NASA Technical Memorandum 101090

Computation of Viscous Incompressible Flows

Dochan Kwak

March 1989

NASA
National Aeronautics and
Space Administration

NASA Technical Memorandum 101090

Computation of Viscous Incompressible Flows

Dochan Kwak
Ames Research Center, Moffett Field, California

March 1989

NASA
National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035

TABLE OF CONTENTS

Page

I. INTRODUCTION

Viscous incompressible flows have played an important role both in basic fluid dynamics research and in industrial applications. In two-dimensional (2-D) studies, incompressible flows have been a convenient choice for laboratory experiments such as low speed wind-tunnel or water-tank experiments. Since computational study of these flow problems has been performed for several decades, various viscous incompressible flow solution methods have been developed. In many, perhaps a majority of, realistic engineering applications, incompressible flows are also encountered. To name a few, there are problems related to low speed aerodynamics and hydrodynamics such as the flow around high lift devices, the flow around submerged vehicles, flow through impeller passages, mixing of the flow in chemical reactors, coolant flow in nuclear reactors, and certain biofluid problems. There are algorithmic simplifications as well as geometric modeling involved in computing these flows. Furthermore, significant physical modeling is also required, such as turbulence modeling for high-Reynolds number flows. Therefore, numerical computation of these flows, especially in three dimensions, becomes partly an art and partly a science, hence we use the term 'numerical simulation'.

To date, computational flow simulation has not been a critical element in resolving many engineering problems. For instance, impellers, automobiles, submarines, and chemical reactors have been designed reasonably well by empirical means. As technologies advance, modern flow devices tend toward a more compact and highly efficient design. Therefore, the approach relying on empiricism and simplified analysis becomes inadequate for resolving problems associated with those devices thar require advanced analysis. For example, in analyzing and redesigning the current Space Shuttle main engine (SSME) power head, computational simulations became an economical and time-saving supplement to experimental data. There are vast numbers of other real-world problems which demand accurate viscous, incompressible flow solutions, such as rocket-engine fuel flow, flow through an impeller, and blood flow through a ventricular assist device. Therefore, it is of considerable interest to have a computational fluid dynamics (CFD) capability for simulating these applications as an alternative to analytical or empirical approaches.

The present lecture note attempts to summarize recent progress in developing viscous incompressible flow solvers and their application to real world problems, both fundamental and applied in nature. Because of the wide range of geometries of interest, flexibility and convenience becomes an important part of the flow solvers. Among various approaches, the primitive variable formulation is considered to be the most promising for three-dimensional (3-D) applications and is therefore emphasized. In the present note, some details of derivation of the equations and algorithms are given as well as the physical interpretation of the solution procedures, mainly within a finite-difference framework.

In Chapter II, a brief review on existing methods using primitive variables is given. In Chapter III, the pseudocompressibility method is explained in some detail followed by a description of a fractional step method in Chapter IV. Then, in Chapter V, various validation computations are presented. Finally, in Chapter VI, examples of real world applications are presented.

II. SOLUTION METHODS

In this chapter, a brief review is given on promising methods for general applications to 3-D real world problems. Naturally for 3-D applications, flexibility, robustness and computational efficiency as well as accuracy become very important. Other methods used in the past for more fundamental fluid dynamics problems can be found in the literature (see, for example, Roach, 1972; Orszag and Israeli, 1974).

2.1 Formulation

Unsteady, 3-D, incompressible flow with constant density is governed by the following Navier-Stokes equations:

$$\frac{\partial u_i}{\partial x_i} = 0 \tag{2.1}$$

$$\frac{\partial u_i}{\partial t} + \frac{\partial u_i u_j}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} \tag{2.2}$$

where t is the time, x_i the Cartesian coordinates, u_i the corresponding velocity components, p the pressure, and τ_{ij} the viscous-stress tensor. Here, all variables are nondimensionalized by a reference velocity and length scale. The viscous stress tensor can be written as

$$\tau_{ij} = 2\nu S_{ij} - R_{ij} \tag{2.3}$$

$$S_{ij} = \frac{1}{2}(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}) \tag{2.4}$$

Here, ν is the kinematic viscosity, S_{ij} is the strain-rate tensor, and R_{ij} is the Reynolds stresses. Various levels of closure models for R_{ij} are possible. In the present study, turbulence is simulated by an eddy viscosity model using a constitutive equation of the following form:

$$R_{ij} = \frac{1}{3}R_{kk}\delta_{ij} - 2\nu_t S_{ij} \tag{2.5}$$

where ν_t is the turbulent eddy viscosity. By including the normal stress, R_{kk}, in the pressure, ν in equation (2.3) can be replaced by $(\nu + \nu_t)$ as follows.

$$\tau_{ij} = 2(\nu + \nu_t)S_{ij} = 2\nu_T S_{ij} \tag{2.6}$$

In the remainder of this note the total viscosity, ν_T, will be represented simply by ν. Therefore, in describing solution methods and algorithms, the incompressible Navier-Stokes equations are modified to allow variable viscosity in the present formulation. Hence, the flow solvers described here include turbulent flow cases as well.

The major difference between the incompressible and the compressible Navier-Stokes formulation is the lack of a time derivative term in the continuity equation in the incompressible formulation. Therefore, satisfying the mass conservation equation is the primary issue in solving the above set of equations. Physically, incompressible flow is characterized by elliptic behavior of the pressure waves, the speed of which in a truly incompressible flow is infinite. The pressure field is desired as a part of the solution. However, the pressure cannot be obtained directly from the governing equations. One can utilize a stream function-vorticity formulation to eliminate the pressure. This method has been successfully used in many 2-D flow problems. Since an extension of this method to three dimensions is difficult, a vorticity-velocity method was developed to remove pressure from the solution procedure. This method eliminates the use of pressure, however, at the expense of introducing vorticity boundary conditions which are the key to the development of the flow field and not straightforward to specify. In realistic 3-D problems, these derived quantities are difficult to define or impractical to use. The primitive variable formulation, namely, using pressure and velocities as dependent variables, then becomes more convenient and flexible in 3-D applications. However, in this formulation, mass conservation and its relation to pressure has to be properly handled while achieving computational efficiency. Various techniques have been developed in the past, none of which have been proven to be universally better than the other.

2.2 Method Based on Pressure Iteration

2.2.1 MAC Method

Perhaps the first primitive variable method is the marker-and-cell (MAC) method developed by Harlow and Welch (1965). In this method, the pressure is used as a mapping parameter to satisfy the continuity equation. By taking the divergence of the momentum equation, the Poisson equation for pressure is obtained:

$$\nabla^2 p = \frac{\partial h_i}{\partial x_i} - \frac{\partial}{\partial t}\frac{\partial u_i}{\partial x_i} = g \tag{2.7}$$

where

$$h_i = -\frac{\partial u_i u_j}{\partial x_j} + \frac{\partial \tau_{ij}}{\partial x_j}$$

The usual computational procedure involves choosing the pressure field at the current time step such that continuity is satisfied at the next time step. The original MAC method is based on a staggered arrangement on a 2-D Cartesian grid (figure 2.1). The staggered grid conserves mass, momentum, and kinetic energy in a natural way and avoids odd-even point decoupling of the pressure encountered in a regular grid (Gresho and Sani, 1987). However, these properties become unclear when applied to generalized coordinates. Even though the original method used an explicit Euler solver, various time advancing schemes can be implemented here. Ever since its introduction, numerous variations of the MAC method have been devised and successful computations have been made. Many more examples can be found in the literature, for example, Roach (1972), Ferziger (1987), and Orszag and Israelli (1974).

The major drawback of this method is the amount of large computing time required for solving the Poisson equation for pressure. When the physical problem requires a very small time step, the penalty paid for an iterative solution procedure for the pressure may be tolerable. But the method as a whole is slow and the pressure boundary condition is difficult to specify.

One important aspect of the numerical solution of the Poisson equation for pressure is tied to the spacial differencing of the second derivatives. To illustrate this, equation (2.7) is solved in three different ways as follows:

Method 1 :

First, an exact form of the Laplacian operator is used in solving equation (2.7). The Fourier transform of equation (2.7) is

$$-k^2\hat{p} = \hat{g}' \tag{2.8a}$$

where

$$\begin{aligned} \hat{p} &= \text{Fourier transform of p} \\ k^2 &= {k_x}^2 + {k_y}^2 + {k_z}^2 \\ k_x, k_y, k_z &= \text{wave numbers in the x}-,\ \text{y}-,\ \text{and z} - \text{directions} \\ g' &= \text{finite difference approximation to} \quad g \end{aligned}$$

The wave number, k_i, in discrete Fourier expansion is defined as

$$\begin{aligned} k_i &= \frac{2\pi}{N\Delta} n = \text{wave number in the } \mathrm{x_i} - \text{direction} \\ n &= -N/2, ...0, 1, ..(N/2-1) \\ \Delta &= \text{mesh space} \\ N &= \text{number of mesh points in the } \mathrm{x_i} - \text{direction} \end{aligned}$$

By inverse transformation, p can be obtained.

Method 2:
A second approach is to use the difference form of the second derivatives in equation (2.7),

$$\left(\frac{\delta^2}{\delta x^2} + \frac{\delta^2}{\delta y^2} + \frac{\delta^2}{\delta z^2}\right) p = g' \qquad (2.8b)$$

The Fourier transform of the above equation is

$$-\bar{k}_i \bar{k}_i \hat{p} = \hat{g}'$$

where $\bar{k}_i$ is the Fourier transform of the difference form of the second derivative.

Method 3:
The finite difference form of the governing equations is

$$\frac{\delta u_i}{\delta t} - h'_i = -\frac{\delta p}{\delta x_i} = Gp \qquad (2.1')$$

$$\frac{\delta u_i}{\delta x_i} = Du_i = 0 \qquad (2.2')$$

where h'_i is the finite difference form of h_i. By applying the divergence operator, D, to the above equations, the following difference form of the Poisson equation is obtained.

$$DGp = -\frac{\delta Du_i}{\delta t} + Dh'_i = g'_i$$

Then, taking the Fourier transform of the above equation,

$$-k'_i k'_i \hat{p} = \hat{g}' \qquad (2.8c)$$

Since equation (2.1′) is solved numerically, this equation is compared in the above three methods. Again, the Fourier transform of equation (2.1′) is

$$\frac{\delta \hat{u}_i}{\delta t} - \hat{h}'_i = -ik'_i\hat{p} \tag{2.9}$$

To satisfy the continuity equation in grid space, the following

$$\frac{\delta D}{\delta t} = 0$$

has to be satisfied for a flow that has $Du_i = 0$ at the beginning. In Fourier space, this is equivalent to $k'_i u_i = 0$ at the first and next time steps. Substituting $\hat{p}$ from the above three methods into equation (2.9), the following results are obtained.

For Method 1:

$$\frac{\delta}{\delta t}(k'_i \hat{u}_i) = (\hat{h}'_i - \frac{k'_i k'_j}{k^2}\hat{h}'_j)k'_i \neq 0$$

For Method 2:

$$\frac{\delta}{\delta t}(k'_i \hat{u}_i) = (\hat{h}'_i - \frac{k'_i k'_j}{\bar{k}^2}\hat{h}'_j)k'_i \neq 0 \tag{2.10}$$

For Method 3:

$$\frac{\delta}{\delta t}(k'_i \hat{u}_i) = (\hat{h}'_i - \frac{k'_i k'_j}{k'^2}\hat{h}'_j)k'_i = 0$$

The error introduced by method 1 and method 2 can be seen by observing the magnitude of k^2, $\bar{k}^2$ and k'^2. For the purpose of comparison, modified wave numbers, $\bar{k}$ and k' are given below when five-point fourth-order central differencing is applied.

$$\begin{aligned} (\bar{k}_i)^2 &= \frac{1}{6\Delta^2}\left[15 - 16cos(\Delta k_i) + cos(2\Delta k_i)\right] \\ (k_i')^2 &= \frac{1}{72\Delta^2}\left[65 - 16cos(\Delta k_i) - 64cos(2\Delta k_i) + 16cos(2\Delta k_i) - cos(4\Delta k_i)\right] \end{aligned} \tag{2.11}$$

The magnitude of these quantities are compared in figure 2.2. Method 3 satisfies the continuity equation at the next time step in grid space, and hence should be used for the pressure field solution. Therefore, in the solution method using a Poisson equation for pressure, the divergence gradient (DG) operator plays an important role in satisfying the mass conservation in grid space. As will be explained later, this strict requirement can be relaxed in other approaches.

2.2.2 SIMPLE Method

The major drawback of the MAC method is that a Poisson equation must be solved for pressure. A direct solver is practical only for simple 2-D cases. However, for 3-D problems, an iterative procedure is the best available choice. The strict requirement of obtaining the correct pressure for a divergence free velocity field in each step significantly slows down the overall computational efficiency. Since, for a steady-state solution, the correct pressure field is desired only when the solution is converged, the iteration procedure for the pressure can be simplified such that it requires only a few iteration at each time step. The best known method using this approach is the Semi-Implicit Method for Pressure-Linked Equations (SIMPLE) developed by Caretto et al. (1972), see also Patankar and Spalding (1972) or Patankar (1980).

The method begins with a guessed pressure p^*, which is usually assumed to be p^n at the beginning of the cycle. Then the momentum equation is solved to obtain an intermediate velocity $u_i{}^*$.

$$u_i{}^* - u_i{}^n = \Delta t\big[fcn(u^n, v^n, u^*, v^*) - \frac{\delta p^*}{\delta x_i}\big] \tag{2.12}$$

The corrected pressure is obtained by setting

$$p = p^* + p' \tag{2.13}$$

The velocity correction is introduced in a similar manner.

$$u_i = u_i{}^* + u_i{}' \tag{2.14}$$

Now the relation between the the pressure correction, p', and the velocity correction, $u_i{}'$, is obtained from a simplified momentum equation. First, the equation for p' and u_i' is obtained from the linearized momentum equations. Then, by dropping all terms involving neighboring velocities, the following form of the pressure correction equation is obtained.

$$u'_i = \omega \frac{\delta p'}{\delta x_i} \tag{2.15}$$

where ω is a function of the particular differencing scheme chosen. Substituting equations (2.14) and (2.15) into the continuity equation, a pressure correction equation is obtained. This is equivalent to taking the divergence of equation (2.15). This procedure in essence results in a simplified Poisson equation for pressure, which can be solved iteratively line-by-line.

The unique feature of this method is the simple way of estimating the velocity correction u_i'. This feature simplifies the computation but introduces empiricism into the method. Despite its empiricism, the method has been used successfully for many computations. It is not the intention of this note to fully evaluate this method. Further details of SIMPLE, SIMPLER and other variations can be found in the literature, for example, see Patankar (1980).

2.2.3 Fractional-Step Method

The fractional-step method can be used for time-dependent computations of the incompressible Navier-Stokes equations (see Chorin, 1968; Yanenko, 1971; Marchuk, 1975). Here, the time evolution is approximated by several steps. Operator splitting can be accomplished in several ways by treating the momentum equation as a combination of convection, pressure, and viscous terms. The common application of this method is done in two steps. The first step is to solve for an auxilary velocity field using the momentum equation in which the pressure-gradient term can be computed from the pressure in the previous time step (Dwyer, 1986) or can be excluded entirely (Kim and Moin, 1985). In the second step, the pressure is computed which can map the auxiliary velocity onto a divergence-free velocity field.

This procedure is illustrated by the following example of Mansour (private communication, NASA Ames, 1986).

Step 1: Calculate auxiliary or intermediate velocity, $\hat{u}_i$, by

$$\frac{\hat{u}_i - u_i{}^n}{\Delta t} = \frac{1}{2}(3H_i{}^n - H_i{}^{n-1}) - \frac{\delta p^n}{\delta x_i} + \frac{1}{2}\frac{1}{Re}\nabla^2(\hat{u}_i + u_i{}^n) \tag{2.16}$$

where

$$H_i = -\frac{\delta}{\delta x_j} u_i u_j$$

The advection terms are advanced by a second-order Adams-Bashforth method while the viscous terms are modified to use an implicit approximate factorization scheme. Here the pressure gradient term is added to the procedure by Kim and Moin (1985), so as to minimize the pressure correction in each time step.

Step 2: Solve for the pressure correction.

The second step of the momentum equation can be written as

$$\frac{u_i{}^{n+1} - \hat{u}_i}{\Delta t} = -\frac{1}{2}\frac{\delta}{\delta x_i}(\phi^{n+1} - \phi^n) \tag{2.17}$$

where

$$p^n = \phi^n - \frac{\Delta t}{2Re}\nabla^2\phi^n$$

This equation combined with continuity equation results in the following Poisson equation for the pressure correction.

$$\nabla^2(\phi^{n+1} - \phi^n) = \frac{2}{\Delta t}\frac{\delta}{\delta x_i}\hat{u}_i \tag{2.18}$$

Once the pressure correction is computed, new pressure and velocities are calculated as follows:

$$\begin{aligned} p^{n+1} &= p^n + (\phi^{n+1} - \phi^n) - \frac{\Delta t}{2Re}\nabla^2(\phi^{n+1} - \phi^n) \\ u_i{}^{n+1} &= \hat{u}_i - \frac{\Delta t}{2}\frac{\delta}{\delta x_i}(\phi^{n+1} - \phi^n) \end{aligned} \tag{2.19}$$

Others (for example, Dwyer, 1986) used essentially the same procedure illustrated above. One particular aspect of the fractional step method requiring special care is the intermediate boundary conditions. Orszag et al. (1986) discussed this extensively. As will be explained later in this note, Rosenfeld et al. (1988) devised a generalized scheme where physical boundary conditions can be used at intermediate steps. As with other pressure based methods, the efficiency of the fractional step method depends on the Poisson solver. A multigrid acceleration, which is physically consistent with the elliptic field, is one possible avenue to enhance the computational efficiency. The procedure described above was applied in Cartesian coordinates to calculate time dependent flows using either a staggered grid (Kim and Moin, 1985) or a regular grid (Dwyer, 1986). Extension of the staggered grid to generalized coordinates is not straightforward and involves features related to geometric conservation. More details of this extension will be described later.

2.3 Pseudocompressibility Method

Recent advances in the state of the art in CFD have been made in conjunction with compressible flow computations. Therefore, it is of significant interest to be able to use some of these compressible flow algorithms. To do this, the artificial compressibility method of Chorin (1967) can be used. In this formulation, the continuity equation is modified by adding a time-derivative of the pressure term resulting in

$$\frac{1}{\beta}\frac{\partial p}{\partial t} + \frac{\partial u_i}{\partial x_i} = 0 \tag{2.20}$$

where β is an artificial compressibility or a pseudocompressibility parameter. Together with the unsteady momentum equations, this forms a hyperbolic-parabolic type of time-dependent system of equations. Thus, fast implicit schemes developed for compressible flows, such as the approximate-factorization scheme by Beam and Warming (1978), can be implemented. It is to be noted that the t no longer represents a true physical time in this formulation. Various applications which evolved from this concept have been reported for obtaining steady-state solutions (e.g., Steger and Kutler, 1977; Kwak et al. 1986, Chang et al., 1985,1988a,b; Choi and Merkle, 1985). To obtain time-dependent solutions using this method, an iterative procedure can be applied in each physical time step such that the continuity equation is satisfied. Merkle and Athavale (1987) and Rogers and Kwak (1988, 1989) reported successful computations using this pseudo time iteration approach.

In an incompressible flow, a disturbance in the pressure causes waves which travel with infinite speed. When pseudocompressibility is introduced, waves of finite speed result for which the magnitude of the speed depends upon the pseudocompressibility constant β. In a true incompressible flow, the pressure field is affected instantaneously by a disturbance in the flow, but with pseudocompressibility, there will be a time lag between the flow disturbance and its effect on the pressure field. In viscous flows, the behavior of the boundary layer is very sensitive to the streamwise pressure gradient, especially when the boundary layer is separated. If separation is present, a pressure wave traveling with finite speed will cause a change in the local pressure gradient which will affect the location of the flow separation. This change in separated flow will feed back to the pressure field, possibly preventing convergence to a steady state. An extensive mathematical account of the pseudocompressibility approach is given by Temam (1979).

This method has been extensively utilized in developing several incompressible Navier-Stokes codes at NASA Ames Research Center. A more detailed discussion of this method is given separately in the next chapter.

2.4 Method Using Vorticity Transport Equations

Among various other nonprimitive variable methods, the vorticity-velocity method is promising for 3-D applications. This method is briefly introduced here.

A vorticity-velocity method was proposed by Fasel (1976) to study the boundary layer stability problem in two dimensions. A number of other authors used this approach in solving incompressible flow problems (i.e. Dennis et al. 1979; Gatski et al., 1982; Osswald et al., 1987; Hafez et al., 1988). However, a three-dimensional extension of this method has been limited to simple geometries to date.

The basic equations can be summarized as given below.
Instead of the momentum equation, the following vorticity transport equation is used

$$\frac{\partial \omega_i}{\partial t} + \frac{\partial \omega_i u_j}{\partial x_j} = \omega_j \frac{\partial u_i}{\partial x_j} + \nu \nabla^2 \omega \tag{2.21}$$

where the vorticity ω is defined in the conventional way as

$$\omega = \nabla \times \mathbf{u}$$

Taking the curl of the above equation and using the continuity equation, the following Poisson equation for velocity is obtained.

$$\nabla^2 \mathbf{u} = -\nabla \times \omega \tag{2.22}$$

Equations (2.21) and (2.22) can be solved for the velocity and vorticity. Here, the pressure term and the continuity equation are removed at the expense of introducing three vorticity equations. This requires vorticity boundary conditions on a solid surface.

As in the case of the fractional step approach, the computational efficiency of this approach depends on the Poisson solver. Overall performance in general 3-D applications remains to be seen.

III. PSEUDOCOMPRESSIBILITY METHOD

As briefly introduced in the previous chapter, several incompressible Navier-Stokes solvers have been developed based on the pseudocompressibility method. In this chapter, physical characteristics of this method is examined followed by a more detailed discussion on solution procedures which were developed utilizing this method.

3.1 Formulation and Its Physical Characteristics

The use of the pseudocompressibility method as shown by equation (2.20) results in a system of hyperbolic-parabolic equations of motion. Physically, this means that waves of finite speed are introduced into the incompressible flow field as a medium to distribute the pressure. For a truly incompressible flow, the wave speed is infinite, whereas the speed of propagation of these pseudo waves depends on the magnitude of the pseudocompressibility. Ideally, the value of the pseudocompressibility is to be chosen as high as the particular choice of algorithm allows so that the incompressibility is recovered quickly. This has to be done without lessening the accuracy and the stability property of the numerical method implemented. On the other hand, if the pseudocompressibility is chosen such that these waves travel too slowly, then the variation of the pressure field accompanying these waves is very slow. This will interfere with the proper development of the viscous boundary layer, especially when the flow separates. For internal flow, the viscous effect is important for the entire flow field and the interaction between the pseudo pressure-waves and the viscous flow field becomes very important. Pseudocompressibility relaxes the strict requirement of satisfying mass conservation in each step. However, to utilize this convenient feature, it is essential to understand the nature of the pseudocompressibility both physically and mathematically. Chang and Kwak (1984) reported details of the pseudocompressibility, of which some key features are summarized below.

