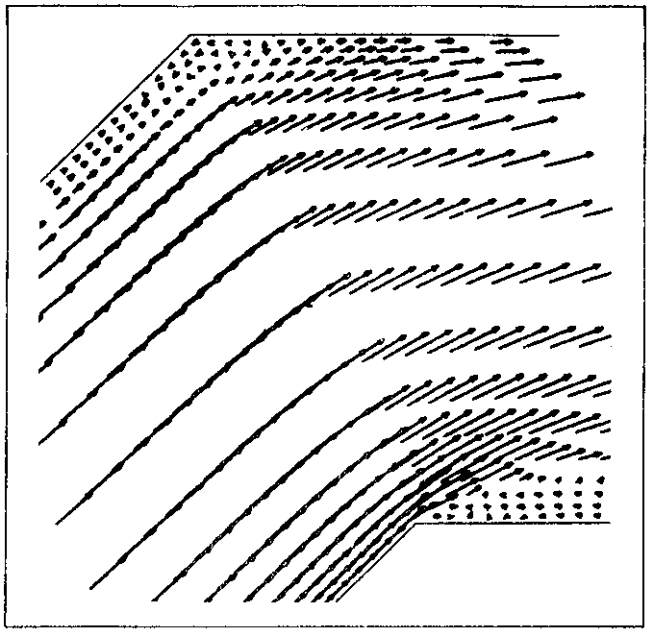


a) Duct geometry.

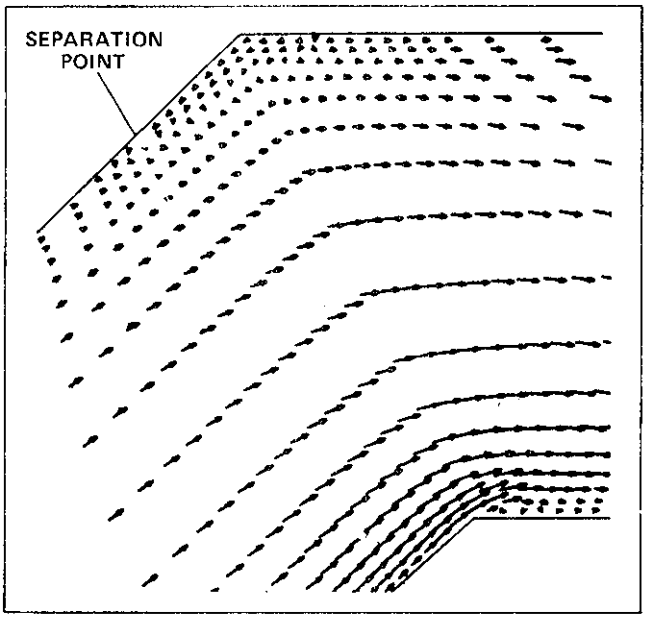


b) secondary flow at z=0 plane (view from down stream).

Fig. 7 Flow through a rectangular duct with a 45° bend.

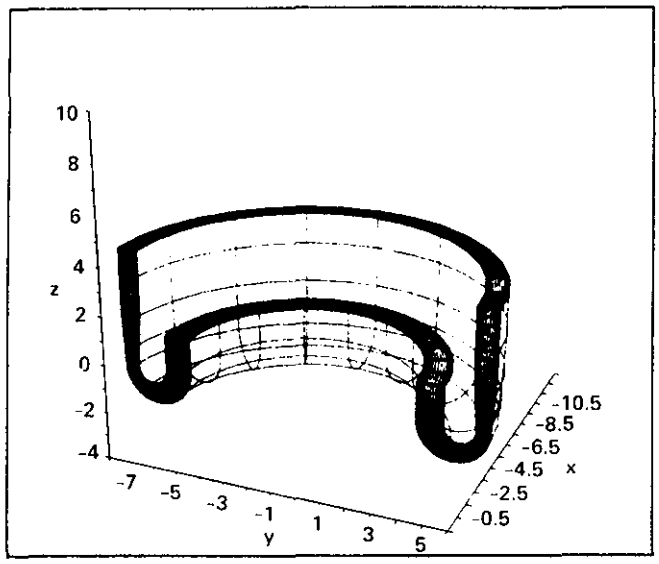


a) Velocity vector at x=0.5 plane.

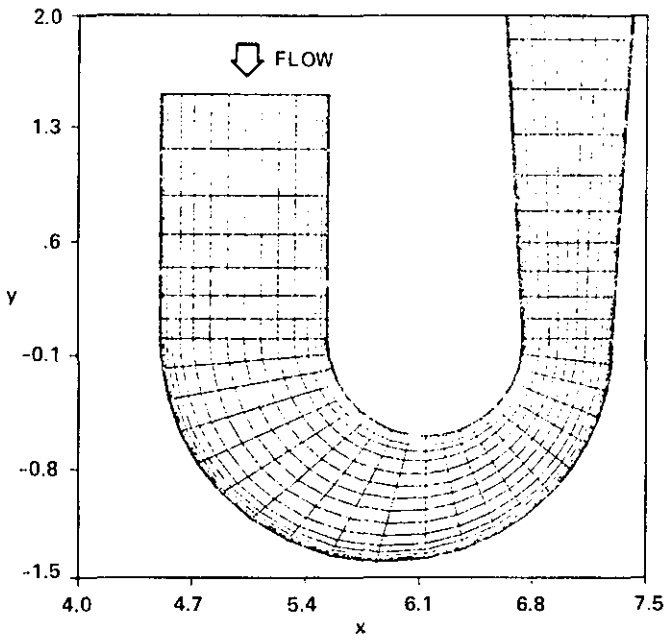


b) velocity vector at x=0.025 plane (top view).

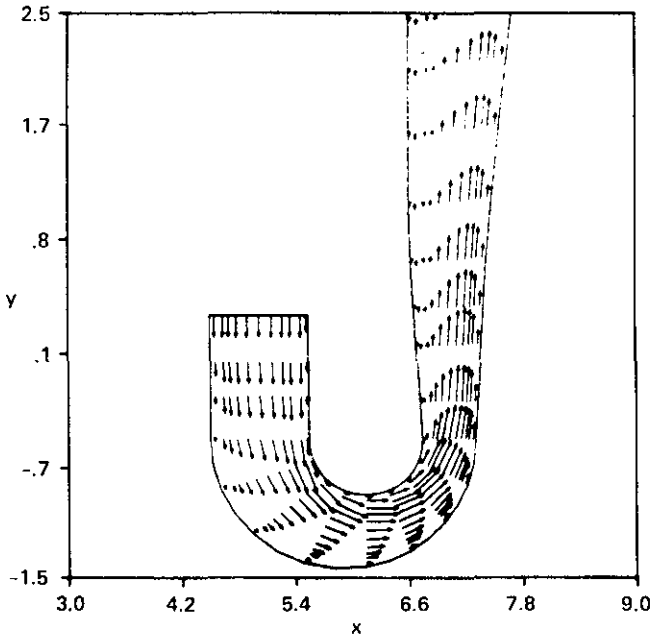
Fig. 8 Flow through a rectangular duct with a 45° bend.



a) Three-dimensional grid.



b)two-dimensional cross section.



c) typical flow pattern with separation.

Fig. 9 Flow through a turn-around duct.