For simplicity of analysis, the 1-D form of the governing equations are considered here.

$$\frac{\partial p}{\partial t} + \beta \frac{\partial u}{\partial x} = 0 \tag{3.1a}$$

$$\frac{\partial u}{\partial t} + \frac{\partial u^2}{\partial x} = -\frac{\partial p}{\partial x} - \tau_w \tag{3.1b}$$

where τ_w is the normalized shear stress at the wall. The normal stress term in the streamwise direction which contributes to the diffusion of the waves is neglected here. Neglecting the shear stress term for simplicity, the following equation is derived from equations 3.1a and 3.1b.

$$\frac{\partial u}{\partial t} + \frac{1}{(u \pm c)}\frac{\partial p}{\partial t} + \left[\frac{\partial u}{\partial x} + \frac{1}{(u \pm c)}\frac{\partial p}{\partial x}\right](u \pm c) = 0 \tag{3.2}$$

The pseudo speed of sound, c, is found to be

$$c = \sqrt{u^2 + \beta} \tag{3.3}$$

Relative to this sound speed, the pseudo Mach number, M, can be expressed as

$$M = \frac{u}{c} = \frac{u}{\sqrt{u^2 + \beta}} < 1 \tag{3.4}$$

It is clear that M is always less than 1 for all $\beta > 0$. Therefore, pseudocompressibility does not introduce shock waves into the system and the flow remains subsonic with respect to the pseudo-sound speed.

To study the main features of the wave propagation, the momentum equation is locally linearized. By cross differencing equations (3.1a) and (3.1b), the following equations are obtained:

$$\frac{\partial^2 p}{\partial t^2} + 2u\frac{\partial^2 p}{\partial t \partial x} - \beta\frac{\partial^2 p}{\partial x^2} = \beta\frac{\partial \tau_w}{\partial x} \tag{3.5a}$$

$$\frac{\partial^2 u}{\partial t^2} + 2u\frac{\partial^2 u}{\partial t \partial x} - \beta\frac{\partial^2 u}{\partial x^2} = -\frac{\partial \tau_w}{\partial t} \tag{3.5b}$$

These equations may be expressed as

$$\left[\frac{\partial}{\partial t} + (u+c)\frac{\partial}{\partial x}\right]\left[\frac{\partial}{\partial t} + (u-c)\frac{\partial}{\partial x}\right]\begin{pmatrix} p \\ u \end{pmatrix} = \begin{pmatrix} \beta\partial\tau_w/\partial x \\ -\partial\tau_w/\partial t \end{pmatrix} \tag{3.6}$$

If the shear stress term on the right hand side of equation (3.6) were absent, characteristic equations for the linear waves would take a simple form expressed by

$$\left[\frac{\partial}{\partial t} + (u+c)\frac{\partial}{\partial x}\right]\begin{pmatrix} p^+ \\ u^+ \end{pmatrix} = 0 \tag{3.7a}$$

$$\left[\frac{\partial}{\partial t} + (u-c)\frac{\partial}{\partial x}\right]\begin{pmatrix} p^- \\ u^- \end{pmatrix} = 0 \tag{3.7b}$$

Waves having a positive sign propagate downstream with a speed $|u + c|$, and waves having a negative sign travel against the stream with a speed $|u - c|$.

The presence of the shear stress term, however, complicates the wave systems since the shear stress depends on the velocity field. The coupling between the pseudo pressure waves and the vorticity spreading depends on their respective time scales. An order-of-magnitude analysis can be done here to obtain a general guideline for pseudocompressibility. Suppose the distance from a point in the flow field such as a point on a flat plate or channel to the downstream boundary is x_L, then the time required for the upstream propagating wave to reach that point from the down stream boundary, τ_L, can be estimated by the following relation:

$$\begin{aligned} x_L &= -\int_0^{t_L} (u_{ref} - c)dt \\ \frac{x_L}{x_{ref}} &= -(1 - \sqrt{1 + \frac{\beta}{u_{ref}^2}})\tau_L \end{aligned} \tag{3.8}$$

On the other hand, the time scale, τ_δ, for the viscous effect to spread for a distance, δ, can be estimated as follows:

$$\frac{\delta}{x_{ref}} \simeq \frac{2\sqrt{\nu t}}{x_{ref}} = 2\sqrt{\frac{\nu}{u_{ref}x_{ref}} \frac{tu_{ref}}{x_{ref}}} = 2\sqrt{\frac{\tau_\delta}{Re}} \tag{3.9}$$

In order for the viscous boundary layer to adjust to its new pressure environment properly and to avoid slow fluctuations of the separation region when it is present, it is required that

$$\tau_\delta \gg \tau_L$$

which leads to

$$\beta \gg [1 + \frac{4}{Re}(\frac{x_{ref}}{\delta})^2(\frac{x_L}{x_{ref}})]^2 - 1 \tag{3.10}$$

This gives an estimate of the lower bound of the pseudocompressibility parameter, β. The physical phenomena described above will be illustrated by numerical experiments later in this report.

3.2 Governing Equations in Generalized Coordinates

To perform calculations on 3-D arbitrarily shaped geometries, the following generalized independent variables are introduced which transform the physical coordinates into general curvilinear coordinates

$$\begin{aligned} \xi &= \xi(x, y, z, t) \\ \eta &= \eta(x, y, z, t) \\ \zeta &= \zeta(x, y, z, t) \end{aligned} \tag{3.11}$$

The Jacobian of the transformation is defined as

$$J = \det\frac{\partial(\xi,\eta,\zeta)}{\partial(x,y,z)} = \begin{vmatrix} \xi_x & \xi_y & \xi_z \\ \eta_x & \eta_y & \eta_z \\ \zeta_x & \zeta_y & \zeta_z \end{vmatrix} \tag{3.12}$$

where

$$\xi_x = \frac{\partial\xi}{\partial x},\ \eta_y = \frac{\partial\eta}{\partial y}$$

In actual coding, metric terms are calculated as follows :

$$\begin{pmatrix} \xi_x \\ \xi_y \\ \xi_z \end{pmatrix} = \frac{1}{J'}\begin{pmatrix} y_\eta z_\zeta - y_\zeta z_\eta \\ x_\zeta z_\eta - x_\eta z_\zeta \\ x_\eta y_\zeta - x_\zeta y_\eta \end{pmatrix}$$

$$\begin{pmatrix} \eta_x \\ \eta_y \\ \eta_z \end{pmatrix} = \frac{1}{J'}\begin{pmatrix} y_\zeta z_\xi - y_\xi z_\zeta \\ x_\xi z_\zeta - x_\zeta z_\xi \\ x_\zeta y_\xi - x_\xi y_\zeta \end{pmatrix} \tag{3.13}$$

$$\begin{pmatrix} \zeta_x \\ \zeta_y \\ \zeta_z \end{pmatrix} = \frac{1}{J'}\begin{pmatrix} y_\xi z_\eta - y_\eta z_\xi \\ x_\eta z_\xi - x_\xi z_\eta \\ x_\xi y_\eta - x_\eta y_\xi \end{pmatrix}$$

$$J' = \det\frac{\partial(x,y,z)}{\partial(\xi,\eta,\zeta)} = \begin{vmatrix} x_\xi & x_\eta & x_\zeta \\ y_\xi & y_\eta & y_\zeta \\ z_\xi & z_\eta & z_\zeta \end{vmatrix} \tag{3.14}$$

Applying the transformation to equation (2.1) and (2.2) yields

$$\frac{\partial}{\partial t}\hat{u} = -\frac{\partial}{\partial\xi}(\hat{e} - \hat{e}_v) - \frac{\partial}{\partial\eta}(\hat{f} - \hat{f}_v) - \frac{\partial}{\partial\zeta}(\hat{g} - \hat{g}_v) = -\hat{r} \tag{3.15}$$

$$\frac{\partial}{\partial\xi}\left(\frac{U - \xi_t}{J}\right) + \frac{\partial}{\partial\eta}\left(\frac{V - \eta_t}{J}\right) + \frac{\partial}{\partial\zeta}\left(\frac{W - \zeta_t}{J}\right) = 0 \tag{3.16}$$

where

$$\hat{u} = \frac{1}{J}\begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

$$\hat{e} = \frac{1}{J}\begin{bmatrix} \xi_x p + uU \\ \xi_y p + vU \\ \xi_z p + wU \end{bmatrix}$$

$$\hat{f} = \frac{1}{J}\begin{bmatrix} \eta_x p + uV \\ \eta_y p + vV \\ \eta_z p + wV \end{bmatrix} \tag{3.17}$$

$$\hat{g} = \frac{1}{J}\begin{bmatrix} \zeta_x p + uW \\ \zeta_y p + vW \\ \zeta_z p + wW \end{bmatrix}$$

$$U = \xi_t + \xi_x u + \xi_y v + \xi_z w$$

$$V = \eta_t + \eta_x u + \eta_y v + \eta_z w$$

$$W = \zeta_t + \zeta_x u + \zeta_y v + \zeta_z w$$

The viscous terms are quite lengthy and therefore given here separately.

$$\begin{aligned} \frac{\partial}{\partial x_j}\tau_{ij} &= \frac{\partial}{\partial x_j}\nu S_{ij} \\ &= \frac{\partial}{\partial x}\begin{bmatrix} u_x + u_x \\ v_x + u_y \\ w_x + u_z \end{bmatrix} + \frac{\partial}{\partial y}\begin{bmatrix} u_y + v_x \\ v_y + v_y \\ w_y + v_z \end{bmatrix} + \frac{\partial}{\partial z}\begin{bmatrix} u_z + w_x \\ v_z + w_y \\ w_z + w_z \end{bmatrix} \end{aligned}$$

where

$$u_x = \frac{\partial u}{\partial x},\ u_y = \frac{\partial u}{\partial y}$$

When ν is constant, the contribution of the second term in each bracket sums up to be zero when the velocity field is divergence free. However, since ν varies in space and time in general, theses terms have to be kept. Then the viscous terms in transformed coordinates are given by

$$\hat{e}_v = \frac{\nu}{J}\left\{ \nabla\xi \cdot (\nabla\xi\frac{\partial}{\partial\xi} + \nabla\eta\frac{\partial}{\partial\eta} + \nabla\zeta\frac{\partial}{\partial\zeta}) \begin{bmatrix} u \\ v \\ w \end{bmatrix} + (\xi_x\frac{\partial u}{\partial\xi_i} + \xi_y\frac{\partial v}{\partial\xi_i} + \xi_z\frac{\partial w}{\partial\xi_i}) \begin{bmatrix} \frac{\partial}{\partial x}\xi_i \\ \frac{\partial}{\partial y}\xi_i \\ \frac{\partial}{\partial z}\xi_i \end{bmatrix} \right\}$$

$$\hat{f}_v = \frac{\nu}{J}\left\{ (\nabla\eta \cdot (\nabla\xi\frac{\partial}{\partial\xi} + \nabla\eta\frac{\partial}{\partial\eta} + \nabla\zeta\frac{\partial}{\partial\zeta}) \begin{bmatrix} u \\ v \\ w \end{bmatrix} + (\eta_x\frac{\partial u}{\partial\xi_i} + \eta_y\frac{\partial v}{\partial\xi_i} + \eta_z\frac{\partial w}{\partial\xi_i}) \begin{bmatrix} \frac{\partial}{\partial x}\xi_i \\ \frac{\partial}{\partial y}\xi_i \\ \frac{\partial}{\partial z}\xi_i \end{bmatrix} \right\}$$

$$\hat{g}_v = \frac{\nu}{J} \left\{ \nabla\zeta \cdot (\nabla\xi \frac{\partial}{\partial\xi} + \nabla\eta \frac{\partial}{\partial\eta} + \nabla\zeta \frac{\partial}{\partial\zeta}) \begin{bmatrix} u \\ v \\ w \end{bmatrix} + (\zeta_x \frac{\partial u}{\partial\xi_i} + \zeta_y \frac{\partial v}{\partial\xi_i} + \zeta_z \frac{\partial w}{\partial\xi_i}) \begin{bmatrix} \frac{\partial}{\partial x}\xi_i \\ \frac{\partial}{\partial y}\xi_i \\ \frac{\partial}{\partial z}\xi_i \end{bmatrix} \right\} \tag{3.18a}$$

The above full viscous terms can be simplified under special conditions. If an orthogonal coordinate system is used, then

$$\nabla\xi_i \cdot \nabla\xi_j = 0 \quad ; \quad \text{for}\, i \neq j$$

For constant ν (i.e., laminar flow), the contribution by the second term in parentheses in the above equation will approach zero as the steady state is reached, where equation (2.1) is numerically satisfied. Therefore, for flow with constant ν in orthogonal coordinates, equation (3.18a) can be reduced as follows:

$$\begin{aligned} \hat{e}_v &= (\frac{\nu}{J})({\xi_x}^2 + {\xi_y}^2 + {\xi_z}^2) \begin{bmatrix} u_\xi \\ v_\xi \\ w_\xi \end{bmatrix} \\ \hat{f}_v &= (\frac{\nu}{J})({\eta_x}^2 + {\eta_y}^2 + {\eta_z}^2) \begin{bmatrix} u_\eta \\ v_\eta \\ w_\eta \end{bmatrix} \\ \hat{g}_v &= (\frac{\nu}{J})({\zeta_x}^2 + {\zeta_y}^2 + {\zeta_z}^2) \begin{bmatrix} u_\zeta \\ v_\zeta \\ w_\zeta \end{bmatrix} \end{aligned} \tag{3.18b}$$

3.3 Steady-State Formulation

The pseudocompressibility relation is introduced by adding a time derivative of pressure to the continuity equation, resulting in

$$\frac{1}{\beta}\frac{\partial p}{\partial\tau} + \frac{\partial u_i}{\partial\xi_i} = 0 \tag{3.19}$$

In the steady-state formulation the equations are to be marched in a time-like fashion until the divergence of velocity in equation (3.19) converges to zero. The time variable for this process no longer represents physical time, so in the momentum equations t is replaced with τ, which can be thought of as a pseudo-time or iteration parameter. Combining equation (3.19) with the momentum equations gives the following system of equations:

$$\frac{\partial}{\partial\tau}\hat{D} = -\frac{\partial}{\partial\xi}(\hat{E} - \hat{E}_v) - \frac{\partial}{\partial\eta}(\hat{F} - \hat{F}_v) - \frac{\partial}{\partial\zeta}(\hat{G} - \hat{G}_v) = -\hat{R} \tag{3.20}$$

where $\hat{R}$ is the right-hand-side of the momentum equation and can be defined as the residual for the steady-state computations, and where

$$\begin{aligned}
\hat{D} &= \frac{D}{J} = \frac{1}{J}\begin{bmatrix} p \\ u \\ v \\ w \end{bmatrix} \\
\hat{E} &= \begin{bmatrix} \beta(U-\xi_t)/J \\ \hat{e} \end{bmatrix} = \frac{1}{J}\begin{bmatrix} \beta(U-\xi_t) \\ \xi_x p + uU \\ \xi_y p + vU \\ \xi_z p + wU \end{bmatrix} \\
\hat{F} &= \begin{bmatrix} \beta(V-\eta_t)/J \\ \hat{f} \end{bmatrix} = \frac{1}{J}\begin{bmatrix} \beta(V-\eta_t) \\ \eta_x p + uV \\ \eta_y p + vV \\ \eta_z p + wV \end{bmatrix} \\
\hat{G} &= \begin{bmatrix} \beta(W-\zeta_t)/J \\ \hat{g} \end{bmatrix} = \frac{1}{J}\begin{bmatrix} \beta(W-\zeta_t) \\ \zeta_x p + uW \\ \zeta_y p + vW \\ \zeta_z p + wW \end{bmatrix} \\
\hat{E}_v &= \begin{bmatrix} 0 \\ \hat{e}_v \end{bmatrix} \\
\hat{F}_v &= \begin{bmatrix} 0 \\ \hat{f}_v \end{bmatrix} \\
\hat{G}_v &= \begin{bmatrix} 0 \\ \hat{g}_v \end{bmatrix}
\end{aligned} \tag{3.21}$$

For flow with constant ν in orthogonal coordinates,

$$\begin{aligned}
\hat{E}_v &= \left(\frac{\nu}{J}\right)\left({\xi_x}^2 + {\xi_y}^2 + {\xi_z}^2\right) I_m \frac{\partial D}{\partial \xi} = \gamma_1 D \\
\hat{F}_v &= \left(\frac{\nu}{J}\right)\left({\eta_x}^2 + {\eta_y}^2 + {\eta_z}^2\right) I_m \frac{\partial D}{\partial \eta} = \gamma_2 D \\
\hat{G}_v &= \left(\frac{\nu}{J}\right)\left({\zeta_x}^2 + {\zeta_y}^2 + {\zeta_z}^2\right) I_m \frac{\partial D}{\partial \zeta} = \gamma_3 D
\end{aligned} \tag{3.18c}$$

where

$$I_m = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3.4 Steady-State Algorithm Using Approximate Factorization

3.4.1 Difference Equations

There are a number of different ways of defining variables in a grid system. Either standard cell-node oriented or a staggered arrangement can be chosen. In Cartesian coordinates, a staggered grid has several favorable properties. In generalized coordinates, these advantages become obscured because of the interpolation required. However, a fully conservative differencing scheme can be devised which maintains the convenience of a staggered arrangement in a Poisson solver (Rosenfeld et al., 1988). Using any grid system, spatial differencing can be done either in finite-difference (Steger and Kutler, 1977) or finite-volume form (Yoon and Kwak, 1988). The finite-volume scheme usually behaves better near geometric singularities. In spacial differencing, both central differencing and upwind differencing have been tried. In this section, the work done using central differencing will be discussed first. Most of the results presented later in this report were obtained using finite-difference schemes. This chapter will focus on algorithms based on a finite-difference approach.

The pseudo-time derivative is replaced by a trapezoidal rule finite-differencing scheme resulting in

$$\hat{D}^{n+1} = \hat{D}^n + \frac{\Delta\tau}{2}\left[\left(\frac{\partial\hat{D}}{\partial\tau}\right)^n + \left(\frac{\partial\hat{D}}{\partial\tau}\right)^{n+1}\right] + O(\Delta\tau^3) \tag{3.22}$$

where the superscript n refers to the nth pseudo-time iteration level. By substituting equation (3.20) into equation (3.22), one obtains

$$\begin{aligned} D^{n+1} &+ \frac{\Delta\tau}{2}J\left[\delta_\xi(\hat{E}-\hat{E}_v)^{n+1} + \delta_\eta(\hat{F}-\hat{F}_v)^{n+1} + \delta_\zeta(\hat{G}-\hat{G}_v)^{n+1}\right] \\ &= D^n - \frac{\Delta\tau}{2}J\left[\delta_\xi(\hat{E}-\hat{E}_v)^{n} + \delta_\eta(\hat{F}-\hat{F}_v)^{n} + \delta_\zeta(\hat{G}-\hat{G}_v)^{n}\right] \end{aligned} \tag{3.23}$$

The problem is to solve for D^{n+1}, and this is nonlinear in nature since $\hat{E}^{n+1} = \hat{E}(D^{n+1})$ is a nonlinear function of D^{n+1} as are $\hat{F}^{n+1}$ and $\hat{G}^{n+1}$. The following linearization procedure is used. A local Taylor expansion about u^n yields

$$\begin{aligned} \hat{E}^{n+1} &= \hat{E}^n + \hat{A}^n(D^{n+1} - D^n) + O(\Delta\tau^2) \\ \hat{F}^{n+1} &= \hat{F}^n + \hat{B}^n(D^{n+1} - D^n) + O(\Delta\tau^2) \\ \hat{G}^{n+1} &= \hat{G}^n + \hat{C}^n(D^{n+1} - D^n) + O(\Delta\tau^2) \end{aligned} \tag{3.24}$$

where $\hat{A}$, $\hat{B}$, and $\hat{C}$ are the Jacobian matrices

$$\hat{A} = \frac{\partial \hat{E}}{\partial D}, \ \hat{B} = \frac{\partial \hat{F}}{\partial D}, \ \hat{C} = \frac{\partial \hat{G}}{\partial D} \tag{3.25}$$

The Jacobian matrices can all be represented by the following,

$$\hat{A}_i = \frac{1}{J}\begin{bmatrix} 0 & L_1\beta & L_2\beta & L_3\beta \\ L_1 & Q + L_1 u & L_2 u & L_3 u \\ L_2 & L_1 v & Q + L_2 v & L_3 v \\ L_3 & L_1 w & L_2 w & Q + L_3 w \end{bmatrix} \tag{3.26}$$

where $\hat{A}_i = \hat{A}, \hat{B}, \text{or } \hat{C}$ for $i = 1, 2, \text{or } 3$, respectively, and

$$\begin{aligned} &Q = L_0 + L_1 u + L_2 v + L_3 w \\ &L_0 = (\xi_i)_t, \ L_1 = (\xi_i)_x, \ L_2 = (\xi_i)_y, \ L_3 = (\xi_i)_z \\ &\xi_i = (\xi, \eta, \text{ or } \zeta) \text{ for } (\hat{A}, \hat{B}, \text{ or } \hat{C}) \end{aligned}$$

Substituting equation (3.24) into equation (3.23) results in the governing equation in delta form

$$\begin{aligned} &\left\{ I + \frac{h}{2} J \left[\delta_\xi(\hat{A}^n - \Gamma_1) + \delta_\eta(\hat{B}^n - \Gamma_2) + \delta_\zeta(\hat{C}^n - \Gamma_3) \right] \right\} (D^{n+1} - D^n) \\ &= -\Delta\tau J \left[\delta_\xi(\hat{E} - \hat{E}_v)^n + \delta_\eta(\hat{F} - \hat{F}_v)^n + \delta_\zeta(\hat{G} - \hat{G}_v)^n \right] \end{aligned} \tag{3.27}$$

where

$$\begin{aligned} \Gamma_1 D^{n+1} &= (\frac{\nu}{J}) \nabla\xi \cdot (\nabla\xi \frac{\partial}{\partial\xi} + \nabla\eta \frac{\partial}{\partial\eta} + \nabla\zeta \frac{\partial}{\partial\zeta}) I_m \frac{\partial D}{\partial \xi} \\ h &= \Delta\tau \quad \text{for trapezoidal differencing} \\ &= 2\Delta\tau \quad \text{for Euler implicit scheme} \end{aligned}$$

At this point it should be noted that the notation of the form $[\delta_\xi(A - \Gamma)]D$ refers to $\frac{\partial}{\partial\xi}(AD) - \frac{\partial}{\partial\xi}(\Gamma D)$ and not $\frac{\partial A}{\partial\xi}D - \frac{\partial\Gamma}{\partial\xi}D$.

3.4.2 Approximate Factorization

ADI scheme:

The solution of equation (3.27) would involve a formidable matrix inversion problem. With the use of an alternating direction implicit (ADI) type scheme, the problem could be reduced to the inversion of three matrices of small bandwidth, for which there exist some efficient solution algorithms. The particular ADI form used here is known as approximate factorization (AF) (Beam and Warming, 1978). It is difficult to apply the AF scheme to equation (3.27) in its full matrix form. Noting that at the steady-state the left-hand side of equation (3.27) approaches zero, a simplified expression for the viscous term as shown in equation (3.18b) is used on the left-hand side. To maintain the accuracy of the solution, the entire viscous term is used on the right-hand side. Using these terms, the governing equation becomes

$$L_\xi L_\eta L_\zeta (D^{n+1} - D^n) = RHS \tag{3.28a}$$

where

$$\begin{aligned} L_\xi &= \left[I + \frac{\Delta\tau}{2} J^{n+1} \delta_\xi (\hat{A}^n - \gamma_1)\right] \\ L_\eta &= \left[I + \frac{\Delta\tau}{2} J^{n+1} \delta_\eta (\hat{B}^n - \gamma_2)\right] \\ L_\zeta &= \left[I + \frac{\Delta\tau}{2} J^{n+1} \delta_\zeta (\hat{C}^n - \gamma_3)\right] \end{aligned} \tag{3.28b}$$

and RHS is the right-hand side of equation (3.27). When second-order central differencing is used, the solution to this problem becomes the inversion of three block tridiagonal matrices. The inversion problem is reduced to the three inversions

$$\begin{aligned} (L_\eta)\Delta\bar{D} &= RHS \\ (L_\xi)\Delta\bar{\bar{D}} &= \Delta\bar{D} \\ (L_\zeta)\Delta D^{n+1} &= \Delta\bar{\bar{D}} \end{aligned} \tag{3.29}$$

These inversions are carried out for all interior points, and the boundary conditions can be implemented explicitly. It is possible, however, to implement the boundary conditions implicitly, which will be discussed in a later section.

A guideline for estimating the lower bound of β was given by equation (3.10) which was derived from physical reasoning. Following the guideline, it is desirable to choose β as large as possible to make the pressure wave faster. There is, however, a bound of β which comes from the particular algorithm chosen here, namely, the error introduced by the approximate factorization. In implementing the AF scheme leading to equation (3.28), the following second-order cross product terms are introduced into the equation:

$$\left(\frac{\Delta\tau}{2} J^{n+1}\right)^2 \left[\delta_\xi(\hat{A}^n - \Gamma_1)\delta_\eta(\hat{B}^n - \Gamma) + \ldots\right]$$

This term must be kept smaller than the original terms in the equation. Including only the terms which contain β, this restriction can be expressed as

$$\frac{\Delta\tau}{2} J^{n+1} \delta_{\xi_i} \hat{A}_i^n \delta_{\xi_j} \hat{A}_j^n < \delta_{\xi_i} \hat{A}_i^n, \quad i \neq j$$

or

$$\frac{\Delta\tau}{2}J^{n+1}\delta_{\xi_j}\hat{A}_j^n < 1$$

Recalling the expression for $\hat{A}_i^n$ given by equation (3.26), the terms which have β in them give the following

$$\frac{\Delta\tau}{2}\beta\delta_{\xi_j}\left(\frac{\partial\xi_j}{\partial x_i}\right) < 1$$

The term to the right of β in this inequality is essentially the change in $\frac{1}{\Delta x_i}$ in either the $\xi, \eta,$ or ζ direction. An estimate of the order of magnitude of this term for the grids generally used is given by

$$O\left[\delta_{\xi_j}\left(\frac{\partial\xi_j}{\partial x_i}\right)\right] \approx 2$$

which puts the restriction on β

$$O(\beta\Delta\tau) < 1 \tag{3.30}$$

For most problems, the restrictions for β given by equations (3.10) and (3.30) are satisfied with a value for β in the range from 1 to 10. As will be shown later, this restriction on the upper value of β can be relaxed if the factorization error involving β is removed, for example, by a line relaxation scheme.

In applying the present approximate factorization scheme, it has been found that the stability of the scheme is dependent on the use of some higher-order smoothing terms. These are used to damp out higher frequency oscillations which arise in the solution because of the use of second-order central differencing, and will be discussed later.

LU-SGS scheme:

Recently, Yoon and Jameson (1987) developed an implicit lower-upper symmetric-Gauss-Seidel (LU-SGS) scheme for the compressible Euler and Navier-Stokes equations. A similar scheme is devised for the pseudocompressible formulation (Yoon and Kwak, 1989). This LU-SGS scheme is not only unconditionally stable but completely vectorizable in three dimensions. Spacial differencing can be either central or upwind depending on the numerical dissipation model which augments the finite volume method (Yoon and Kwak, 1988). This scheme is briefly described below.

Starting from an unfactored implicit scheme similar to equation (3.27)

$$\begin{aligned}&\left\{I + \frac{h}{2}\left[\delta_\xi\hat{A} + \delta_\eta\hat{B} + \delta_\zeta\hat{C}\right]\right\}(D^{n+1} - D^n)\\ &\quad = -\Delta t\left[\delta_\xi(\hat{E} - \hat{E}_v) + \delta_\eta(\hat{F} - \hat{F}_v) + \delta_\zeta(\hat{G} - \hat{G}_v)\right]\end{aligned} \tag{3.27'}$$

The LU-SGS implicit factorization scheme can be derived as

$$L_l {L_d}^{-1} L_u = RHS \tag{3.31a}$$

where

$$\begin{aligned} L_l &= I + \frac{h}{2}(\delta^-{}_\xi \hat{A}^+ + \delta^-{}_\eta \hat{B}^+ + \delta^-{}_\zeta \hat{C}^+ - \hat{A}^- - \hat{B}^- - \hat{C}^-) \\ L_d &= I + \frac{h}{2}(\hat{A}^+ - \hat{A}^+ + \hat{B}^+ - \hat{B}^- + \hat{C}^+ - \hat{C}^-) \\ L_u &= I + \frac{h}{2}(\delta^+{}_\xi \hat{A}^- + \delta^+{}_\eta \hat{B}^- + \delta^+{}_\zeta \hat{C}^- + \hat{A}^+ + \hat{B}^+ + \hat{C}^+) \end{aligned} \tag{3.31b}$$

where $\delta^-{}_\xi$ and $\delta^+{}_\xi$ are the backward- and forward-difference operators respectively. In the framework of the LU-SGS algorithm, a variety of schemes can be developed by different choices of numerical dissipation models and Jacobian matrices of the flux vectors. Further details of this method will be given in a later report (Yoon and Kwak, 1989).

3.4.3 Numerical Dissipation or Smoothing

Higher-order smoothing terms are required to make the present algorithm stable. These terms help to damp out the higher-order oscillations and odd- and even-point decoupling in the solution which are caused by the use of central differencing. The smoothing term can be related to an upwind finite-difference approximation. The idea of splitting the upwinding scheme into the central differencing plus dissipation was successfully implemented into the first- and the third-order dissipation model by Jameson et al. (1981). Pulliam (1986) discussed an implicit dissipation model extensively. More recently Yoon and Kwak (1988) unified the dissipation models in the framework of a finite volume total variation diminishing (TVD) method. These models are being extended into the pseudocompressible formulation (Yoon and Kwak, 1989). Here, only those specifics are discussed that are relevant to the constant coefficient model, which was originally used in conjunction with the pseudocompressible formulation.

By including these smoothing terms, equations (3.28a) and (3.28b) become

$$L_\xi L_\eta L_\zeta (D^{n+1} - D^n) = \text{RHS of (3.27)} - \epsilon_e[(\nabla_\xi \Delta_\xi)^2 + (\nabla_\eta \Delta_\eta)^2 + (\nabla_\zeta \Delta_\zeta)^2] D^n \tag{3.28}$$

where

$$\begin{aligned} L_\xi &= \left[I + \frac{\Delta\tau}{2} J^{n+1} \delta_\xi (\hat{A}_1^n - \gamma_1) + \epsilon_i \nabla_\xi \Delta_\xi \right] \\ L_\eta &= \left[I + \frac{\Delta\tau}{2} J^{n+1} \delta_\eta (\hat{A}_2^n - \gamma_2) + \epsilon_i \nabla_\eta \Delta_\eta \right] \\ L_\zeta &= \left[I + \frac{\Delta\tau}{2} J^{n+1} \delta_\zeta (\hat{A}_3^n - \gamma_3) + \epsilon_i \nabla_\zeta \Delta_\zeta \right] \end{aligned}$$

Here, ∇ and Δ represent forward and backward spacial-differencing operators, respectively. To preserve the tridiagonal nature of the system, only second-order smoothing can be used on the left-hand side of the equation, whereas fourth-order smoothing is used on the right-hand side. When the diagonal algorithm (described in a later section) is used, however, it is feasible to increase the bandwidth of the system to a pentadiagonal. This makes it possible to use fourth-order smoothing on the left-hand side of the equations also. The AF algorithm will be stable if ϵ_i and ϵ_e satisfy a certain relation (see Pulliam, 1986; Jameson and Yoon, 1986) as discussed below.

To study the nature of the numerical smoothing, a 1-D form of the dissipation terms is represented as follows:

$$[1 - \epsilon_i \nabla_\xi \Delta_\xi](p^{n+1} - p^n) = -\epsilon_e(\nabla_\xi \Delta_\xi)^2 p^n \tag{3.32}$$

Suppose p is represented by the discrete Fourier expansion

$$p = \sum_n \hat{p}(k) e^{ik\xi} \tag{3.33}$$

where

$$\begin{aligned} \hat{p} &= \text{Fourier transform of p} \\ k &= \frac{2\pi}{N\Delta\xi} n = \text{wave number} \\ n &= -N/2, ...0, 1, ..(N/2 - 1) \\ N &= \text{number of mesh points} \end{aligned}$$

Equation (3.32) can be written as

$$[1 - \epsilon_i k'](\hat{p}^{n+1} - \hat{p}^n) = -\epsilon_e (k')^2 \hat{p}^n \tag{3.34}$$

where

$$\begin{aligned} k' &= -2 + 2cos(k) \\ (k')^2 &= 6 - 8cos(k) + 2cos(2k) \end{aligned}$$

From this, the amplification factor can be defined as

$$\sigma = \frac{\hat{p}^{n+1}}{\hat{p}^n} = \frac{[1 - \epsilon_i k' - \epsilon_e (k')^2]}{[1 - \epsilon_i k']} \tag{3.35}$$

To damp out the numerical fluctuations as time advances, the absolute value of the amplification factor σ has to be less than one for all possible frequencies, i.e.,

$$|\sigma| < 1$$

Noting that k' is always negative, this requirement leads to the following relation.

$$\epsilon_e \leq 2(1 - \epsilon_i k') \tag{3.36a}$$

It can be shown that the above inequality is always satisfied if

$$2\epsilon_e \leq \epsilon_i \tag{3.36b}$$

The exact relation between these two coefficients can be determined only by a nonlinear stability analysis. In the present study, ϵ_i is taken to be three times larger than ϵ_e. From the expression given in equation (3.35), it is clear that if ϵ_i is too large, the rate of damping will be diminished. It may not be advantageous to take a very large value for ϵ_i over ϵ_e. The choice of ϵ_e depends on the Reynolds number and the grid spacings. However, as discussed later, large values of ϵ_e adversely affect the accuracy of the continuity equation. Therefore, the magnitude of ϵ_e is usually taken to be small. If grid sizes are fine enough to resolve the changes in the flow field, it can be taken as small as 10^{-3}.

There are two major sources of inaccuracy associated with the numerical dissipation terms; namely, (1) the numerical dissipation terms effectively change the Reynolds number of the flow, and (2) the explicit smoothing terms added to the continuity equation do not conserve mass. In particular, the explicit smoothing on the pressure can affect whether or not the solution converges to an incompressible flow field. Chang and Kwak (1984) showed that the pseudo-pressure waves decay exponentially with time, and vanish as the solution converges. Thus the change in pressure with time approaches zero. When there is no explicit smoothing added to the equation, the divergence of the velocity field identically approaches zero. However, when explicit smoothing is included, as the change in pressure approaches zero, the divergence of the velocity approaches

$$\frac{\delta u_i}{\delta x_i} \rightarrow -\frac{\epsilon_{e1}}{\beta \Delta\tau}\left[(\nabla_\xi \Delta_\xi)^2 + (\nabla_\eta \Delta_\eta)^2 + (\nabla_\zeta \Delta_\zeta)^2\right] p \tag{3.37}$$

where $\frac{\delta}{\delta x_i}$ is a difference form of the divergence operator and ϵ_{e1} is the explicit smoothing parameter for the pressure. If ϵ_e is scaled by h, i.e., $\epsilon_e = \Delta\tau\varepsilon_e$, equation (3.37) becomes independent of the time step. Depending on the magnitude of β and the local pressure gradient, this term can deteriorate the conservation of mass. However, it is to be noted that, in the numerical computations, the central differencing scheme is modified to include numerical dissipation terms resulting in what is essentially an upwinding scheme. Therefore, in the present approach, the mass conservation is satisfied in an upwinding sense. In numerical experiments, we have seen that the divergence of the velocity field evaluated using second order differences does not decrease as fast as the residual terms which can be regarded as the divergence of the velocity in an upwinding sense.

3.4.4 Diagonal Algorithm

In a diagonal algorithm, a similarity transform is used to diagonalize the Jacobian matrices and uncouple the set of equations. The equations can then be solved by solving scalar tridiagonal matrices instead of solving block tridiagonal matrices. A similarity transform which symmetrizes and diagonalizes the matrices of the compressible gas dynamic equations has been used by Warming et al. (1975) and Turkel (1973). This method was used by Pulliam and Chaussee (1981) to produce a diagonal algorithm for the Euler equations. This method can be applied to the compressible Navier-Stokes equations to obtain a considerable savings in computing time (Flores, 1985). In this section, similarity transforms for the matrices used in the method of pseudocompressibility are presented. They are used in a diagonal algorithm, which results in a substantial reduction in computer time (Rogers et al., 1987).

Similarity transformations exist which diagonalize the Jacobian matrices

$$\hat{A}_i = T_i \hat{\Lambda}_i T_i^{-1} \tag{3.38}$$

where $\hat{\Lambda}_i$ is a diagonal matrix whose elements are the eigenvalues of the Jacobian matrices and which is given by

$$\hat{\Lambda}_i = \begin{bmatrix} Q & 0 & 0 & 0 \\ 0 & Q & 0 & 0 \\ 0 & 0 & Q - L_0/2 + c & 0 \\ 0 & 0 & 0 & Q - L_0/2 - c \end{bmatrix} \tag{3.39}$$

and where c is the pseudospeed of sound, which is given by

$$c = \sqrt{(Q + L_0/2)^2 + \beta(L_1^2 + L_2^2 + L_3^2)} \tag{3.40}$$

The T_i matrix is composed of the eigenvectors of the Jacobian matrix. For more details on the derivation of T_i, its inverse and eigenvectors, see Rogers et al. (1987).

The implementation of the diagonal scheme involves replacing the Jacobian matrices in the implicit operators with the product of the similarity transform matrices and the diagonal matrix as given in equation (3.38). The identity matrix in the implicit operators is replaced by the product of the similarity transform matrix and its inverse. A modification is made to the implicit viscous terms by replacing the I_m matrix with an identity matrix so that the transformation matrices may also be factored out of these terms. This implicitly adds an additional viscous dissipation term to the pressure. The transformation matrices are now factored out of the implicit operators to give

$$\begin{aligned}
\mathcal{L}_\xi &= T_\xi[I + \frac{\Delta\tau}{2} J\delta_\xi(\Lambda_\xi - \hat{\gamma}_1)]T_\xi^{-1} \\
\mathcal{L}_\eta &= T_\eta[I + \frac{\Delta\tau}{2} J\delta_\eta(\Lambda_\eta - \hat{\gamma}_2)]T_\eta^{-1} \\
\mathcal{L}_\zeta &= T_\zeta[I + \frac{\Delta\tau}{2} J\delta_\zeta(\Lambda_\zeta - \hat{\gamma}_3)]T_\zeta^{-1}
\end{aligned} \tag{3.41}$$

where the implicit viscous terms are now given by

$$\hat{\gamma}_i = \frac{\nu}{J}\nabla\xi_i \cdot \nabla\xi_i I\delta_{\xi_i} \tag{3.42}$$

Since the transformation matrices are dependent on the metric quantities, factoring them outside the difference operators introduces an error. No modification has been made to the right-hand side of the equation and therefore these linearization errors will not affect the steady-state solution, only the convergence path toward the solution.

The implementation of this algorithm over the block algorithm will result in a substantial reduction in computational time per iteration because of the decrease in the number of operations performed. Additionally, considerably less memory is required to store the elements on the left-hand side. This additional memory was used to further vectorize the existing code as follows. Since the solution of a tridiagonal block or scalar matrix is recursive, it is not vectorizable for loops which use the current sweep direction as the inner do-loop index. However, if a large number of these matrices are passed into the inversion routines at once, then vectorization can take place in the 'nonsweep' direction.

3.4.5 Boundary Conditions

Once the flow solver is developed, a boundary condition procedure has to be devised to be compatible with the solution algorithm. Boundary conditions play an important part in determining the overall accuracy, the stability property, and the convergence speed of the solution. There are different types of boundaries encountered in numerical simulations which include solid surface, inflow and outflow, and far-field boundaries.

Solid Surface :

At a solid surface boundary, the usual no-slip condition is applied. In general the grid point adjacent to the surface will be sufficiently close so that constant pressure normal to the surface in the viscous boundary layer can be assumed. For a ζ = constant surface, this can be expressed as

$$\left(\frac{\partial p}{\partial \zeta}\right)_{L=1} = 0. \tag{3.43}$$

This boundary condition can be implemented either explicitly or implicitly. The implicit implementation, however, will enhance the stability of the code. This can be done during the ζ sweep by including the following in the matrix to be inverted.

$$I\Delta D_{j,k,1} + \hat{c}\Delta D_{j,k,2} = \hat{f} \tag{3.44}$$

where

$$\hat{c} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \hat{f} = \begin{bmatrix} p_{L=2} - p_{L=1} \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Inflow, Outflow and Far-field Conditions :

The inflow and outflow boundary conditions for an internal flow problem and the far-field boundary conditions for an external flow problem can be handled in much the same way. The incoming flow for both problems can be set to an appropriate constant as dictated by the problem. For example, at the inlet to a pipe, the pressure can be set to a constant and the velocity profile can be specified to be uniform. The downstream conditions, however, are the most difficult to provide. Simple upwind extrapolation is not well-posed. The best convergence rate is obtained if global mass is conserved. So to ensure the best results, the velocities and pressure are first updated using a second-order upwind extrapolation. For an exit at L = LMAX this is written

$$Q^{\hat{n}}_{lmax} = \frac{Q^{n+1}_{lmax-1}\left(\frac{\Delta z_2}{\Delta z_1}\right) - Q^{n+1}_{lmax-2}}{\frac{\Delta z_2}{\Delta z_1} - 1} \tag{3.45}$$

where

$$\Delta z_1 = z_{lmax} - z_{lmax-1}$$

$$\Delta z_2 = z_{lmax} - z_{lmax-2}$$

Then, these extrapolated velocities are integrated over the exit boundary to obtain the outlet mass flux

$$\dot{m}_{out} = \int_{exit} \vec{V}^{\hat{n}} \cdot d\hat{a} \tag{3.46}$$

Then the velocity components are weighted by the mass flux ratio to conserve global mass

$$\vec{V}^{n+1} = \frac{\dot{m}_{in}}{\dot{m}_{out}} \vec{V}^{\hat{n}} \tag{3.47}$$

If nothing further is done to update the boundary pressure, this can lead to discontinuities in the pressure because momentum is not being conserved. A method of weighting the pressure by a momentum correction was presented by Chang et al. (1985), where the pressure condition is obtained by the mass weighted velocities

$$p^{n+1} = p^{\hat{n}} - \frac{1}{\zeta_z}\left[(wW)^{n+1} - (wW)^{\hat{n}}\right] + \frac{\nu}{\zeta_z}(\nabla\zeta \cdot \nabla\zeta)\left[\left(\frac{\partial w}{\partial \zeta}\right)^{n+1} - \left(\frac{\partial w}{\partial \zeta}\right)^{\hat{n}}\right] \tag{3.48}$$

where W is the contravariant velocity. In obtaining this formula, it has been assumed that the streamlines near the exit plane are nearly straight. Any appreciable deviation will cause a discontinuity in the pressure and may lead to an instability. To avoid this, a momentum weighted pressure was used. This was obtained by integrating the momentum corrected pressure p^{n+1} and the extrapolated pressure $p^{\hat{n}}$ across the exit

$$\begin{aligned} I_p^{n+1} &= \int_{exit} p^{n+1} d\hat{a} \\ I_p^{\hat{n}} &= \int_{exit} p^{\hat{n}} d\hat{a} \end{aligned} \tag{3.49}$$

The final outlet pressure is then taken to be

$$p^{n+1} = \left(\frac{I_p^{n+1}}{I_p^{\hat{n}}}\right) p^{\hat{n}} \tag{3.50}$$

Under these downstream boundary conditions, global conservation of mass and momentum are ensured. Many practical applications have been solved using the current procedure. However, the nonreflecting type boundary conditions according to Rudy and Skrikwerda (1980) may enhance the convergence speed.

3.5 Time-Accurate Formulation

Time dependent calculation of incompressible flows are especially time consuming due to the elliptic nature of the governing equations. Physically, this means that any local change in the flow has to be felt by the entire flow field. Numerically, this means that in each time step, the pressure field has to go through one complete steady-state iteration cycle, for example, by Poisson-solver-type pressure iteration or pseudocompressibility iteration method. In transient flow, the physical time step has to be small and consequently the change in the flow field may be small. In this situation, the number of iterations in each time step for getting a divergence-free flow field may not be as high as regular steady-state computations. However, the time-accurate computations are in general extremely time-consuming. Therefore, it is particularly desirable to develop computationally efficient methods either by implementing a fast algorithm and by utilizing computer characteristics such as vectorization and parallel processing. In this section, a time-accurate method using pseudocompressibility developed by Rogers and Kwak (1988, 1989) is described.

In this formulation the time derivatives in the momentum equations are differenced using a second-order, three-point, backward-difference formula

$$\frac{3\hat{u}^{n+1} - 4\hat{u}^{n} + \hat{u}^{n-1}}{2\Delta t} = -\hat{r}^{n+1} \tag{3.51}$$

where the superscript n denotes the quantities at time $t = n\Delta t$ and $\hat{r}$ is the right-hand side given in equation (3.15). To solve equation (3.51) for a divergence free velocity at the $n+1$ time level, a pseudo-time level is introduced and is denoted by a superscript m. The equations are iteratively solved such that $\hat{u}^{n+1,m+1}$ approaches the new velocity $\hat{u}^{n+1}$ as the divergence of $\hat{u}^{n+1,m+1}$ approaches zero. To drive the divergence of this velocity to zero, the following artificial compressibility relation is introduced:

$$\frac{p^{n+1,m+1} - p^{n+1,m}}{\Delta\tau} = -\beta\nabla\cdot\hat{u}^{n+1,m+1} \tag{3.52}$$

where τ denotes pseudo-time and β is an artificial compressibility parameter. Combining equation (3.52) with the momentum equations gives

$$\begin{aligned} &I_{t\tau}(\hat{D}^{n+1,m+1} - \hat{D}^{n+1,m}) \\ &= -\hat{R}^{n+1,m+1} - \frac{I_m}{\Delta t}(1.5\hat{D}^{n+1,m} - 2\hat{D}^{n} + 0.5\hat{D}^{n-1}) \end{aligned} \tag{3.53}$$

where $\hat{D}$ is the same vector defined in equation (3.21), $\hat{R}$ is the same residual vector defined in equation (3.20), $I_{t\tau}$ is a diagonal matrix, and I_m is a modified identity matrix given by

$$\begin{aligned} I_{t\tau} &= \text{diag}\left[\frac{1}{\Delta\tau}, \frac{1.5}{\Delta t}, \frac{1.5}{\Delta t}, \frac{1.5}{\Delta t}\right] \\ I_m &= \text{diag}[0,\ 1,\ 1,\ 1] \end{aligned}$$

Finally, the residual term at the *m+1* pseudo-time level is linearized giving the following equation in delta form

$$\begin{aligned}\left[\frac{I_{t\tau}}{J}+\left(\frac{\partial\hat{R}}{\partial D}\right)^{n+1,m}\right]&(D^{n+1,m+1}-D^{n+1,m})\\ &=-\hat{R}^{n+1,m}-\frac{I_m}{\Delta t}(1.5\hat{D}^{n+1,m}-2\hat{D}^n+0.5\hat{D}^{n-1})\end{aligned}\tag{3.54}$$

As can be seen, this equation is very similar to the steady-state formulation given by equation (3.27), which can be rewritten for the Euler implicit case as

$$\left[\frac{1}{J\Delta\tau}I+\left(\frac{\partial\hat{R}}{\partial D}\right)^n\right](D^{n+1}-D^n)=-\hat{R}^n \tag{3.55}$$

Both systems of equations will require the discretization of the same residual vector $\hat{R}$. The derivatives of the viscous fluxes in this vector are approximated using second-order central differences. The convective flux terms can be discretized using central differences as was done in section 3.4. This will require numerical dissipation terms for stability. Since, in pseudo compressible formulation, the governing equations are changed into the hyperbolic-parabolic type, some of the upwind differencing schemes which have recently been developed for the compressible Euler and Navier-Stokes equations by a number of authors (i.e., Roe, 1981; Chakravarthy and Osher, 1985; Steger and Warming, 1981; Harten et al 1983) can be utilized. In the present work, the method of Roe (1981) was used in differencing the convective terms by an upwind method that is biased by the signs of the eigenvalues of the local flux Jacobian. This is accomplished by casting the governing equations in their characteristic form and then forming the differencing stencil such that it accounts for the direction of wave propagation. In the current formulation, the set of numerical equations are solved using a nonfactored line relaxation scheme similar to that employed by MacCormack (1985). This implicit scheme described in the next section makes use of large amounts of processor memory (approximately 180 times the number of grid points) to be coded in an efficient manner.

3.6 Time-Accurate Algorithm Using Upwind Differencing

Even though the current method is explained for time-accurate calculations, the same method can be used for steady state problems.

3.6.1 Upwind Differencing

The upwind scheme for the convective flux derivatives is derived from 1-D theory, and then is applied to each of the coordinate directions separately. Flux-difference splitting is used here to structure the differencing stencil based on the sign of the eigenvalues of the convective flux Jacobian. The scheme presented here was originally derived by Roe (1981) as an approximate Riemann solver for the compressible gas dynamics equations.

The derivative of the convective flux in the ξ direction is approximated by

$$\frac{\partial \hat{E}}{\partial \xi} \approx \frac{[\bar{E}_{i+1/2} - \bar{E}_{i-1/2}]}{\Delta \xi} \tag{3.56}$$

where $\bar{E}_{i+1/2}$ is a numerical flux and i is the discrete spatial index for the ξ direction.

The numerical flux is given by

$$\bar{E}_{i+1/2} = \frac{1}{2}\left[\hat{E}(D_{i+1}) + \hat{E}(D_i) - \phi_{i+1/2}\right] \tag{3.57}$$

where the $\phi_{i+1/2}$ is a dissipation term. For $\phi_{i+1/2} = 0$ this represents a second-order central difference scheme. A first-order upwind scheme is given by

$$\phi_{i+1/2} = \left(\Delta E^+_{i+1/2} - \Delta E^-_{i+1/2}\right) \tag{3.58}$$

where $\Delta E^{\pm}$ is the flux difference across positive or negative traveling waves. The flux difference is computed as

$$\Delta E^{\pm}_{i+1/2} = A^{\pm}(\bar{D})\Delta D_{i+1/2} \tag{3.59}$$

where the Δ operator is given by

$$\Delta D_{i+1/2} = D_{i+1} - D_i \tag{3.60}$$

The plus (minus) Jacobian matrix has only positive or negative eigenvalues and is computed from

$$\begin{aligned} A^{\pm} &= X_1 \Lambda_1^{\pm} X_1^{-1} \\ \Lambda_1^{\pm} &= \frac{1}{2}(\Lambda_1 \pm |\Lambda_1|) \end{aligned} \tag{3.61}$$

where the subscript 1 denotes matrices corresponding to the ξ-direction flux. The matrices X_1 and X_1^{-1} are the right and left eigenvectors of the Jacobian matrix of the flux vector and Λ_1 is a diagonal matrix consisting of its eigenvalues. All matrices appearing in the upwind dissipation term must be evaluated at a half point (denoted by i+1/2). To do this a special averaging of the dependent variables at neighboring points must be performed.

The Roe properties, which are necessary for a conservative scheme, are satisfied if the following averaging procedure is employed

$$\bar{D} = \frac{1}{2}(D_{i+1} + D_i) \tag{3.62}$$

A scheme of arbitrary order may be derived using these flux differences. Implementation of higher-order-accurate approximations in an explicit scheme does not require significantly more computational time if the flux differences $\Delta E^{\pm}$ are all computed at once for a single line. A third-order upwind flux is then defined by

$$\phi_{i+1/2} = -\frac{1}{3}\left(\Delta E^{+}_{i-1/2} - \Delta E^{+}_{i+1/2} + \Delta E^{-}_{i+1/2} - \Delta E^{-}_{i+3/2}\right) \tag{3.63}$$

The primary problem with using schemes of accuracy greater than third-order occurs at the boundaries. Special treatment is needed, requiring a reduction of order or a much more complicated scheme. Therefore, when going to a higher-order-accurate scheme, compactness is desirable. Such a scheme was derived by Rai (1987) using a fifth-order-accurate, upwind-biased stencil. A fifth-order, fully upwind difference would require 11 points, but this upwind-biased scheme requires only seven points. It is given by

$$\begin{aligned}\phi_{i+1/2} = -\frac{1}{30}[& -2\Delta E^{+}_{i-3/2} + 11\Delta E^{+}_{i-1/2} - 6\Delta E^{+}_{i+1/2} - 3\Delta E^{+}_{i+3/2} \\ & +2\Delta E^{-}_{i+5/2} - 11\Delta E^{-}_{i+3/2} + 6\Delta E^{-}_{i+1/2} + 3\Delta E^{-}_{i-1/2}]\end{aligned} \tag{3.64}$$

Next to the boundary, near second-order accuracy can be maintained by the third- and fifth-order schemes by using the following

$$\phi_{i+1/2} = \epsilon\left(\Delta E^{+}_{i+1/2} - \Delta E^{-}_{i+1/2}\right) \tag{3.65}$$

For $\epsilon = 0$, this flux leads to a second-order central difference. For $\epsilon = 1$, this is the same as the first-order dissipation term given by equation (3.58). By including a nonzero ϵ, dissipation is added to the second-order, central-difference scheme to help suppress any oscillations. A value of $\epsilon = 0.01$ is commonly used for many applications.

To form the delta fluxes used in this scheme, the eigensystem of the convective flux Jacobian is needed. For the current formulation, a generalized flux vector is given by equation (3.21) and the Jacobian matrix $A_i = \frac{\partial \hat{E}_i}{\partial D}$ of the flux vector is given by equation (3.26). The normalized metrics are redefined as

$$k_x = \frac{1}{J}\frac{\partial \xi_i}{\partial x}, i = 1, 2, 3$$

$$k_y = \frac{1}{J}\frac{\partial \xi_i}{\partial y}, i = 1,2,3$$
$$k_z = \frac{1}{J}\frac{\partial \xi_i}{\partial z}, i = 1,2,3$$
$$k_t = \frac{1}{J}\frac{\partial \xi_i}{\partial t}, i = 1,2,3$$

As explained in conjunction with the diagonal algorithm in the previous section, a similarity transform for the Jacobian matrix is introduced here

$$\hat{A}_i = X_i \Lambda_i X_i^{-1}$$

where Λ_i and c are defined by equations (3.39) and (3.40) respectively. The matrix of the right eigenvectors is given by

$$X_i = \begin{bmatrix} 0 & 0 & \beta(c - k_t/2) & -\beta(c + k_t/2) \\ x_k & x_{kk} & u\lambda_3 + \beta k_x & u\lambda_4 + \beta k_x \\ y_k & y_{kk} & v\lambda_3 + \beta k_y & v\lambda_4 + \beta k_y \\ z_k & z_{kk} & w\lambda_3 + \beta k_z & w\lambda_4 + \beta k_z \end{bmatrix} \tag{3.66}$$

where

$$\begin{aligned} x_k &= \frac{\partial x}{\partial \xi_{i+1}} \\ y_k &= \frac{\partial y}{\partial \xi_{i+1}} \\ z_k &= \frac{\partial z}{\partial \xi_{i+1}} \\ x_{kk} &= \frac{\partial x}{\partial \xi_{i+2}} \\ y_{kk} &= \frac{\partial y}{\partial \xi_{i+2}} \\ z_{kk} &= \frac{\partial z}{\partial \xi_{i+2}} \end{aligned} \tag{3.67}$$

$$\xi_{i+1} = \eta, \zeta, \text{ or } \xi \text{ for } i = 1, 2, \text{ and } 3, \text{ respectively}$$
$$\xi_{i+2} = \zeta, \xi, \text{ or } \eta \text{ for } i = 1, 2, \text{ and } 3, \text{ respectively}$$

and its inverse can be obtained similarly. It should be noted that this transformation is nonsingular in any combination of metrics.

3.6.2 Implicit Scheme

This section describes the way in which equations (3.54) and (3.55) are numerically represented and solved. The first consideration is the formation of the Jacobian matrix of the residual vector $\hat{R}$ required for the implicit side of the equation. Applying the difference formula given in equation (3.56) to the convective flux vectors and applying a second-order, central-difference formula to the viscous terms, the residual at a discrete point $(x_{ijk}, y_{ijk}, z_{ijk})$ is given by

$$\begin{aligned}\hat{R}_{ijk} = {} & \frac{\bar{E}_{i+1/2,j,k} - \bar{E}_{i-1/2,j,k}}{\Delta\xi} + \frac{\bar{F}_{i,j+1/2,k} - \bar{F}_{i,j-1/2,k}}{\Delta\eta} + \frac{\bar{G}_{i,j,k+1/2} - \bar{G}_{i,j,k-1/2}}{\Delta\zeta} \\ & - \frac{(\hat{E}_v)_{i+1,j,k} - (\hat{E}_v)_{i-1,j,k}}{2\Delta\xi} - \frac{(\hat{F}_v)_{i,j+1,k} - (\hat{F}_v)_{i,j-1,k}}{2\Delta\eta} - \frac{(\hat{G}_v)_{i,j,k+1} - (\hat{G}_v)_{i,j,k-1}}{2\Delta\eta}\end{aligned} \tag{3.68}$$

The generalized coordinates are chosen so that $\Delta\xi$, $\Delta\eta$, and $\Delta\zeta$ are equal to one. To limit the bandwidth of the implicit system of equations, the Jacobian of the residual vector will be formed by considering only first-order contributions to the upwind numerical fluxes as well as the second-order differencing of the viscous terms. Hence, the only portion of the residual vector that is actually linearized is the following.

$$\begin{aligned}\hat{R}_{ijk} = {} & \frac{1}{2}(\hat{E}_{i+1,j,k} - \hat{E}_{i-1,j,k} + \ldots \\ & -\Delta E^{+}_{i+1/2,j,k} + \Delta E^{-}_{i+1/2,j,k} + \Delta E^{+}_{i-1/2,j,k} - \Delta E^{-}_{i-1/2,j,k} - \ldots \\ & -(\hat{E}_v)_{i+1/2,j,k} + (\hat{E}_v)_{i-1/2,j,k} - \ldots\end{aligned} \tag{3.69}$$

The exact Jacobian of this residual vector will result in a banded matrix of the form

$$\begin{aligned}\frac{\partial\hat{R}}{\partial D} = B\Bigg[& \frac{\partial\hat{R}_{ijk}}{\partial D_{i,j,k-1}}, 0, \ldots, 0, \frac{\partial\hat{R}_{ijk}}{\partial D_{i,j-1,k}}, 0, \ldots, 0, \\ & \frac{\partial\hat{R}_{ijk}}{\partial D_{i-1,j,k}}, \frac{\partial\hat{R}_{ijk}}{\partial D_{i,j,k}}, \frac{\partial\hat{R}_{ijk}}{\partial D_{i+1,j,k}}, \\ & 0, \ldots, 0, \frac{\partial\hat{R}_{ijk}}{\partial D_{i,j+1,k}}, 0, \ldots, 0, \frac{\partial\hat{R}_{ijk}}{\partial D_{i,j,k+1}}\Bigg]\end{aligned} \tag{3.70}$$

These exact Jacobians can be very costly to form. Therefore, approximate Jacobians of the flux differences as derived and analyzed by both Yee (1986) and Barth (1987) are used.

These are given as follows

$$
\begin{aligned}
\frac{\partial \hat{R}_{ijk}}{\partial D_{i,j,k-1}} &\approx \frac{1}{2}(-\hat{C}_{i,j,k-1} - C^{+}_{i,j,k-1/2} + C^{-}_{i,j,k-1/2}) - (\bar{\gamma}_3)_{i,j,k-1/2} \\
\frac{\partial \hat{R}_{ijk}}{\partial D_{i,j,k}} &\approx \frac{1}{2}(A^{+}_{i+1/2,j,k} + A^{+}_{i-1/2,j,k} - A^{-}_{i+1/2,j,k} - A^{-}_{i-1/2,j,k} \\
&\quad + B^{+}_{i,j+1/2,k} + B^{+}_{i,j-1/2,k} - B^{-}_{i,j+1/2,k} - B^{-}_{i,j-1/2,k} \\
&\quad + C^{+}_{i,j,k+1/2} + C^{+}_{i,j,k-1/2} - C^{-}_{i,j,k+1/2} - C^{-}_{i,j,k-1/2}) \\
&\quad + (\bar{\gamma}_1)_{i+1/2,j,k} + (\bar{\gamma}_2)_{i,j+1/2,k} + (\bar{\gamma}_3)_{i,j,k+1/2} \\
&\quad + (\bar{\gamma}_1)_{i-1/2,j,k} + (\bar{\gamma}_2)_{i,j-1/2,k} + (\bar{\gamma}_3)_{i,j,k-1/2} \\
\frac{\partial \hat{R}_{ijk}}{\partial D_{i+1,j,k}} &\approx \frac{1}{2}(\hat{A}_{i+1,j,k} - A^{+}_{i+1/2,j,k} + A^{-}_{i+1/2,j,k}) - (\bar{\gamma}_1)_{i+1/2,j,k}
\end{aligned}
\tag{3.71}
$$

where $\hat{A} = \hat{A}_1, \hat{B} = \hat{A}_2$, and $\hat{C} = \hat{A}_3$ as given by equation (3.26), and where only the orthogonal mesh terms are retained for the implicit viscous terms.

The matrix equation is solved using a line-relaxation method. First, the entire numerical matrix equation is formed from values at the *nth* time level. A significant amount of memory is required to store all of these values, but it is justified by the savings in computing time. At this point the numerical equation is stored as the following banded matrix

$$\mathcal{B}[T,0,...,0,U,0,...,0,X,Y,Z,0,...,0,V,0,...,0,W]\Delta D = \hat{R} \tag{3.72}$$

where $\Delta D = D^{n+1} - D^{n}$ and T, U, V, W, X, Y, and Z are vectors of 4 x 4 blocks which lie on the diagonals of the banded matrix, with the Y vector on the main diagonal. This matrix equation is approximately solved using an iterative approach. One family of lines is used as the sweep direction. Using, for example, the ξ family, a tridiagonal matrix is formed by multiplying the elements outside the tridiagonal band by the current ΔD and shifting them over to the right-hand side. This system can be represented by the following

$$
\begin{aligned}
\mathcal{B}[X,Y,Z]\Delta D^{l+1} = \hat{R} - T\Delta D^{l}_{i,j,k-1} - U\Delta D^{l}_{i,j-1,k} \\
- V\Delta D^{l}_{i,j+1,k} - W\Delta D^{l}_{i,j,k+1}
\end{aligned}
$$

where l is an iteration index. The number of iterations is generally very small. This equation can be solved most efficiently by first performing and storing the LU decomposition of the tridiagonal matrix before the iteration is begun. Then for each iteration, the right-hand side is formed using the latest known ΔD (which is set to zero for the first iteration), and the entire system is backsolved. The LU decomposition can be entirely vectorized, but the back solution is inherently recursive and cannot be vectorized.

3.6.3 Boundary Conditions for Upwind Scheme

Implicit boundary conditions are used at all of the boundaries, which enables the use of large time steps. At a viscous no-slip surface, the velocity is specified to be zero, and the pressure at the boundary is obtained by specifying that the pressure gradient normal to the wall be zero. The boundary conditions used for the inflow and outflow regions are based on the method of characteristics. The formulation of these boundary conditions is similar to that given by Merkle and Tsai (1986), but the implementation is slightly different. The scheme is derived here for a constant ξ boundary, with similar results for a constant η or a constant ζ boundary. The finite-speed waves which arise with the use of artificial compressibility are governed by the following

$$\frac{\partial \hat{D}}{\partial \tau} = -\frac{\partial \hat{E}}{\partial \xi} = -\frac{\partial \hat{E}}{\partial D}\frac{\partial D}{\partial \xi} = -\hat{A}\frac{\partial D}{\partial \xi} = -X\Lambda X^{-1}\frac{\partial D}{\partial \xi}$$

Multiplying by X^{-1} gives

$$X^{-1}\frac{\partial \hat{D}}{\partial \tau} = -\Lambda X^{-1}\frac{\partial D}{\partial \xi} \tag{3.73}$$

If the X^{-1} matrix is moved inside the spatial- and time-derivative, the result is a system of independent scalar equations, each having the form of a wave equation. These equations are termed the characteristic equations. The sign of the eigenvalues in the Λ matrix determines the direction of travel of each of the waves. For a positive or negative eigenvalue, there corresponds a wave that propagates information in the positive or negative ξ direction. The number of positive or negative eigenvalues determines the number of characteristic equations propagating information from the interior of the computational domain to the boundary. Thus, at the boundary, the characteristics equations which bring information from the interior will be chosen as part of the boundary conditions. The rest of the information should come from outside the computational domain, which leaves some variables to be specified. For inflow boundaries, two different sets of specified variables have been used successfully. One set consists of the total pressure and the cross-flow velocity. This set is useful for problems in which the inflow velocity profile is not known. For outflow boundaries, static pressures have been specified for computations presented later in this note.

IV. A FRACTIONAL STEP METHOD IN GENERALIZED COORDINATES

In the previous chapter, a pseudocompressibility method was described where a fictitious time derivative of pressure is added to the continuity equation so that the modified set of equations can be solved implicitly by marching in time. When a steady solution is reached the original equations are recovered. Time accuracy could be achieved by employing an iterative technique in a pseudo time level. An alternative approach for time-dependent computations is the pressure iteration method as outlined in chapter II. Among several choices in this category, the fractional step method allows flexibility in combining various operator splitting techniques. In light of successful computations in Cartesian coordinates using its numerous variants, we were interested in developing a general 3-D solver based on a fractional step approach. In this chapter, some details of the method developed by Rosenfeld et al. (1988, 1989) will be discussed.

4.1 Overview

Several key features must be carefully handled in developing this flow solver. Accuracy in an arbitrary domain is related to the method of discretizing the governing equations. Geometric quantities, for example, have to be discretized consistently with the flow differencing. For computational efficiency, an efficient Poisson solver for pressure is critical in this approach. To accomplish this, variable definitions and mesh arrangement are chosen to develop an efficient Poisson solver in curvilinear coordinates. A ZEBRA scheme with four color ordering has been devised for the present Poisson solver, in which a multigrid acceleration procedure can be readily incorporated in the future. In this respect, the governing equations are discretized by a finite-volume scheme on a staggered grid. As for the choice of dependent variables, the volume fluxes across each face of the computational cells are used. Thus the discretized mass conservation equation can be satisfied consistently with the flux balancing scheme explained later. In solving the momentum equations, a significant part of the viscous terms are solved implicitly by an approximate factorization scheme to minimize time step limitation resulting from use of a viscous grid. A staggered grid has favorable properties in cartesian coordinates, such as coupling odd-even points. However, it is debatable whether a staggered grid has clear advantages over a regular grid in generalized curvilinear coordinates. In the present procedure, a staggered arrangement is chosen to simplify the pressure boundary condition in devising a Poisson solver.

4.2 Formulation

The equations governing the flow of isothermal, constant density, incompressible, viscous fluids in a time-dependent control volume with the face area vector $\mathbf{S}(t)$ and volume $V(t)$ are written in integral form for the conservation of mass

$$\frac{\partial V}{\partial t} + \oint_S d\mathbf{S} \cdot (\mathbf{u} - \mathbf{v}) = 0 \tag{4.1}$$

and for the conservation of momentum

$$\frac{\partial}{\partial t} \int_V \mathbf{u}\, dV = \oint_S d\mathbf{S} \cdot \bar{T} \tag{4.2}$$

where t is the time, $\mathbf{u}$ is the velocity vector, $\mathbf{v}$ is the surface element velocity resulting from the motion of the grid, and $d\mathbf{S}$ is a surface area vector. The tensor $\bar{T}$ is given by

$$\bar{T} = -(\mathbf{u} - \mathbf{v})\mathbf{u} - pI + \nu\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right) \tag{4.3}$$

where $\nabla\mathbf{u}$ is the gradient of $\mathbf{u}$ while $(\cdot)^T$ is the transpose operator.

The only differences between the fixed and the moving-grid equations are the terms that include the surface element velocity $\mathbf{v}$ and the time dependence of the cell geometry (volume and face area). The volume conservation of each time-varying cell requires

$$\frac{\partial V}{\partial t} - \oint_S d\mathbf{S} \cdot \mathbf{v} = 0 \tag{4.4}$$

where the term $d\mathbf{S}\cdot\mathbf{v}$ represents the volume swept out by the face $\mathbf{S}$ over the time increment dt. Thus, the mass conservation equation has exactly the same form as for fixed grids

$$\oint_S d\mathbf{S} \cdot \mathbf{u} = 0 \tag{4.5}$$

The usual practice is to transform equations (4.2) and (4.5) into a differential form. In the present work, the integral formulation is maintained for convenience in deriving the finite-volume scheme for arbitrarily moving geometries.

4.3 Discretization

In the present integral formulation, spatial derivative terms are discretized in a different way from the differential equation approach as used in the previous chapter. In this section, details of the spatial discretization method are described.

4.3.1 Geometric Quantities

A general nonorthogonal coordinate system (ξ, η, ζ) is defined by

$$\mathbf{r} = \mathbf{r}(\xi, \eta, \zeta, t) \tag{4.6}$$

The center of each primary cell is designated by the indices (i, j, k). The area of the face l of a primary cell (figure 4.1) is given by the vector quantity

$$\mathbf{S}^l = \frac{\partial \mathbf{r}}{\partial(l+1)} \times \frac{\partial \mathbf{r}}{\partial(l+2)} \tag{4.7}$$

where the computational coordinates $l = \xi, \eta$ or ζ are in cyclic order and × is the cross product operator. The vector $\mathbf{S}^l$ has the magnitude of the face area and a direction normal to it and is the contravariant base vector ∇l scaled by the inverse of the Jacobian $1/J$, i.e., $\mathbf{S}^l = \frac{1}{J}\nabla l$. As pointed out by Vinokur (1986), an accurate discretization should satisfy certain geometric identities. The condition that a cell is closed should be satisfied exactly in the discrete form

$$\sum_l \mathbf{S}^l = 0 \tag{4.8}$$

where the summation is over all the faces of the computational cell. Equation (4.8) can be satisfied if $\mathbf{S}^l$ is approximated from equation (4.7) by a proper approximation of $\frac{\partial \mathbf{r}}{\partial l}$. For example, $\mathbf{S}^\xi$ is computed at the point $(i+\frac{1}{2}, j, k)$ by using the second order approximation:

$$\left(\frac{\partial \mathbf{r}}{\partial \eta}\right)_{i+\frac{1}{2}} = \frac{1}{2}\left(\mathbf{r}_{j+\frac{1}{2},k-\frac{1}{2}} - \mathbf{r}_{j-\frac{1}{2},k-\frac{1}{2}} + \mathbf{r}_{j+\frac{1}{2},k+\frac{1}{2}} - \mathbf{r}_{j-\frac{1}{2},k+\frac{1}{2}}\right)_{i+\frac{1}{2}}$$
$$\left(\frac{\partial \mathbf{r}}{\partial \zeta}\right)_{i+\frac{1}{2}} = \frac{1}{2}\left(\mathbf{r}_{j-\frac{1}{2},k+\frac{1}{2}} - \mathbf{r}_{j-\frac{1}{2},k-\frac{1}{2}} + \mathbf{r}_{j+\frac{1}{2},k+\frac{1}{2}} - \mathbf{r}_{j+\frac{1}{2},k-\frac{1}{2}}\right)_{i+\frac{1}{2}}$$

The volumes of all discrete computational cells will sum up to the total volume at a given time if the volume of each computational cell is computed by dividing the cell into three pyramids having in common the main diagonal and one vertex of the cell, resulting in the following

$$V = \frac{1}{3}\left(\mathbf{S}^\xi_{i-\frac{1}{2}} + \mathbf{S}^\eta_{j-\frac{1}{2}} + \mathbf{S}^\zeta_{k-\frac{1}{2}}\right) \cdot \left(\mathbf{r}_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} - \mathbf{r}_{i-\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}}\right) \tag{4.9}$$

In the present method, the volume conservation equation (4.4) is satisfied discretely by interpreting the term $d\mathbf{S} \cdot \mathbf{v}$ in equation (4.4) as the rate of the volume swept by the face $d\mathbf{S}$. For example the volume swept by the face $\mathbf{S}^{\xi}{}_{i+\frac{1}{2}}$ can be computed by a formula similar to equation (4.9).

$$(\delta V^{\xi}{}_{i-\frac{1}{2}})^{n+\frac{1}{2}} = \frac{1}{3}((\mathbf{S}^{\xi}_{i-\frac{1}{2}})^{n} + \delta\mathbf{S}^{\eta}_{j-\frac{1}{2}} + \delta\mathbf{S}^{\zeta}_{k-\frac{1}{2}}) \cdot (\mathbf{r}^{n+1}_{i-\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} - \mathbf{r}^{n}_{i-\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}}) \quad (4.10)$$

where the time level is given by n. The quantities $\delta\mathbf{S}^{\eta}_{j-\frac{1}{2}}$ and $\delta\mathbf{S}^{\zeta}_{k-\frac{1}{2}}$ are the areas swept by the motion of the face $\mathbf{S}^{\xi}{}_{i-\frac{1}{2}}$ - see the shaded areas in figure 4.1b. The area $\delta\mathbf{S}^{\eta}_{j-\frac{1}{2}}$ is computed from

$$\delta\mathbf{S}^{\eta}_{j-\frac{1}{2}} = \frac{1}{2}\left((\mathbf{r}^{n+1}_{k-\frac{1}{2}} - \mathbf{r}^{n}_{k+\frac{1}{2}}) \times (\mathbf{r}^{n+1}_{k+\frac{1}{2}} - \mathbf{r}^{n}_{k-\frac{1}{2}})\right)_{i-\frac{1}{2},j-\frac{1}{2}} \quad (4.11)$$

and $\delta\mathbf{S}^{\zeta}_{k-\frac{1}{2}}$ can be computed in a similar way. The volume of the cell at the time level $(n+1)$ is computed from equation (4.4).

$$V^{n+1} = V^{n} + \sum_{l}(\delta V^{l})^{n+\frac{1}{2}} \quad (4.12)$$

Note that $\delta V^{l}/\Delta t$ is the volume flux resulting from the motion of the coordinate system and has a meaning similar to the volume-flux U^{l}. An accurate computation of the volume of each cell at the time level $(n+1)$ is important for the accurate representation of the momentum equation.

In the present finite volume formulation, no coordinate derivatives appear directly in the discrete equations, as in the case of finite-difference formulas. Instead, quantities with clear geometric meaning such as the volume and the face areas of the computational cells are used. The discrete approximation of these quantities satisfies the geometric conservation laws. A principal difference between the finite-volume and the finite-difference approach to moving grids is in the interpretation of the quantity $\delta V^{l}/\Delta t$. In the finite-volume method, it is treated as a geometric quantity which expresses the rate of displacement of a cell face, whereas in the finite-difference method, the grid velocity is combined with the fluid velocity to define a "relative flow velocity"(see Vinokur, 1986).

4.3.2 Mass Conservation Equation

The discretization of the mass conservation equation (4.5) over the faces of the primary computational cells yields

$$(\mathbf{S}^{\xi} \cdot \mathbf{u})_{i+\frac{1}{2}} - (\mathbf{S}^{\xi} \cdot \mathbf{u})_{i-\frac{1}{2}} + (\mathbf{S}^{\eta} \cdot \mathbf{u})_{j+\frac{1}{2}} - (\mathbf{S}^{\eta} \cdot \mathbf{u})_{j-\frac{1}{2}} + (\mathbf{S}^{\zeta} \cdot \mathbf{u})_{k+\frac{1}{2}} - (\mathbf{S}^{\zeta} \cdot \mathbf{u})_{k-\frac{1}{2}} = 0 \quad (4.13)$$

Note that the default indices (i, j, k) and the time-level $(n+1)$ are usually omitted for simplicity. Each term on the left-hand side of equation (4.13) approximates the volume-flux over a face of the primary cell. A simple discretized mass conservation equation can be obtained by using the following variables

$$\begin{aligned} U^{\xi} &= \mathbf{S}^{\xi} \cdot \mathbf{u} \\ U^{\eta} &= \mathbf{S}^{\eta} \cdot \mathbf{u} \\ U^{\zeta} &= \mathbf{S}^{\zeta} \cdot \mathbf{u} \end{aligned} \tag{4.14}$$

as the unknowns instead of the Cartesian velocity components. The quantities U^{ξ}, U^{η}, and U^{ζ} are the volume fluxes over the ξ, η, and ζ faces of a primary cell, respectively. In tensor algebra nomenclature, these are the contravariant components of the velocity vector scaled by the inverse of the Jacobian $(1/J)$. With this choice of the dependent variables, the continuity equation takes a form identical to the Cartesian case

$$U^{\xi}_{i+\frac{1}{2}} - U^{\xi}_{i-\frac{1}{2}} + U^{\eta}_{j+\frac{1}{2}} - U^{\eta}_{j-\frac{1}{2}} + U^{\zeta}_{n+\frac{1}{2}} - U^{\zeta}_{n-\frac{1}{2}} = D_{iv} U^{l} = 0 \tag{4.15}$$

As explained in chapter II, it is crucial to satisfy the discrete mass conservation equation. Therefore, the simple form of equation (4.15) suggests that the volume fluxes can be chosen as the 'natural' dependent variables for fractional step methods. On the other hand, this choice complicates the discretization of the momentum equations. In the present work, volume fluxes and the pressure are chosen as the dependent variables.

4.3.3 Momentum Conservation Equation

Spatial discretization of the momentum-conservation-law equation (4.2) for a computational cell with volume V yields

$$\frac{\partial}{\partial t}(V\,\mathbf{u}) = \sum_{l} \mathbf{S}^{l} \cdot \bar{T} = \mathbf{F} \tag{4.16}$$

Following the development of the scheme given by Rosenfeld et al. (1988), a general two-step temporal discretization of equation (4.16) for a constant time-step Δt is

$$(1+\epsilon)(V\mathbf{u})^{n+1} - (1+2\epsilon)(V\mathbf{u})^{n} + \epsilon(V\mathbf{u})^{n-1} = \Delta t[\theta \mathbf{F}^{n+1} + (1-\theta)\mathbf{F}^{n} + \phi(\mathbf{F}^{n} - \mathbf{F}^{n-1})] \tag{4.17}$$

where n is the time level and ϵ, θ, and ϕ are arbitrary parameters. A truncation error analysis shows that second-order accuracy in time is obtained if

$$\frac{1}{2} + \epsilon = \theta + \phi \tag{4.18}$$

In order to replace $\mathbf{u}$ by the new dependent variables U^l, the corresponding area vectors are dotted with the momentum equations and the integral momentum equation is applied on different computational cells for each unknown U^l. Each cell has the dimensions $\Delta\xi \times \Delta\eta \times \Delta\zeta$, but the centers are located at $(i+\frac{1}{2}, j, k)$, $(i, j+\frac{1}{2}, k)$ and $(i, j, k+\frac{1}{2})$ for the U^ξ, U^η, and U^ζ momentum equations, respectively.

The derivation of the ξ-momentum equation will be described next. The other two momentum equations can be obtained by cyclic permutation. Using the identity

$$\mathbf{u} = \mathbf{S}_\xi U^\xi + \mathbf{S}_\eta U^\eta + \mathbf{S}_\zeta U^\zeta = \mathbf{S}_m U^m \tag{4.19}$$

the dot product of equation (4.17) and $\mathbf{S}^\xi$ results in

$$\begin{aligned}(1+\epsilon)(V\,U^\xi)^{n+1} - (1+2\epsilon)(V\,U^l)^n(\mathbf{S}^\xi \cdot \mathbf{S}_l^n) + \epsilon(V\,U^l)^{k-1}(\mathbf{S}^\xi \cdot \mathbf{S}_l^{k-1}) \\ = \Delta t[\theta \mathbf{S}^\xi \cdot \mathbf{F}^{n+1} + (1-\theta)\mathbf{S}^\xi \cdot \mathbf{F}^n + \phi \mathbf{S}^\xi \cdot (\mathbf{F}^n - \mathbf{F}^{n-1})]\end{aligned} \tag{4.20}$$

where $\mathbf{S}_m$ is the inverse base of $\mathbf{S}^l$. Note that the $n+1$ is the default time-level and therefore $\mathbf{S}^\xi$ is computed at $n+1$. To minimize the temporal truncation error, the coefficients in the above equation are chosen as

$$\epsilon = \frac{1}{2}, \quad \phi = 1 - \theta, \qquad \theta = 1$$

Then equation (4.20) becomes

$$3(V\,U^\xi)^{n+1} - 4(V\,U^l)^n(\mathbf{S}^\xi \cdot \mathbf{S}_l^n) + (V\,U^l)^{n-1}(\mathbf{S}^\xi \cdot \mathbf{S}_l^{n-1}) = 2\Delta t L^{n+1} \tag{4.21}$$

where

$$L = \mathbf{S}^\xi \cdot \mathbf{F}$$

The difference between a fixed-grid and a moving-grid case is in the computation of the convection terms which should include the motion of the grid. For example, the convection flux of the ξ-momentum equation on the ξ-face center (i, j, n) is given by

$$\left(-(U^\xi - \frac{\delta V^\xi}{\Delta t})U^l \mathbf{S}_l\right)_{i,j,k}$$

The difference equations are second-order accurate in time if δV^ξ would not lag in time by $\frac{\Delta t}{2}$ over the volume-flux terms U^l (see eq. (4.10)). The resulting discrete equations are conservative in any moving coordinate system and are spatially second-order accurate. For high Reynolds number flows, fourth-order dissipation is added to eliminate high frequency components of the solution. The dissipation terms are interpreted in terms of fluxes and therefore the conservation properties of the equations are maintained.

The above time–marching scheme requires the computation of the operator L_ξ.

$$L_\xi = \mathbf{S}^\xi_{i+\frac{1}{2}} \cdot \sum_l \mathbf{S}^l \cdot \bar{T} \tag{4.22}$$

where the summation is over all the faces of a computational cell. It is to be noted that

$$U^l = \mathbf{S}^l \cdot \mathbf{u} = \mathbf{S}^l \cdot \mathbf{S}_m U^m \tag{4.23a}$$

and the invariance of the velocity vector requires:

$$\mathbf{S}^l \cdot \mathbf{S}_m = \delta_{lm} \tag{4.23b}$$

where δ_{lm} is the Kronecker delta, and $\mathbf{S}^\mathbf{l}$ is the inverse base to $\mathbf{S}_\mathbf{m}$ and has the differential analogue $\mathbf{S}_m = J\dfrac{\partial \mathbf{r}}{\partial m}$. In terms of tensor algebra, $\mathbf{S}_\mathbf{m}$ is the covariant base vector scaled by the Jacobian J while $\mathbf{S}^\mathbf{l}$ is the contravariant base vector scaled by $1/J$. A uniform velocity field can be numerically preserved if the base $\mathbf{S}_\mathbf{m}$ is computed at each point from the relation

$$\mathbf{S}_m = \frac{\mathbf{S}^{m+1} \times \mathbf{S}^{m+2}}{\mathbf{S}^m \cdot (\mathbf{S}^{m+1} \times \mathbf{S}^{m+2})} \tag{4.24}$$

which satisfies (4.23b) identically. The variable m is the cyclic permutation of (ξ, η, ζ).

The product $\mathbf{S}^l \cdot \bar{T}$ should be computed for each face of each momentum equation, see equation (4.22). For example, the ξ face-center for the U^ξ momentum cell is located at (i, j, n) (see fig. 2). The flux over this face is computed from

$$(\mathbf{S}^\xi \cdot \bar{T})_{i,j,n} = \left(-(U^\xi - \frac{\delta V^\xi}{\Delta t})U^l \mathbf{S}_l - \mathbf{S}^\xi P + \mathbf{S}^\xi \cdot \nu(\nabla \mathbf{u} + (\nabla \mathbf{u})^T)\right)_{i,j,n} \tag{4.25}$$

The conservative form of the velocity vector gradient is

$$\nabla \mathbf{u} = \frac{\oint_S d\mathbf{S}\,\mathbf{u}}{V} \tag{4.26}$$

Applying equation (4.26) for the computation of $\nabla \vec{u}_{i,j,n}$ yields

$$\begin{aligned}\nabla \mathbf{u}_{i,j,n} = \frac{1}{V}(&\mathbf{S}^\xi_{i+\frac{1}{2}}\mathbf{u}_{i+\frac{1}{2}} - \mathbf{S}^\xi_{i-\frac{1}{2}}\mathbf{u}_{i-\frac{1}{2}} \\ &+ \mathbf{S}^\eta_{j+\frac{1}{2}}\mathbf{u}_{j+\frac{1}{2}} - \mathbf{S}^\eta_{j-\frac{1}{2}}\mathbf{u}_{j-\frac{1}{2}} + \mathbf{S}^\zeta_{k+\frac{1}{2}}\mathbf{u}_{k+\frac{1}{2}} - \mathbf{S}^\zeta_{k-\frac{1}{2}}\mathbf{u}_{k-\frac{1}{2}})\end{aligned} \tag{4.27}$$

where only those indices different from (i, j, n) are given.

The η or ζ face-centers are located at $(i+\frac{1}{2}, j-\frac{1}{2}, n)$ and $(i+\frac{1}{2}, j, n-\frac{1}{2})$, respectively. The fluxes over these faces are computed in a similar way. The convection and diffusion fluxes in equation (4.25) may be approximated in various ways. In the present work, all of the unknowns at the point (i, j, n) are computed by simple averaging and therefore the scheme is equivalent to the results computed by second order central differencing.

4.4 Solution Procedure

The dependent variables are the pressure, defined at the center of the primary cells, and the volume fluxes defined on the faces of the primary cells. This selection is equivalent to a finite-difference formulation over a staggered grid with the choice of scaled contravariant velocity components as the unknowns. The mass and momentum conservation equations (4.5) and (4.20) are solved by a fractional step method.

First the momentum equations are solved for an approximate U^l. This is done by computing ΔU^l defined by

$$\Delta U^l = (U^l)^{n+1} - (U^l)^n$$

This first step of the velocity correction is obtained from equation (4.20) by evaluating the pressure gradient term at time step (n). As explained in chapter II, various time advancing schemes can be implemented here. One possibility is an Adams-Bashforth type scheme used by Rosenfeld et al. (1988) which can be written as follows

$$\left(V_{i+\frac{1}{2}} I - \Delta t D_l\right) \Delta \bar{U}^l = \Delta t \left(\frac{3}{2} L_l^n - \frac{1}{2} L_l^{n-1} - R_\xi(\Delta P^{n-1}) - D_\xi(\Delta U^{\xi,n-1})\right) \qquad (4.28)$$

where D_l represents the viscous terms and R_l the pressure gradient terms. Here, only orthogonal viscous terms are included on the implicit side. Later linearized convection terms were included implicitly (Rosenfeld and Kwak, 1989), thus a larger time step could be taken. Discretization of each term in the above equation is given in the previous section.

In the second step, the velocity is updated such that the continuity equation is satisfied at the next time level. This is achieved by a single Poisson-type equation as follows:

$$D_{iv}\left(\frac{R_l(\phi)}{V_{m+\frac{1}{2}}}\right) = -D_{iv}\left((U^l)^n + \Delta \bar{U}^l\right) \qquad (4.29)$$

As explained in chapter II, it is to be noted that the Laplacian operator is written as a discrete divergence of a discrete gradient operator. This is an important step to satisfy the mass conservation equation in discretized space. Then the variables at the new time level $n+1$ are computed by

$$\begin{aligned}(U^l)^{n+1} &= (U^l)^n + \Delta U^l \\ P^{n+1} &= P^n + \phi\end{aligned} \tag{4.30}$$

It can be shown that U^l_{k+1} is the exact discrete solution of the momentum and mass conservation equations, but $\Delta P = \phi + O(\Delta t)$ and therefore P^{n+1} is not the exact discrete solution. In a Cartesian coordinate, Kim and Moin (1985) used the exact relationship

$$P^{n+1} = \phi - \frac{\nu \Delta t}{2} \nabla^2 \phi$$

where ∇^2 is the Laplacian operator. However, such a simple relationship cannot be found in generalized coordinate systems with the present splitting of the diffusion terms. Braza et al. (1986) have not found significant differences between solutions which used the approximation $P = \phi$ and solutions which computed the discrete pressure exactly, although the approximate solution may be less stable. As the difference between ϕ and ΔP is proportional to ν, the direct substitution $\Delta P = \phi$ is reasonable for high Reynolds number flows. However, this approximation may degrade the formal second order accuracy in time of the present method. This issue requires additional study.

The first step is a consistent approximation of the momentum equations, e.g. as $(\Delta t, \Delta l) \to 0, \quad U^l_{k+1} \to U^l$, where U^l is the solution of the continuous problem. Therefore, the physical boundary conditions may be specified at the $(n+1)$ time level. In some fractional step methods, the steps are not consistent and special boundary conditions should be devised for the intermediate steps.

One drawback of the present method is the increased memory required for storing variables at three time levels. Also, the effect of the many geometric approximations on the overall accuracy is yet to be fully evaluated. Even though the current procedure has been validated by many test cases including the moving domain problem, the relative advantages or disadvantages to pseudocompressibility method will have to be determined.

V. DEVELOPMENT AND VALIDATION OF FLOW SOLVERS

In the previous two chapters, solution methods based on two major primitive variable approaches, namely, pseudocompressibility and a fractional step approach, have been described for 3-D applications in generalized coordinates. Several computer codes have been developed according to these procedures. Once these codes are written, the accuracy, convergence speed, computational efficiency, and robustness have to be carefully evaluated.

The code development effort was motivated by a demand for efficient flow solvers in several major application areas, namely, simulation of the flow through an impeller in the SSME power head and other complex internal flows, simulation of a full aerodynamic configuration at low speed involving a wing-body type juncture flow, and time-dependent flow analysis of an artificial heart. Validation of the above flow solvers was done by choosing several basic configurations relevant to the local flow encountered in the above applications. To remove uncertainties caused by turbulence modeling, validation cases were chosen in the laminar range of Reynolds numbers first followed by turbulent cases. In this chapter, a summary of these codes and several representative validation computations will be presented.

5.1 Code Development

The following is a list of these incompressible Navier-Stokes codes. Each code is given a name and the results presented here will be identified by these names.

5.1.1 INS3D Code

A computer code hasbeen developed based on the steady-state algorithm described in section 3.4. A steady formulation of the pseudocompressibility approach is solved using an approximate factorization scheme. This takes advantage of the recent advances in state-of-the-art CFD made in conjunction with compressible flow computations. The spatial discretization utilizes second-order central differencing with additional numerical dissipation terms. The code has been validated and numerous 3-D problems have been solved using this code.

5.1.2 INS3D-UP Code

To obtain time-accurate solutions using the pseudocompressibility formulation, it is necessary to satisfy the continuity equation at each time step by subiteration in pseudotime. In order to use a large time step in the pseudotime iteration, an upwind differencing scheme based on flux-difference splitting is used combined with an implicit line relaxation scheme. This removes the factorization error and the need for numerical dissipation terms. The code has been validated. Major unsteady flow applications include simulation of the flow through an artificial heart.

5.1.3 INS3D-LU Code

This code is also based on the pseudocompressibility formulation. However, a finite volume scheme in conjunction with either central or upwind differencing is used for spacial discretization. An LU-SGS implicit algorithm is employed for temporal discretization. This code has an option to utilize a rotating coordinate system so that rotor-steady-state solutions (steady in relative frame of reference) can be computed. Therefore, the flowfield of a propeller in an infinite medium or in a ducted enviroment can be solved using this code. The code is completely vectorized; validation is in progress in conjunction with the Space Shuttle main engine flow analysis.

5.1.4 INS3D-FS Code

A generalized flow solver based on a fractional-step method has been developed for time-dependent computations of the incompressible Navier-Stokes equations. The governing equations are discretized conservatively using finite-volume approach on a staggered grid. Here, the discretized equations are advanced in time by decoupling the solution of the momentum equation from that of the continuity equation. This procedure, combined with accurate and consistent approximations of the geometric quantities, satisfies the discretized mass conservation equation exactly in a discrete sense. A four-color ZEBRA scheme is devised for solving the Poisson equation for the pressure correction. Several sets of computational results have been compared with experiments and other numerical solutions.

5.2 Channel Flow

The physical interpretation of the pseudocompressibility was given in chapter III. It is therefore interesting to test the validity of the estimate on the permissible range of the pseudocompressibility parameter, β. To do this, a validation calculation is performed over a range of β using a simple test problem. This study is done using the INS3D code. The channel flow is perhaps the simplest internal-flow test problem, where the pressure wave propagates between the in- and outflow boundaries while the viscous effect spreads inward from two walls. The coordinate system of a 2-D rectangular channel with a width of 1 and length of 15 is shown in figure 5.1, where velocity vectors for a converged solution are also shown. To obtain fully developed velocity profiles within a reasonable channel length, a partially developed boundary layer profile can be imposed at the inflow boundary. In this experiment, a uniform inlet flow is prescribed, and the Reynolds number based on the duct width and the average velocity is 1000, and the time step $\Delta\tau = 0.1$. Then the recommended range of β is estimated to be

$$0.12 < \beta < 10$$

To illustrate the pressure wave propagation phenomena and its effect on the convergence property, the channel is impulsively started. Here, five different values of β (0.1, 1, 5, 10 and 50) were chosen such that two cases are outside the range while three valuesare kept within the range.

Pressure contours at three different time levels are shown in figure 5.2, for $\beta = 5$ which is within the recommended range. The expansion wave from the exit plane propagates upstream sufficiently fast to balance the spreading of the viscous effect. The solution converges nicely in this case. However, for $\beta = 0.1$ which is lower than the recommended lower bound of β, the speed of the upstream propagating pressure wave from the exit plane is very low. Therefore, the expansion wave is confined near the exit plane while the viscous effect spreads into the flow field from both upper and lower surfaces. Thus the viscous field is not properly balanced by the right pressure gradient which causes the spurious fluctuations to amplify, as shown at $\tau = 2.5$, and eventually the computation blows up at $\tau = 4.0$.

The convergence history is shown in figure 5.3(a). The log of the root-mean-square of the change in pressure and velocities (RMSDQ) is plotted against the computation time τ. It can be seen that calculations for β=0.1 and 50 become unstable within 50 steps and start to diverge, whereas other cases converge to a stable solution. The effect of β values on the incompressibility of the fluid is shown in figure 5.3(b) in the form of the log of the root-mean-square of the divergence of the velocity field (RMSDIV) plotted against the pseudo time τ.

5.3 Flow Over A Backward-Facing Step

The flow over a 2-D backward-facing step is computed using INS3D and INS3D-UP. Figure 5.4 shows the schematic of the problem, where the step height is equal to the inlet height and the Reynolds number is based on twice the step height. The upstream boundary is located at the step and a fully developed channel flow is imposed at the inlet. This problem is very challenging computationally as it involves a primary and a secondary separation bubble. The size and the location of these separation zones are very sensitive to the pressure gradient, thereby providing a good viscous flow validation case.

Results obtained using INS3D have been reported previously by Rogers et al. (1985). The separation lengths were found to be sensitive to the magnitude of the numerical dissipation coefficient at higher Reynolds numbers. This in essence is equivalent to changing the effective Reynolds number of the flow as the Reynolds number approaches 1000. Therefore, it is quite desirable to remove the dissipation model dependence on the solution at a high laminar range of Reynolds numbers. In light of the similarity of the pseudocompressibility formulation to that of the compressible flows, the study done by Yoon and Kwak (1988) may be applied to achieve this. When the flow becomes turbulent, the Reynolds number based on turbulent eddy viscosity is in the order of a few hundred, and thus the present dissipation model may perform adequately.

More extensive validation has been done by using the INS3D-UP code by Rogers (1988). The computed separation and reattachment locations are then compared to experimental values given by Armaly et al. (1983) in figure 5.5. For the primary reattachment length, x1, good agreement is observed between the experiments and the computation until the secondary separation appears at a Reynolds number of about 400. At a higher Reynolds number, the primary separation length, x1, and the secondary separation point, x2, deviates from the experimental data. Similar results were reported by Kim and Moin (1985) using a fractional step method. Armarly et al. reported that 3-D flow was observed near the step when the Reynolds number is greater than 400. For more complete validation, the numerical simulation needs to include the entire experimental configuration including the possible 3-D effect. In figure 5.6, the convergence history is shown. Using the grid of 100 points in the streamwise direction and 53 points in the crossflow direction, a good converged solution is obtained within 100 iterations and under 11.5 seconds of computing time on the Cray 2.

5.4 Flow Over A Circular Cylinder

To represent an external flow, the flow past a circular cylinder was computed. For external flows in which the computational domain extends a large distance from the body, the pressure waves originating from the body surface propagate into the farfield. Therefore, to obtain the near-field solution only, the distance traveled by the waves and the spreading of the vorticity can be considered approximately the same in magnitude. The range of β in INS3D can then be estimated based on this reasoning. For example, at Re=40, if the viscous region is taken to be approximately two diameters away from the body, the following range for β using $\Delta\tau = 0.1$ can be estimated as

$$0.1 << \beta < 10.$$

Physically, for external flows, the pressure wave can quickly travel a short distance to balance the viscous region close to the body. Therefore, the magnitude of β is less restrictive than for internal flow cases such as the channel flow described above, which is consistent with our past experience. Good agreement with literature data was found for steady state computation at Re=40 (Kwak et al. 1986, INS3D). Far more extensive validation was done at Re=5, 10, 20, and 40 by Rogers and Kwak (1988) using INS3D-UP and showed good agreement with experiments and other computations (not illustrated here).

To capture the near-field detail of transient flow, impulsively started circular cylinders at Re=40 and 200 is computed using INS3D-FS code (see Rosenfeld et al., 1988, 1989). In figure 5.7, time evolution of the separation length is compared with experiments by Coutanceau and Bouard (1977) and other computations by Collins and Dennis (1973). The comparison is very good.

As the Reynolds number increases above 40, nonsymmetric wake develops and periodic vortex shedding sets in (for a comprehensive review, see Morkovin, 1964). Both INS3D-UP and INS3D-FS codes are validated using this simple yet challenging problem. These computations are compared with other numerical and experimental results. The calculation was performed using an O-type grid clustered near the body. To obtain time-accurate solutions from INS3D-UP, the subiterations were carried out at each physical time step. Starting impulsively from rest, over 20 subiterations were required during the transient phase. Non-symmetric wake develops spontaneously followed by shedding of vortices without introducing any artificial disturbance, probably due to biasing in the upwinding scheme which in turn may have introduced enough disturbance into the flow field. When a central differencing scheme is used as in INS3D-FS, it is necessary to introduce asymmetric disturbance to initiate the shedding. Figure 5.8 shows a Strouhal number plot versus Reynolds number from computed results compared to experiments. Good agreement is observed. The Strouhal number is perhaps relatively easy to predict. However, the time-accuracy of the code itself has been validated using other problems with exact solutions (see Rogers and Kwak, 1988a).

The staggered pattern of the vortex shedding known as Karman's vortex street has been a subject of many flow visualization studies. For the purpose of comparison, particle traces are generated from the time-dependent solution of flow around a circular cylinder at a Reynolds number of 105 using the INS3D-UP (see Rogers and Kwak, 1988). Figure 5.9(a) shows this computed vortex street. Shown in figure 5.9(b) is an experimental photograph of the same conditions taken by Taneda and reproduced from Van Dyke (1982). The streaklines in the experiment are shown by electrolytic precipitation in water. The vortex structure is seen to be very similar between the two. The experimental picture is digitized and displayed on a workstation along with the computationally generated flow visualization picture. This example illustrates the potential of using postprocessing of the CFD results for studying fundamental fluid dynamic phenomena.

5.5 Curved Duct of Square Cross-Section

The flow through a square duct with a 90° bend offers a good test case for a full Navier-Stokes solver. This flow is rich in secondary flow phenomena both in the corner regions and through the curvature in a streamwise direction. This geometry was studied experimentally by Humphrey et al. (1977) and Taylor et al. (1981, 1982), and extensive laminar flow data are available. This particular geometry was used as a steady state test case for both INS3D and INS3D-UP code. The geometry is shown in figure 5.10; the Reynolds number of the flow is 790. The problem was nondimensionalized using the side of the square cross-section, H.

For validating INS3D-UP (Rogers and Kwak, 1989), the inflow velocity was specified to be that of the fully developed, laminar, straight square duct (see White, 1974). The velocity is normalized by the average inflow velocity. Three different grids were used and the pseudocompressibility parameter β was set to be unity throughout the computation. The computed results are compared to the experimental results of Humphrey et al. (1977) as shown in figure 5.11. Overall, the comparison is quite satisfactory. The velocity profile through the middle of the bend is very good. However, at the latter part of the bend, the computation shows a double inflection in the velocity profile, which is not evident in the experimental results. It is seen that the peak velocity is very well predicted.

Extensive validation of the INS3D code was performed by McConnaughey et al. (1989). Utilizing the detailed inflow measurement by Taylor et al. (1981, 1982), they investigated various aspect of the flow solver including a grid refinement study and a sensitivity study of the numerical dissipation terms. Computed results on a 90° bend and an S-bend are extensively compared in the same report. Figures 5.12 and 5.13 show detailed velocity contours which illustrate the quality of the numerical solutions compared with experiments. The flow field details compare very nicely and the overall comparison is excellent.

5.6 Juncture Flow

The flow around a cylinder-plate or a wing-body juncture produces an interesting viscous phenomena due to the interaction between the boundary layer from the plate and the cylinder. The 3-D separation of the boundary layer and subsequent formation of the so-called horseshoe vortex and its development is very challenging to analyze both experimentally and numerically. This type of flow can occur in many practical engineering problems. Flow around a wing-fuselage junction and an appendage-submarine body are simple examples of its kind, and the flow near the endwall of turbomachinery blades might be one of the most complicated juncture flow problems. One major motivation of studying this type of juncture flow is related to the flow analysis of the Space Shuttle main engine, which will be presented in detail in the next chapter. In the SSME, liquid oxygen posts are densely packed in the main injector region. Even though a single cylinder-plate flow is an extreme idealization of the flow in the actual SSME environment, validating the computer code in this simplified case is of considerable value in extending the code to realistic case where detailed experimental measurement is very difficult and scarce.

5.6.1 Cylinder-Flat Plate

Most of the earlier studies on cylinder-flat plate juncture flow have been experimental. Baker (1979) shows that laminar juncture flow is confined to a very limited region. A similar result has been obtained most recently by Thomas (1986). Eckerle and Langston (1986) reported a single primary vortex and saddle point contrary to multiple vortex systems observed earlier by other researchers. Interpretation of the phenomena also varies (Thomas, 1986 and Peak and Tobak, 1980).

Computational simulation of these flows involves distinctively different features from those of external aerodynamics. For instance, the thickness of the viscous layer for these types of flows is of the same order as the characteristic flow-field dimension, while the viscous region tends to be confined in a thin layer near the body for external flows. Realistic juncture flows under an internal flow environment are likely to have a large amount of deflection as in the case of LOX post regions in the SSME. Recently, some numerical studies on this flow have been attempted. Kaul et al. (1985) reported a numerical study on a single cylinder-plate flow using INS3D code. Highlights of this and other studies were reported by Kwak et al. (1986). Independently, Kiehm et al (1986) reported a numerical study of flow around a single post in a channel. These computational results show qualitatively similar phenomena. Here, several representative results using INS3D are presented.

In figure 5.14, the computational domain for a single post on a flat plate is illustrated. The upstream boundary layer thickness is varied by using partially and fully developed channel flow profiles. The convergence characteristics of the flow solver are shown in figure 5.15 by the history of RMSDQ, which denotes the root-mean-square value of the change per iteration in the pressure and velocities. The three curves in the figure show three variations of the INS3D code; namely a block tridiagonal, a diagonal version with second-order implicit smoothing terms, and a diagonal version with fourth order implicit smoothing terms as explained in chapter III. The flow solver converges fast to about four orders of magnitude reduction in RMSDQ. The computing time per iteration per grid point is 91 μsec for the block tridiagonal version and 32 μsec for the diagonal version of the code.

In figure 5.16, particle traces for a single post at Re= 1000 are shown. A saddle point separation and a horseshoe vortex can be seen from the traces near the flat plate. The secondary flow in front of the cylinder wraps around toward the wake region and forms a counter-rotating pair of vortex filaments. These spiraling twin vortices demonstrate a striking difference between this type of juncture flow and a 2-D cylinder. The vortex filaments are washed upward. They attenuate as they interact and move downstream. In reality, vortex shedding and possible unsteady motion take place at this stage. These tornado shaped vortices are very difficult to observe experimentally, and validation of this phenomenon was very much needed. Recently, Schewe (Schewe, G., Private communication, 1985, DFVLR, West Germany) produced an oil flow visualization around a single post which shows a clear evidence of the twin vortex behind the cylinder as shown in figure 5.17. This experimental observation is qualitatively similar to the computed results, shown by the particle traces in Figure 5.16. This juncture flow structure will lead to a strong variation in skin friction and pressure along the cylinder and hence significantly affects the overall loading on the post.

5.6.2 Wing-Flat Plate

Another similar juncture flow, namely the wing on a flat plate case, was studied by Burke (1989). Numerical results were obtained by applying INS3D and compared with the experimental data of Dickinson (1986). The wing shape is a hybrid consisting of a 1.5:1 elliptic nose and a NACA 0020 tail joined at the location of maximum thickness. This is a generic configuration characterizing a wing-fuselage juncture of aircraft or a hull-appendage juncture of ships. The Reynolds number of the experiments, based on the chord length of the wing, is 5.0×10^5. The coordinate system is shown in figure 5.18.

Turbulence modeling for this type of complex flow is very challenging. A high level turbulence model may be necessary for detailed study on the juncture flow itself. Since this study was done to extend the flow solver to more realistic applications, computing efficiency is of major importance. Therefore, the computational simulation was done by devising a simple algebraic turbulence model derived from Patankar et al. (1979). The results compares remarkably well with experimental data capturing all important features of the flow. Some of the results are reproduced here. In figure 5.19, surface pressure on a flat plate near the wing is compared with measurement while, in figures 5.20 and 5.21, velocity contours at two different vertical planes are shown. Overall, the computed results compare very favorably with the experimental data.

So far several representative validation cases have been presented in this chapter. Other validation cases can be found in the references cited throughout this note. In the next chapter, two important recent applications of these codes will be presented.

VI. APPLICATIONS

As discussed in Chapter I, there are a wide variety of 3-D problems where the present flow solvers can be utilized to analyze current configurations and to verify new design concepts. In this chapter, computed results of two major 3-D applications will be discussed.

6.1 SSME Power Head Flow Simulation

As the role of the Space Shuttle becomes increasingly important in scientific and commercial applications, it is desirable to increase the payload capability. Recently, an upgrade of the SSME power head was initiated to substantially increase the operating margin and the engine durability. To achieve this goal without increasing the weight and size of the existing engine, it became essential to understand the dynamics of the hot-gas flow in the engine power head. Because of the complexity of the geometry, an experimental approach is extremely difficult as well as time consuming. Computational simulation, therefore, offers an economical alternative to complement the experimental work in analyzing the current configuration, and to suggest new, improved design possibilities. In the past few years, major milestones have been established from the computational effort. Here, highlights of our initial task are presented.

6.1.1 Background of SSME flow analysis

In the SSME staged combustion cycle, the fuel is partially burned at very high pressure and relatively high temperature in the preburners. The resulting hot gas is used to run the turbine and is then routed to the main injector where, along with additional oxidizer, it is injected into the main combustion chamber. The Reynolds number of the primary flow in the manifold is of the order of 10^6 per inch. Because of the high gas temperature, the Mach number is less than 0.12. The flow is turbulent and is practically incompressible.

Figure 6.1 illustrates the current arrangement for the SSME power head components. Hot gas discharged from the gas turbine enters the annular turnaround duct (TAD) and experiences a 180° turn before it diffuses into the fuel bowl. This assembly is called the hot gas manifold (HGM). The gas flows into the main injector through three transfer ducts on the left side of the power head (fuel preburner side) and enters into the region of the main injector posts. On the right side of the power head (oxidizer preburner side), there are two transfer ducts connected to the right side of the main injector assembly. Around the bottom portion of each liquid oxygen (LOX) post in the main injector assembly, there are a number of small holes through which the hot gas flows into the main combustion chamber. There it mixes with the oxidizer, which comes through a circular passage along the centerline of each LOX post. As a part of the engine development effort, a CFD study has been conducted to simulate the dynamics of the hot-gas flow in the power head. At the time this effort was started only the first version of INS3D was available. Therefore, the computed results presented in this section have been obtained using INS3D code.

6.1.2 The computer model and the grid

A computational model of the power head is chosen to analyze critical areas where the dynamics of the hot-gas flow are expected to have a significant effect on the overall performance of the SSME. As shown in figure 6.2, the model starts from the gas turbine exit on the fuel preburner side, and extends to the main injector assembly. The main injector consists primarily of a bundle of LOX posts, which is physically modeled by a porous media.

Figures 6.2(a) and 6.2(b) demonstrate the 3-D grid for the SSME HGM. They show a horizontal and a vertical cross section, respectively, of the HGM. Figure 6.2(c) illustrates the details of H-grids for the cross-section of the three transfer ducts. This H-grid is generated for a unit circle. Near the boundary the grid lines are concentric circles except in the vicinity of the four singular points. Using the nearly orthogonal grid in this unit circle, one can obtain H-grids for tubes or ducts of any given shape and dimension by a simple linear transformation. Figure 6.2d shows an unwrapped surface of the annular fuel bowl with openings. The elliptical shape shown in figure 6.2c represents a cross section of the transfer duct of an advanced two-duct design, which will be explained later in this section. The current H-type grid topology for the circular cross-section was chosen to allow smooth transition from the axisymmetric TAD to the transfer ducts. Also, this arrangement is convenient in clustering the grid near the duct wall.

The grid for the entire HGM system is generated by using algebraic functions, and is written with a high degree of flexibility for changing geometric configurations. By specifying the shape, the dimension, and the desired number of transfer ducts, a grid for a variety of new HGM configurations can be obtained in a short time. The ducts described in this note are connected directly to the fuel bowl without any fairings, while in the current engine the three transfer ducts are connected smoothly to the annular fuel bowl with fairings. This configuration with an abrupt change in geometry is more demanding computationally than smooth configurations.

6.1.3 Multiple-Zone Computation

A large number of mesh points are required to solve the 3-D turbulent flow in the SSME. To facilitate numerical simulations, the domain of interest is divided into several zones. This requires a special treatment at the interface for a smooth continuation of the solution between zones. Figure 6.3 illustrates a five-zone arrangement for the HGM flow field. Zone 1 includes the TAD and fuel bowl. Zones 2, 3, and 4 comprise the three transfer ducts for the current engine configuration. The annular region denoted as the racetrack of the main injector is represented by Zone 5. Also shown in the figure are some overlapping grids in the various zonal interfaces. The grid is chosen to be continuous and smooth across the zonal boundaries. In the first attempt at simulating this model, the racetrack (Zone 5) was not included in the computation. In a later computation involving a new design configuration, the racetrack region was included as well as the main injector assembly which was modeled by a porous medium. More detailed study of the LOX post in the main injector assembly was also performed using the results of the HGM computation as the boundary condition but will not be given here. Since the vertical plane through the center of the fuel bowl and the main injector is taken to be a plane of symmetry, results for the HGM presented in this note were obtained using only half of the HGM. When the swirling flow from the gas turbine exit is included, the symmetry assumption should not be used.

In the pseudocompressible formulation, waves are propagating in both up- and downstream directions while the solution approaches a steady state. In the present problem, the interfaces between zones are locations where the geometry changes abruptly. Therefore, in the neighborhood of those interfaces, the flow is expected to experience a rapid change. To maintain a smooth continuation of the solutions across these zones, and hence to achieve a stable and fast-converging computation, a means for providing adequate communication for the traveling waves must be established. Overlapping regions and a proper zonal interpolation scheme are thus required for this purpose.

A forward or backward differencing, if applied to the interfaces of multiple zones, would distort the geometric representation. To maintain a smooth transition of the flow field across a zonal boundary, the Jacobian and the metrics at the interfaces are computed using grid points in neighboring zones. Then the pressure and the velocities, Q, are updated explicitly at each iteration. Let values at $(n+1/2)$ denote conditions to be used to advance the computation to $n+1$. The values of $Q^{n+1/2}$ for Zone 1 at the exit plane are obtained from the values of the corresponding plane of Zone 2 at n, i.e.,

$$\left[Q_{B.C.}^{n+1/2}\right]_{Zone1} = [Q_{interior}^{n}]_{Zone2}$$

Values of $Q^{n+1/2}$ for Zone 2 at the zonal interfaces are taken from the latest computed result of Zone 1 as

$$\left[Q_{B.C.}^{n+1/2}\right]_{Zone2} = \left[Q_{interior}^{n+1}\right]_{Zone1}$$

When more than two points are overlapped, the latest values in the interior of this overlapping region must be properly transmitted to the next zone. There are a number of ways to treat this problem. The simplest one is take an average of the two values computed in Zones 1 and 2, as below:

$$\left[Q^{n+1/2}\right] = \frac{1}{2}\left([Q^{n}]_{Zone2} + [Q^{n+1}]_{Zone1}\right)$$

A scheme using updated zonal boundary values, but using original interior values, has also been tested. Either way, converged steady-state solutions have been obtained. However, the scheme with interior updates converges at a much faster rate.

6.1.4 Grid effect

The present flow solver (INS3D code) has been validated by computing fundamental fluid dynamics problems as illustrated in the last chapter. There are, however, many other aspects to be clarified in the real world applications, such as the grid induced error due to skewness and 3-D stretching, and grid clustering and its relation to numerical dissipation. In real geometry, the flow is likely to go through strongly curved sections. This will introduce yet another subject, that of strong curvature effect on turbulence structure.

In the present application, for continuity and smoothness across zonal boundaries, an H-type grid is chosen for the circular transfer duct as shown in figure 6.2c. This is then connected to the side-wall grid of the HGM as illustrated in figure 6.2d. In generating this grid for the SSME, errors are introduced mainly due to grid singularities, skewness, and stretching.

The present AF algorithm integrates the difference equations along the transformed coordinates. At the junction of the two H-grid directions, flow particles in the two coordinate directions could communicate only indirectly via the interior mesh points. This produces some corner-effect error. Even though this error is not as severe as the one caused by external flows, an ad hoc method of eliminating this corner effect is devised based on a finite-element concept. Let j and k denote the indices for the grid points along the increasing ξ and η directions, respectively. Then let $j = k = 1$ be the corner point, which is singular. First, the pressure at this point, p_{11}, is determined by an extrapolation along the diagonal direction $j = k$. Second, p_{13} and p_{31} are obtained in the usual manner. Then p_{12} and p_{21} are established by an interpolation along the circular surface.

The full viscous term given by equation (3.18a) can be simplified to equation (3.18b) when the grid is orthogonal. Even though full viscous terms can be used, it is convenient and economical to keep only the orthogonal part. It is, therefore, of practical interest to estimate the overall error caused by the orthogonal formulation when a nonorthogonal grid system is used.

As a quick measure of an overall error caused by grid skewness, a 2-D channel flow is computed using two grids, namely, 1) a stretched Cartesian grid (orthogonal) and 2) a nonorthogonal grid where the skewness is controlled on the upper half of the channel as sketched in figure 6.4. In the computation only orthogonal terms are kept. Converged solutions on the lower half, where the two grids are identical, are then compared. Total error depends additionally on the Reynolds number and the third-directional skewness. However, this quick experiment indicates that the orthogonal assumption can be used without significantly impacting on the overall solutions if the grid is basically orthogonal. Full investigation should include more severe cases involving separation and recirculating regions.

6.1.5 Turbulence models

The quality of the solution under turbulent conditions will be validated next. The flow through the SSME power head offers a variety of rich internal flow phenomena. Among others, the curvature effect of turbulence was most uncertain at the time the current simulation started. Therfore, the validation effort which focused on this will be summarized here.

The current configuration of the SSME hot gas manifold was designed including an axisymmetric annular 180° TAD. In the sharp U-turn region, the ratio of the internal shear layer thickness to the radius of curvature is of order one. On the convex side, i.e., the inner wall, the turbulence is expected to be greatly reduced, while on the concave side, it is expected to be substantially enhanced. Furthermore, strong adverse and favorable pressure gradients coexist and interact with each other. Because of the sharp turn, the streamwise variations and the normal gradient of the streamwise velocity are of the same order of magnitude. Furthermore, the flow rapidly changes direction. Understanding the structure of turbulence in this flow is crucial for a successful computation of the TAD.

Since the original work by Prandtl in 1929, the effect of mild steamline curvature on the structure of a turbulent boundary layer has been studied by many investigators (for a comprehensive review, see, for example, Bradshaw, 1973). It is well known that a mild curvature produces a profound effect on the turbulent flow structure. For example, if the ratio of the boundary layer thickness to the radius of the wall curvature is merely 1 percent, there will be approximately 10 percent or more change in the turbulent quantities. In most external flows such as the flow over an airfoil, the streamline curvature is designed to be very mild except near the nose region where flow remains laminar. For problems of this kind, the literature on the curvature effect is quite extensive (see, for example, Wattendorf, 1935; Eskinazi and Yeh, 1954; Gillis and Johnston, 1983; Moser and Moin, 1984). However, study on a strong curvature effect is very limited.

To meet this demand, several experimental investigations were started to study the fundamental structure of turbulence associated with sharp U-turn internal flows. In one experiment, which was performed at Rocketdyne by Sharma et al. (1987), turbulent flows in an axisymmetric U-duct were studied. Figure 6.5 shows the TAD model for this experiment. The duct width is 2 inches, and the radius of the centerplane of the annular duct is 10 inches upstream of the turn and 14 inches downstream of the 180° bend. The radius of curvature is 1 inch for the convex inner wall, and 3 inches for the concave outer wall. From the beginning through the completion of the turn, the annular cross-sectional area undergoes a 40 percent increase. The boundary layer on the inner wall side is therefore expected to separate in the neighborhood of the 180° turn. The experiment is conducted at atmospheric pressure. The Reynolds number is approximately 10^5, and the Mach number is about 0.1. Hot wire and hot film probes are used for data acquisition.

For such complicated turbulent flow as encountered in the SSME, high level turbulence models such as a two-equation k-ϵ model or Reynolds-stress model may be needed. However, for many engineering applications, a simple yet adequately accurate model will be of considerable value for design purposes. Therefore, developing and validating a simple algebraic turbulence model which can be used for simulating flows with strong streamline curvatures is currently of interest. This case is used as a part of the general flow solver and turbulence model validation for application to the SSME.

Several levels of turbulence models have been implemented into INS3D. These include the Baldwin and Lomax (1978) algebraic model in which length scale is determined by the location of the maximum moment of vorticity. This model has been widely applied in the external flow problems. However, the maximum moment of vorticity is not as well defined for fully turbulent internal flows as for external flows. In particular, the moment of vorticity is almost constant for a fully developed pipe or channel flow except in the sublayer region.

For fully turbulent internal flow in a duct, the two boundary layers associated with the opposite walls merge together. For these flows, there exists a pair of opposite-sense vortices. Therefore, there exists a position where these two opposite vortices cancel each other. This position is clearly the one which divides the two boundary layers. Chang et al. (1985b) proposed a simple extended Prandtl-Karman mixing length theory which, together with the strength of vorticity, forms an eddy viscosity model. The vorticity length scale is defined to be the distance between the location of maximum vorticity and the point where vorticity first vanishes. This length is an active measure of the turbulent eddy size. The location of the maximum vorticity for an attached boundary layer is at the wall. When the layer begins to separate in the presence of a strong adverse pressure gradient, the point where the magnitude of vorticity is maximum starts to move away from the wall. The point of vanishing vorticity can be replaced by the location of minimum vorticity in merged layers for the purpose of a numerical study. The accuracy of this model is evaluated here using the present test case.

In the present experimental setup, it is very difficult to produce a fully turbulent flow in the entire duct. Even with a boundary layer trip mechanism, the artificially thickened layers upstream of the turn cover only about half the duct width. In the center core region, the flow remains practically inviscid. This inviscid core flow acts like a free-stream in an external flow. As free-stream conditions are approached near the edge of the boundary layer, an intermittent zone of turbulent and nonturbulent flow is encountered. This and the wake component in the outer layer change the structure of the shear layer drastically from that of a fully turbulent flow. For these flows, the mixing length is modified accordingly. For more details on the model, see Chang and Kwak (1988). A review on various levels of turbulence models can be found in articles by Reynolds (1976) or Ferziger (1987).

In the present problem, only three azimuthal planes are required because of the axisymmetry. The center plane is at θ =0 and other two planes are set on either side of the center plane offset by a small angle. Across the duct width, 81 grid points are distributed with an expansion factor of about 1.2 . The smallest grid size next to the wall is 0.02% of the duct width which corresponds to y^+ of less than 5. In order to resolve the detail of the separated flow profile, the largest grid size in the core region is set to be 0.025 times the width.

The 180° U-turn region is divided by 42 equal intervals. An orthogonal cylindrical grid system is used here to eliminate any uncertainty in accuracy due to non-orthogonality. The grids upstream and downstream of the bend are then fixed by smooth expansions from the bend region. Close to both walls these grids are orthogonal. The flow path is therefore, represented by a 3x81x95 grid in the circumferential, radial and flow directions respectively. This grid in the U-turn region is shown in figure 6.6.

Figure 6.7 shows the comparison of the static pressure distributions between the measured data and the computed results. The measured data were obtained in an earlier experiment in which no turbulent tripping mechanism was used. The resulting boundary layer on the outer wall at 8 inches upstream of the bend, i.e., x/h = -4, is very thin. By using these measured velocities as an inlet condition at that location, the computed result shown with a solid line agrees very well with the experiment. In the neighborhood of the 180° turn, pressure on the inner wall shows a dip followed by a long recovery from the trough. In external flows, boundary layer separations usually interrupt the pressure recovering processes. The resulting pressure downstream of the separation will be almost constant. In the present confined duct, the flow which is pushed outward due to inner layer separations is squeezed back by the flow on the opposite side. As a result, the effective flow area is reduced creating a region of local acceleration which results in a local pressure dip.

In a more recent experiment, the layer is tripped. It is artificially thickened to about 0.5 inches at the position 10 inches upstream. Because of the thickened layer at the outer wall, a larger portion of kinetic energy is carried by the inner-wall region inducing a larger acceleration in the bend. As a result, the static pressure on the inner convex wall is expected to be a little lower than in the previous run.

Comparisons of the measured and the computed streamwise velocity profiles are shown in figure 6.8. Agreement is in general very good except near the outer concave wall in the bend region. The numerical results show a slight separation from the beginning of the turn through about 103°, while the experimental data shows attached layers all the way through the turn. Accuracy of the measured velocities in this region is still uncertain due to two factors. First, because of the sharp concave wall effect, the enhanced turbulent intensities obtained are as high as 45 percent of the measured mean velocities. Since the hot-wire probe measures only the rate of heat transfer, it does not detect the direction of the flow. In a region where heat transfer is significantly affected by the turbulent fluctuations, it is extremely difficult to obtain the true mean flow velocities as evidenced by experiments in the low speed region of a mixing layer. Second, the mass flow obtained by integration of the measured velocities over the cross-sectional surface of these few planes is substantially larger than the amount of mass coming from the inlet. The computed velocity profiles conserve the mass flow to an order of 10^{-5}. Because of these uncertainties, no conclusion can be drawn for the discrepancies at this time. On the convex inner-wall side, the layer begins to separate at about 125° into the turn. The data obtained by the hot-wire probe registered a high value due to large turbulent fluctuations in the separated region, and the data do not reveal the direction of the flow.

Overall, the accuracy of the solution using the present algebraic turbulence model is quite acceptable. Therefore, this model has been used quite extensively for simulating the SSME power head. In the remainder of this section, flow solutions in a variety of different HGM configurations at various Reynolds numbers are presented. Here, the Reynolds number is based on the mean velocity and the duct width at the entrance of the TAD.

6.1.6 Current Three-circular-duct HGM Analysis

First, steady-state solutions are obtained for the current three-circular-duct HGM at Re=1000. As illustrated in figure 6.9, for the initial computation, the racetrack of the main injector is not connected to the ducts. The three transfer ducts are assumed to discharge the flow separately, which results in no communication of the pressure between the center and the outer ducts at their exit planes. Therefore, the downstream condition is not a good approximation to the real case. For this reason, small residual waves have remained in the computed results. However, the root-mean-square value of the change in the flow variables, RMSDQ, has dropped below 10^{-5}, and an essentially steady-state solution has been obtained.

In figures 6.10(a) and 6.10(b), velocity vectors are shown in the horizontal and vertical cross-sections corresponding to figure 6.2. The flow in the center transfer duct is highly nonuniform, and a large separation region is formed just downstream of the entrance to the transfer ducts. By comparison of the vector length in figure 6.10(a), the flow in the center duct is much slower than it is in the outer ones. The results shown in these figures agree qualitatively with the airflow test data conducted at a Reynolds number of about 10^6. The predicted mass flow through the center duct is 9.8% of the total mass flow, which agrees with the test data.

Figures 6.11(a) illustrates the 3-D velocity vectors at an unwrapped plane near the inner wall of the fuel bowl. A reverse-flow pattern is clearly visible near the inner wall. Three-dimensional swirl patterns are predicted in the vicinity of the entrance to the transfer ducts. Figure 6.11(b) is a photograph that indicates, by means of surface-streak (shear-pattern) visualization, the similar swirls at the corresponding locations in the airflow test.

The existence of the swirls can be explained as follows. The flow coming from below has a large momentum due to the relatively small width of the annular duct. Among the streamlines of this flow in between the two ducts, there exists a dividing streamline. This streamline has a stagnation point at the top of the fuel bowl as shown in Figure 6.11a. On the left side of this dividing streamline, the flow is bent leftward to the center duct. Because of symmetry, a rightward flow is also approaching the center from the other side. When these opposite currents approach each other, another dividing streamline is formed with a stagnation point, again at the top wall. The stagnation pressure forces the streams to bend downward, and at the same time, the streams make a right-angle turn into the circular duct. Conservation of momentum thus requires the formation of swirls.

The pattern of the swirl and its center depends on the relative strength of the appoaching currents. Near the center duct, double swirls of equal strength are formed because of symmetry. In the vicinity of the entrance to the outer duct, the current approaching leftward from the rear part of the bowl is larger than the one approaching rightward. A stronger swirl is thus formed, located sideways toward the weaker stream.

Figures 6.12 and 6.13 are the perpendicular cross-sectional views showing three different sections of the transfer ducts; namely, near the entrance, at the midsection, and near the exit plane. Near the entrance (figures 6.12a and 13a), the velocity vectors in the center duct have symmetric double swirls, while the outer duct has a strong swirl accompanied by a much weaker one. The swirling velocities are largely reduced at the midsection and are physically dissipated before entering the main injector regions.

6.1.7 Development of New Design Configuration

From this computational flow analysis and also from experiments, the center duct of the current three-duct HGM is found to transfer a limited amount of mass flow (about 10% of the total flow). Also the transverse pressure gradient remains large with a large bubble of separation after the 180° turn. To improve the quality of the flow, a large-area, two-duct design concept has been developed. In addition, the ducts are chosen to have an elliptical shape in order to distribute the mass flow evenly to the main injector region. An outer-surface grid for a two-duct model is illustrated in figure 6.14.

First, to reduce the large separation bubble after the 180° turn, a parametric study is performed to find the best possible configuration. In figures 6.15(a) and 6.15(b), comparison of the current three-duct configuration and the new two-duct design is illustrated. As shown in figure 6.15(a), a large separation bubble existing in the present design is practically removed in the new configuration, shown in figure 6.15(b). This is confirmed by experiment as shown in figures 6.15(c) and 6.15(d) for the current and new designs, respectively, where velocity measurements at five different locations across the channel between the inner and the outer wall are shown. To find the most favorable flow conditions, over 20 different two duct configurations were studied computationally, thus providing potentially the optimum geometry to the designers. Ideas from researchers were incorporated into the current human optimization process. Fully numerical optimization may be considered in the future in conjunction with development of a faster code.

Figure 6.16(a) illustrates an example of the triple-swirl pattern in the elliptical duct near the entrance at Re=10^3. Here, because of the absence of the center duct, the stream approaching from the left side is much stronger than in the previous case. Another point of interest is that the upcoming stream entering the elliptical duct directly from below is also more massive. The three currents are of almost the same strength, resulting in a triple-swirl flow. The swirling is greatly dissipated along the duct. Figure 6.16b shows the remaining small swirling vectors at the duct exit. A steady-state, turbulent-flow solution for an HGM with two elliptical transfer ducts at Re = 10^5 has been obtained by Chang et al. (1985). In figure 6.17, the turbulent swirling flow in the elliptical transfer duct is illustrated. In comparing these results with the laminar solution in figure 6.16, it is seen that only a double-swirl pattern exists.

Once the new configuration is developed, it is necessary to analyze the new engine relative to the old one. To simulate the new two-duct engine accurately, a more complete numerical model was developed as shown in figure 6.18 where both the fuel side (left side) and the oxidizer side (right side) of the power head are included in the computer model.

One important objective of the redesign was to reduce the transverse pressure gradient at the exit of the gas turbine underneath the preburner. Therefore, the turbulent solution for the new two-duct HGM is compared to that of experiments performed using both the current three-duct HGM and the new two-duct configuration. As shown in Figure 6.19, the pressure gradient around the fuel bowl of the HGM is greatly reduced from that of the original configuration. From the hardware perspective this results in substantially reduced load on the bearings which hold the gas turbine and the turbopump sketched in figure 6.1.

The most significant objective of the present study is to pinpoint the locations where flow experiences the greatest energy losses. An important measure of the energy losses is the mass-weighted average total pressure along the flow. Figure 6.20 illustrates the decreasing coefficient of the mass-weighted total pressure along the centerline of the TAD, the fuel bowl, and the transfer duct. The total pressure coefficient C_{p_0} is defined as

$$C_{p_0} = \frac{\bar{p}_0 - \bar{p}_{01}}{\bar{p}_0 1}$$

where

$$\bar{p}_0 = \frac{1}{M} \int \left[P + \frac{1}{2}(u^2 + v^2 + w^2) \right] dm$$

The discontinuities shown in the figure correspond to the entrance of the duct where energy fluxes are computed over different planes. In the figure, three different HGM configurations are compared. The initial two-duct design shows 28% less total pressure drop compared to the current three-duct version. After fine-tuning the two-duct configuration computationally, the pressure drop decreased even further to 36% less than the original configuration. This final configuration is then tested using cold air flow, which shows 40% reduction in pressure loss.

6.2 Artificial Heart Flow Simulation

As shown thus far, the major advances in CFD have been made in aerospace engineering. With the advent of supercomputers as well as the development of fast algorithms, computational flow simulations have become a practical means for aerospace designs. Therefore, it will be of considerable benefit to medical researchers to extend this CFD technology to biofluid analysis. Some of the important studies in biofluid mechanics deal with aspects of blood flow, such as heart and blood vessel problems (see, for example, Peskin, 1982), which are relevant to cardiovascular diseases and their treatment. Understanding the flow phenomena through numerical simulations will be of significant value toward finding treatments for these problems. Therefore, the potential payoff to human health will be tremendous. In this chapter, the potential of CFD in biofluid mechanics research is demonstrated by simulating the flow in an artificial heart.

6.2.1 Background

Recently, the demand has grown for mechanical hearts or ventricular-assist devices (VAD) for use as a temporary life support system. These devices are becoming a powerful tool for assisting patients to recover from heart attacks or as a temporary bridge to transplants. Presently, There are several problems with these devices mainly involving the flow of blood. A research and development effort is in progress to improve the flow quality as well as to develop better material and control systems. Experimental investigations on these devices are very limited and many aspects of the flow are yet to be studied. Therefore, it will be very valuable to medical researchers to simulate the flow in these devices by applying state-of-the-art CFD technology.

Blood flow through these mechanical devices is very complicated in many respects. The fluid may exhibit significant non-Newtonian characteristics locally and the geometry is usually very complicated. In an artificial organ, as red blood cells go through high-shear turbulence regions, they may be damaged; the downstream region of an artificial heart valve is an example. The flow is unsteady, possibly periodic, and very viscous and incompressible. This problem is very much interdisciplinary and an attempt for a complete simulation would be a very formidable task. However, an analysis based on a simplified model may provide much-needed physical insight into mechanics of the blood flow through these devices.

The formulation of the flow solvers described earlier in this note is based on a Newtonian fluid assumption. However, since the governing equations are solved in a generalized coordinate system, viscosity that varies in space and time as well as moving geometry is allowed. These flow solvers can be applied to analyze mechanical hearts and ventricular-assist devices. A full simulation of viscoelastic flow is very difficult because of the nonlinearities of the fluid. However, as a first step toward full simulations, non-Newtonian effects of the blood flow can be simplified by a constitutive model for the viscous stresses. The primary purpose of the current application is to transfer technology developed for aerospace applications to artificial heart research. This requires the state-of-the-art in CFD for treating unsteady internal flow with moving boundaries. Researchers in the aerospace field will benefit from the advances made in unsteady flow simulation while the artificial heart developers benefit by gaining a better understanding of the fluid flow within their devices which hopefully will lead to an improved design.

There are many variations of VAD and artificial hearts. Among those, two different approaches are illustrated in figures 6.21 and 6.22. In the device depicted in Figure 6.21, the blood is pumped by pressurized air supplied externally. Simulation of this device requires a moving boundary procedure where the boundary itself should be determined by the motion of two fluids. Another approach is developed by Pennsylvania State University, where the pumping is done by an electric motor. This heart is sketched in figure 6.22. Our demonstration calculation is performed on this Penn State artificial heart. Experiments on various configurations are in progress at Penn State University. In the computational simulation, the time-accurate INS3D-UP code is used. The geometry of a computer model and some preliminary solutions are presented in this section.

6.2.2 Geometry and Grid Generation

The actual model of the Penn State artificial heart poses some very difficult problems from a computational standpoint. A computer model of the heart, i.e., surface grid and a shaded-surface representation, is shown in figure 6.23. The heart is composed of a cylindrical chamber with two openings on the side for valves. The pumping action is provided by a piston surface which moves up and down inside the chamber. The actual heart has a cylindrical tube extending out of each of the valve openings. These tubes contain tilting flat disks which act as the valves. The current computational model will neglect the valves altogether and will use the right and left openings shown in figure 6.23 for the inflow and outflow boundaries, respectively. In the computations, as the piston reaches its topmost position, the outflow valve closes and the inflow valve will open instantaneously. Similarly, as the piston reaches its bottommost position, the outflow valve will open and the inflow valve will close.

In the actual heart device the piston moves through the entire chamber volume, which includes most of the valve opening. This will cause some severe problems for the computational grid as the piston moves across the valve boundaries. Since the flow solver is designed to use body-fitted coordinates, it is necessary to place the grid lines around the valve to coincide with the valve opening boundaries. Yet since the piston moves past this opening, the grid will have to accommodate both of these surfaces. There are several ways in which a computational grid can handle this motion. One method is with the use of a Chimera scheme (see Beneck *et al.*, 1985) in which two grids are used, one which moves with the piston, and one which is attached to the rest of the body. Information is passed between the two grids by interpolating variables at the grid interfaces. This method can be quite expensive and very complicated to implement. Two other methods which are somewhat simpler to implement require one grid inside the computational domain. One of these methods use a stationary grid through which the piston surface travels. Boundary conditions applied at the piston surface would ensure that mass and momentum are conserved for the partial grid cells as the piston moves through them. However, with the curved surfaces at the valve boundaries, this method would require partial grid cells with an arbitrary number of sides, for which generalizations are difficult. Another single grid method uses a grid which varies in time as the piston moves through the boundary. The generalized coordinate transformation enables provision for grid motion in the equations by use of the time varying metrics. This method requires that a separate grid be generated for each discrete position of the piston during the calculations. This may be the simplest method that will accommodate the entire piston motion, yet it does leave a difficult grid generation problem.

In an effort to eliminate this problem, a simpler piston motion was decided upon for the calculations of the present study. The piston motion was restricted so that the piston did not rise above the bottom of the valve openings. The single grid approach that varies in time was used, so that a constant number of grid points are used, and the grid in between the piston and the bottom of the valve openings compresses and expands as the piston moves up and down. To facilitate this action, the piston was allowed to travel further down, so that the overall volume of the chamber in the computations was larger than in the actual device.

To make the most efficient use of grid points, an H-H grid topology was used to fit the grid to the computational domain. The grid dimensions were chosen to be 39 x 39 x 51. In order to generate a grid at each time step for this geometry, the surface grid was first generated, and from that, an algebraic grid generator and elliptic smoother was used to generate the interior points using the distribution given on the surface grid. To generate the surface grid, the side boundary was divided into seven different zones. The grid points were distributed along each of the zonal boundaries, and then a biharmonic solver was used to generate the grid interior to each of the zones. The biharmonic solver was also used to generate an H-grid for the the top and bottom surfaces of the heart device. The inner points were then computed with the use of an algebraic solver coupled with an elliptic smoother. This approach made it relatively simple to repeat the process at each time step for any given position of the piston surface.

6.2.3 Computed Results

The calculations were carried out on a Cray 2 supercomputer with a core memory of 256 million words. This large amount of memory has proven to be very useful because the implementation of the implicit scheme in INS3D-UP requires about 180 times the number of grid points in words of memory. For the current modest grid of 77,571 points, over 14 million words are required. The computations started with fluid at rest, the piston at the bottom position, and the outflow valve open. The computations were carried out for a Reynolds number of 500. The pseudotime-step $\Delta\tau$ was set to 10^{12} and the physical time step Δt was set to 0.025. The velocity of the piston during the motion in between its maximum and minimum positions was set to be constant, which is very nearly the same as that in the actual heart device. The period of the entire piston cycle was set to a time of 5 nondimensional units, so that 200 physical time steps were required for each full cycle of the piston.

During each time step, the subiterations were carried out until the maximum residual dropped below a specified small value or until a maximum of 20 subiterations were used. During most of the piston cycle only 12-15 subiterations were required, but when the piston was changing directions, it did not completely converge in 20 iterations. This did not cause any stability problems, yet it remains to be seen what effect this has on the accuracy of the solution. The computing time required for each period of piston motion was approximately four hours. The computations were run for four periods during which time particle paths were computed after being released near the inflow valve.

Some very interesting flow physics were observed during this period of motion. Various post-processing techniques have been utilized in analyzing the results, such as the velocity vectors colored by pressure level, and vorticity magnitude contours which can be related to the wall shear stress. Figure 6.24a shows particle traces as the piston nears its bottom position. Two distinct vortices are seen to have formed from the flow separating as it enters through the inflow valve. In figure 6.24(b), an experimental photograph (J.M. Tarbell, Pennsylvania State University, 1988) shows bubbles entering the inflow valve as the piston nears its bottom position. A similar two-vortex system is seen to form here. Figures 6.25a and 6.25b show velocity vectors during the inflow phase in planes passing through the center of the inflow valve. The first of these show a top view of vectors in a plane to the piston, while figure 6.25(b) shows a side view of vectors in a plane perpendicular to the piston. These figures portray the complexity of the vortical structure of this flow. Figure 6.25(a) again shows the presence of two vortices formed as the incoming flow forms a jet. Figure 6.25(b) shows also how the flow separates underneath the valve opening, although the flow there is strongly three-dimensional. Also in the figure is seen the presence of additional vortices against the back wall opposite the valve opening.

Some measurements of turbulent flow through the artificial heart geometry were taken by Tarbell *et al.* (1986). They were able to measure some of the mean flow characteristics, as well as turbulent shear stresses. However, these measurements were taken with the disk valves in operation. These valves greatly affect the incoming fluid by directing it along the walls so as to set up one large vortex inside the chamber which washes the walls free of any stagnating fluid. This effect is by design so that the blood would not tend to clot due to a large residence time in a stagnation region. Since these valves were not included in the current calculation, no qualitative comparison can be drawn between the experiment and the current calculation.

The present solution shows the capability of the computational procedure for simulating complicated internal flows with moving boundaries. For this initial calculation, the motion of the piston of the actual device was simplified. Also simplified was the valve opening and closing procedure. One difficulty in performing the present study is that there is currently very little validation available for these computations. It is realized, however, that this work represents a first step toward developing a CFD tool for this type of flow. With the use of more advanced grid techniques, and multiple zones to handle the valves, results of the entire artificial heart project will be reported in the future.

VII. CONCLUDING REMARKS

In the present note, numerical simulation methods for viscous incompressible flows are discussed from an application point of view. Our main interest has been in the 3-D real-world geometry. Naturally, the computational requirements for these problems are different from the fluid dynamics studies in the fundamental category. Therefore, the main emphasis has been placed on a primitive variable formulation which has been the most commonly used approach for 3-D problems to date. Detailed discussion has been given on flow solvers developed at NASA Ames Research Center. The validation cases and applications presented are the results from these solvers.

There are wide spectrum of algorithms and procedures for incompressible flows not covered here. The number of numerical studies of various incompressible flow problems are also staggering. It is an almost impossible task to review all this material. The material presented here is that familiar to the author; thus many important results and references are naturally left out, some for brevity and some due to ignorance. Many comments are made based on the author's experience without substantiation with firm numerical proof; they do not represent a consensus.

There are several important aspects left out in the discussion. In differencing the governing equations, conservative properties have been limited to momentum and mass conservation. Another property to be considered is the kinetic energy conservation. This is especially important in simulating turbulence involving small scale motions. In general, for the type of applications we have been interested in, the total number of mesh points and the total computing time limitations did not allow very fine mesh computations. For the class of problems requiring a fine mesh, such as the large eddy simulation of turbulence, kinetic energy conservation becomes crucial so as not to numerically create small-scale turbulence, which will eventually lead to nonphysical accumulation of energy in high-frequency motion.

In a more practical sense, even though computer speed and memory have been increased substantially in the recent past, the speed and the memory requirements of a flow solver are still dictated by the turn around time. In many engineering applications, it is very important to generate solutions in a timely fashion to have any impact on the design and analysis. We feel that numerical simulations can now provide complementary information to measured data, thus reducing the number of experimental trials required for developing advanced flow devices. A more sophisticated approach, such as computer optimization of a design, can be attempted when the flow solver becomes at least one order of magnitude faster, or the computer speed improves that much perhaps by massive parallelism.

There are several areas to be considered in developing a new generation of fast flow solvers. Several additions to the technology presented here will help to make the current methods as efficient as possible. Multigrid acceleration is one possibility which is physically consistent with the incompressible formulation. Use of a solution adaptive grid is also a practical approach to improve the accuracy and efficiency of the computation. This, as well as problems involving a moving domain, may require the generation of a new grid at each time step. For generating a grid at each time step, automated grid generation would be ideal. Overall, the solution procedure should be developed to best utilize computer characteristics such as vectorization, parallel processing, and access to memory.

Regarding the development of a universal code, it is hard to devise the best scheme for all flow speeds and for many different types of flow. Considering the uncertainties related to the turbulence model, it seems better to have a specialized code suitable for each different class of problems. For example, one can imagine having a code for an impeller type problem while having another for flow over aerodynamic shapes. Despite the limitations in algorithm speed and accuracy, and computer speed and memory, when the state-of-the-art flow solvers are utilized by creative researchers, the result will be tremendously beneficial for developing modern devices requiring viscous incompressible flow analysis. The current work in INS at NASA-Ames will continue to develop each of the codes presented in this note, which are considered to be the most promising methods for real-world flow simulations. As these codes are applied to these problems, we hope to quantify the advantages and shortcomings of these codes.

ACKNOWLEDGEMENT

The author wishes to thank all of his collaborators at NASA (S.E. Rogers, S. Yoon, M. Rosenfeld and P. McConnaughey) for providing materials included in the present lecture note. Special thanks are due to James L.C. Chang, R-J Yang, S.J. Lin, K. Zhu, S. Barson and G. Belie of Rocketdyne and to S. Chakravarthy who was then working at Stanford University. Special thanks are due also to G. Schewe of DFVLR in West Germany for providing the oil flow visualization picture around a single post and to J. Tarbell of Penn State University for providing the experimental particle flow visualization picture of an artificial heart.

REFERENCES

Armaly, B. F., Durst,F., Pereira, J.C.F., and Schönung,B., "Experimental and Theoretical Investigation of Backward Facing Step Flow,"J. Fluid Mech., vol. 127, pp.473-496,1983

Baker, C. J., "The Laminar Horseshoe Vortex,"J. Fluid Mech., vol. 95, part 2, pp. 346-367, 1979.

Baldwin, B.S. and Lomax,H., "Thin layer approximation and algebraic model for separated turbulent flows", AIAA Paper 78-257, 1978.

Barth, T. J., "Analysis of Implicit Local Linearization Techniques for Upwind and TVD Algorithms," AIAA Paper 87-0595, 1987.

Beam, R. M., and Warming, R. F., "An Implicit Factored Scheme for the Compressible Navier-Stokes Equations,"AIAA J., vol. 16, pp. 393-402, 1978.

Beneck, J. A., Buning, P. G., and Steger, J. L., "A 3-D Chimera Grid Embedding Technique," AIAA Paper 85-1523-CP, 1985.

Bradshaw, P., "Effect of Curvature on Turbulent Flow,"AGARD-AG-169, Aug. 1973.

Braza, M., Chassaing, P. and Ha Minh, H., "Numerical Study and Physical Analysis of the Pressure and Velocity Fields in the Near Wake of a Circular Cylinder", J. Fluid Mech., vol. 165, pp. 79-130, 1986.

Burke, R.W., "Computation of Turbulent Incompressible Wing-Body Junction Flow,"AIAA Paper 89-0279, 1989, Reno, NE.

Caretto, L.S., Gosman, A.D., Patankar, S.V. and Spalding, D.B., "Two Calculation Procedures for Steady Three-Dimensional Flows with Recirculation,", Proceedings of the 3rd Int. Conference on Numerical Methods in Fluid Dynamics, Paris, France, p. 60, 1972.

Chakravarthy, S. R., and Osher, S., "A New Class of High Accuracy TVD Schemes For Hyperbolic Conservation Laws," AIAA Paper 85-0363, 1985.

Chang, J. L. C., and Kwak, D., "On the Method of Pseudo Compressibility for Numerically Solving Incompressible Flows,"AIAA Paper 84-0252, 1984.

Chang, J. L. C., Kwak, D., and Dao, S. C., "A Three Dimensional Incompressible Flow Simulation Method and its Application to the Space Shuttle Main Engine, Part 1, Laminar Flow,"AIAA Paper 85-0175, 1985.

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 II. Turbulent flow", AIAA Paper 85-1670, 1985.

Chang, J. L. C. and Kwak, D., "Numerical Study of Turbulent Internal Shear Layer Flow in an Axisymmetric U-Duct, "AIAA Paper 88-0596, 1988.

Chang, J.L.C., Kwak, D., Rogers, S. E. and Yang, R-J, "Numerical Simulation Methods of Incompressible Flows and an Application to the Space Shuttle Maine Engine, "Int. J. Numerical Method in Fluids, vol. 8, pp. 1241-1268, 1988

Choi, D. and Merkle, C.L., "Application of time-iterative schemes to incompressible flow", AIAA J., vol. 23, no. 10, 1518-1524, 1985.

Chorin, A. J., "A Numerical Method for Solving Incompressible Viscous Flow Problems", J. Comp. Phys., vol. 2, pp.12-26, 1967.

Chorin, A.J., "Numerical solution of Navier-Stokes equations", Mathematics of Computation, vol. 22, no. 104, 745-762, 1968.

Collins, W. M. and Dennis, S. C. R., "Flow Past an Impulsively Started Circular Cylinder", J. Fluid Mech., vol. 60, pp. 105–127, 1973.

Coutanceau, M. and Bouard, R., "Experimental Determination of the Main Features of the Viscous Flow in the Wake of a Circular Cylinder in Uniform Translation. Part 2. Unsteady Flow", J. Fluid Mech., vol. 79, pp. 257–272, 1977.

Dennis, S.C.R., Ingham, D.B. and Cook, R.N., "Finite-Difference Methods for Calculating /steady Incompressible Flows,"J. Comp. Phys., vol. 33, pp. 325-339, 1979.

Dickinson, S.C., "An Experimental Investigation of Appendage-Flat Plate Junction Flow, Volumes 1 and 2,"David Taylor Research Center, Bethesda, MD, DTNSRDC Reports 86/051, 86/052, Dec. 1986.

Dwyer, H.S., Soliman, M. and Hafez, M., "Time accurate solutions of the Navier-Stokes equations for reacting flows", Proceedings of the 10th Int. Conference on Numerical Methods in Fluid Dynamics, Beijing, China, pp. 247-251, Springer-

Verlag, June 1986.

Eckerle, W. A. and Langston, L. S., "Horseshoe Vortex around a Cylinder, ASME Int. Gas Turbine Conference, Dusseldorf, West Germany, June 8-12, 1986.

Eskinazi, S. and Yeh, H., "An Investigation on Fully Developed Turbulent Flows in a Curved Channel,"J. Aero. Sci. Vol. 23, 1956. (Also, The Johns Hopkins University, Mechnaical Engineering Department, Internal Flow Research Report I-20, August, 1954).

Fasel, H., "Investigation of the Stability of Boundary Layers by a Finite-Difference Model of the Navier-Stokes Equations,"J. Fluid Mech. vol. 78, part 2, pp. 355-383, 1972.

Ferziger,J.H., "Incompressible Turbulent Flows", J. Comp. Phys., vol. 69, pp. 1-48, 1987.

Flores, J., "Convergence Acceleration for a Three- Dimensional Euler/Navier-Stokes Zonal Approach,"AIAA Paper 85-1495, 1985.

Gatski, T.B., Grosch, C.E. and Rose, M.E., "A Numerical Study of the Two-Dimensional Navier-Stokes Equations in Vorticity-Velocity Variables,"J. Comp. Phys., vol. 48, pp. 1-22, 1982.

Gillis, J.C. and Johnston, J.P., "Turbulent Boundary-Layer Flow and Structure on a Convex Wall and Its Redevelopment on a Flat Wall,"J. Fluid Mech., Vol. 135 pp 123-153, 1983.

Gresho, M. P. and Sani, R. L., "On Pressure Boundary Conditions for the Incompressible Navier–Stokes Equations", Int. J. Numerical Methods in Fluids, vol. 7, pp. 1111–1145, 1987.

Hafez, M., Dacles, J. and Soliman, M., "A Velocity Vorticity Method for Viscous Incompressible Flow Calculations,"11th Int. Conference on Numerical Methods in Fluid Dynamics, Williamsburg, Virginia, June 27-July 1, 1988.

Harlow, F. H. and Welch, J. E., "Numerical Calculation of Time-Dependent Viscous Incompressible Flow with Free Surface," Phys. Fluids, vol. 8, no. 12, pp. 2182-2189, Dec. 1965.

Harten, A., Lax, P. D., and Van Leer, B., "On Upstream Differencing and Godunov-Type Schemes for Hyperbolic Conservation Laws," Siam Review, vol. 25, no. 1,

p. 35, 1983.

Humphrey, J. A. C., Taylor, A. M. K., and Whitelaw, J. H., "Laminar flow in a square duct of strong curvature," J. Fluid Mech., vol. 83, part 3, pp. 509–527, 1977.

Jameson, A., Schmidt, W., and Turkel, E., "Numerical Solution of the Euler Equations by Finite Volume Methods Using Runge-Kutta Stepping Schemes," AIAA Paper 81-1259, 1981.

Jameson, A., and Yoon, S., "Multigrid Solution of the Euler Equations Using Implicit Schemes,"AIAA J., vol. 24, pp. 1737-1743, Nov. 1986.

Kaul, U. K., Kwak, D., and Wagner, C., "A Computational Study of Saddle Point Separation and Horseshoe Vortex System,' AIAA Paper 85-182, 1985.

Kiehm, P., Mitra, N. K. and Fiebig, M., "Numerical Investigation of Two- and Three-Dimensional Confined Wakes behind a Circular Cylinder in a Channel,"AIAA Paper 86-0035, 1986.

Kim, J. and Moin, P., "Application of a Fractional-Step Method to Incompressible Navier-Stokes Equations," J. Comp. Phys., vol. 59, pp. 308–323, 1985.

Kwak, D., Chang, J. L. C., Shanks, S. P., and Chakravarthy, S., "A Three-Dimensional Incompressible Navier-Stokes Flow Solver Using Primitive Variables,"AIAA J, vol. 24, no. 3, 390-396, Mar. 1986.

Kwak, D., Chang, J. L. C., and Shanks, S. P., "A Solution Procedure for Three-Dimensional Incompressible Navier-Stokes Equation and Its Application,"9th Int. Conference on Numerical Methods in Fluid Dynamics, CEN-Saclay, France, June 25-29, 1984 (Also NASA TM 85978, June 1984)

Kwak, D., Rogers,S.E., Kaul, U.K. and Chang, J.L.C., "A numerical study of incompressible juncture flows", 10th Int. Conference on Numerical Methods in Fluid Dynamics, Beijing, Peoples Republic of China, Jun. 23-27, 1986.

MacCormack, R. W., "Current Status of Numerical Solutions of the Navier-Stokes Equations," AIAA Paper 85-0032, 1985.

Marchuk, G.M., Methods of Numerical Mathematics, Springer-Verlag, 1975.

McConnaughey, P., Cornelison, J. and Barker, L. "The Prediction of Secondary

Flow in Curved Ducts of Square Cross-Section,"AIAA Paper 89-0276, 1989.

Merkle, C. L. and Athavale, M., "Time-Accurate Unsteady Incompressible Flow Algorithms Based on Artificial Compressibility," AIAA Paper 87-1137, 1987.

Merkle, C. L. and Tsai, P. Y. L., "Application Of Runge-Kutta Schemes to Incompressible Flows," AIAA Paper 86-0553, 1986.

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, pp. 102-118, 1964.

Moser, R.D. and Moin, P., "Direct Numerical Simulation of Curved Turbulent Channel Flow,"NASA TM 85974, 1984.

Orszag, S. A., and Israeli, M., "Numerical Simulation of Viscous Incompressible Flows,"Annual Rev. Fluid Mech., vol. 6, 1974.

Orszag, S. A., Israeli, M. and Deville, M.O., "Boundary Conditions for Incompressible Flows", J. Scientific Computing, vol. 1, pp. 75–111, 1986.

Osswald, G., Ghia, K.N. and Ghia, U., "Direct Algorithm for Solution of Incompressible Three-Dimensional Unsteady Navier-Stokes Equations,"AIAA Paper 87-1139, 1987.

Patankar, S. V. and Spalding, D. B., "A Calculation Procedure for Heat, Mass and Momentum Transfer in Three–Dimensional Parabolic Flows", Int. J. Heat and Mass Transfer, vol. 15, pp. 1787–1806, 1972.

Patankar, S. V., Ivanovic, M. and Sparrow, E.M., "Amnalysis of Turbulent Flow and Heat Transfer in Internally Finned Tubes and Annuli,"J. Heat Transfer, vol. 101, 9929-37, 1979.

Patankar, S.V., "Numerical heat transfer and fluid flow", Hemisphere Publishing Co., New York, 1980.

Peake, D. J. and Tobak, M., "Three-Dimensional Interactions and Vortical Flows with Emphasis on High Speeds,"NASA TM 81169, March 1980.

Peskin, C. S., "The Fluid Dynamics of Heart Valves: Experimental, Theoretical and Computational Methods," Ann. Rev. Fluid Mech., vol. 14, pp. 235–259, 1982.

Prandtl, L., "Effects of Stabilizing Forces on Turbulence,"NACA TM 625, 1931 (original version in 1929).

Pulliam, T. H., and Chaussee, D. S., "A Diagonal Form of an Implicit Approximate-Factorization Algorithm"J. Comp. Phys., vol. 39, pp. 347-363, 1981.

Pulliam, T.H., "Artificial dissipation models for the Euler equations", AIAA J., vol. 24, pp. 1931-1940, Dec. 1986.

Rai,M.M., "Navier-Stokes Simulations of Blade-Vortex Interaction Using High-Order Accurate Upwind Schemes," AIAA Paper 87-0543, 1987.

Raithby,G.D. and Schneider, G.E., "Numerical solution of problems in incompressible fluid flow: treatment of the velocity-pressure coupling", Numerical Heat Transfer, 2, 417-440, 1979.

Reynolds,W.C., "Computation of turbulent flows", Ann. Rev Fluid Mech., vol 8, 183-208, 1976.

Roe, P. L., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," J. Comput. Phys., vol. 43, p. 357, 1981.

Roach, P.J., Computational Fluid Dynamics, Hermosa Publishers, Albuquerque, New Mexico, 1972.

Rogers, S. E., Chang, J. L. C., and Kwak, D., "A Diagonal Algorithm for the Method of Pseudocompressibility,"J. Comp. Phys. vol. 73, No. 2, pp. 364-379, 1987.

Rogers, S. E., Kwak, D., and Kaul, U., "On the Accuracy of the Pseudocompressibility Method in Solving the Incompressible Navier-Stokes Equations,"AIAA Paper 85-1689, 1985.

Rogers, S. E. and Kwak, D., "An Upwind Differencing Scheme For the Time-Accurate Incompressible Navier-Stokes Equations," AIAA Paper 88-2583, 1988a.

Rogers, S. E. and Kwak, D., "An Upwind Differencing Scheme for the Steady-State Incompressible Navier-Stokes Equations," NASA TM 101051, 1988b. Also submitted to Appl. Num. Math.

Rogers, S. E. and Kwak, D., "Numerical Solution of the Incompressible Navier-Stokes Equations for Steady and Time-Dependent Problems,"AIAA Paper 89-0463, 1989.

Rosenfeld, M., Kwak, D. and Vinokur, M., "A Solution Method for the Unsteady and Incompressible Navier-Stokes Equations in Generalized Coordinate Systems,"AIAA Paper 88-0718, 1988.

Rosenfeld, M., and Kwak, D., "Numerical Solution of Unsteady Incompressible Flows in Generalized Moving Coordinate Systems,"AIAA Paper 89-0466, 1989.

Roshko, A., "On the Development of Turbulent Wakes from Vortex Streets,"NACA TN 2913, 1953.

Rudy, D.M., and Strikwerda, J.C., "A Nonreflecting Outflow Boundary Condition for Subsonic Navier-Stokes Calculations,"J. Comp. Phys., vol. 36, pp. 55-70, 1980.

Sharma, L., Ostermier, B., Nguyen, L., Dang, P. and O'connor, G., "Turbulence Measurements in an Axisymmetric Turnaround Duct Air Flow Model, "Rocketdyne Division, Rockwell International, Report RSS-8763, ATU-87-5237, Oct. 1987.

Steger, J. L. and Kutler, P., "Implicit Finite-Difference Procedures for the Computation of Vortex Wakes,"AIAA J. vol. 15, no. 4, pp. 581-590, Apr. 1977.

Steger, J. L., and Warming, R. F., "Flux Vector Splitting of the Inviscid Gasdynamic Equations With Application to Finite Difference Methods," J. Comp. Phys., vol. 40, No. 2, pp. 263-293, 1981.

Tarbell, J. M., Gunshinan, J. P., Geselowitz, D. B., Rosenburg, G., Shung, K. K., and Pierce, W. S., "Pulse Ultrasonic Doppler Velocity Measurements Inside a Left Ventricular Assist Device," J. Biomech. Engr., Trans. ASME, vol. 108, pp. 232–238, 1986.

Taylor, A.M.K.P., Whitelaw, J.H., and Yianneskis, M., "Measurements of Laminar and Turbulent Flow in a Curved Duct with Thin Inlet Boundary Layers,"NASA CR 3367, Jan. 1981.

Taylor, A.M.K.P., Whitelaw, J.H., and Yianneskis, M., "Curved Ducts with Strong Secondary Motion: Velocity Measurements of Developing of Laminar and Turbulent Flow,"J. Fluid Engineering, vol. 104, pp. 350-359, Sep. 1982.

Temam, R., Navier Stokes Equations, Revised Edition., North Holland, 1979.

Thomas, A., "Laminar Juncture Flow - A Visualization Study,"Phys. Fluids Letters, Feb. 1987.

Turkel, E., “Symmetrization of the Fluid Dynamic Matries with Applications”, Mathematics of Computation, vol. 27, pp. 729-736, 1973.

Van Dyke, M., An Album of Fluid Motion, The Parabolic Press, Stanford, Calif., 1982.

Vinokur, M., “An Analysis of Finite–Difference and Finite–Volume Formulations of Conservation Laws”, NASA CR–177416, June 1986.

Warming, R. F., Beam, R. M., and Hyett, B. J., “Diagonalization and Simultaneous Symmetrization of the Gas-Dynamic Matrices,”Mathematics of Computation, vol. 29, pp. 1037-1045, 1975.

Wattendorf, F.L., “A Study of the Effect of Curvature on Fully Developed Turbulent Flow,”Proc. Roy. Soc., Vol. 148, 1935.

White, F. M., Viscous Fluid Flow, McGraw-Hill, New York, p. 123, 1974.

Yanenko, N.N., The Method of Fractional Steps, Springer-Verlag, Berlin, 1971.

Yang, R-J, Chang, J. L. C., and Kwak, D., “A Navier-Stokes Simulation of the Spce Shuttle Main Engine Hot Gas Manifold , ”AIAA Paper 87-0368, 1987.

Yee, H. C., “Linearized Form of Implicit TVD Schemes For The Multidimensional Euler and Navier-Stokes Equations,” Comp. and Maths. with Appls., vol. 12A, Nos. 4/5, pp. 413–432, 1986.

Yoon, S. and Jameson, A., “An LU-SSOR Scheme for the Euler and Navier-Stokes Equations,”AIAA Paper 87-0600, 1987.

Yoon, S. and Kwak, D., “Artificial Dissipation Models for Hypersonic External Flow,’AIAA Paper 88-3708, 1988.

Yoon, S. and Kwak, D., “LU-SGS Implicit Algorithm for Three-Dimensional Incompressible Navier-Stokes Equations with Source Term,”AIAA Paper 89-1964, 1989.

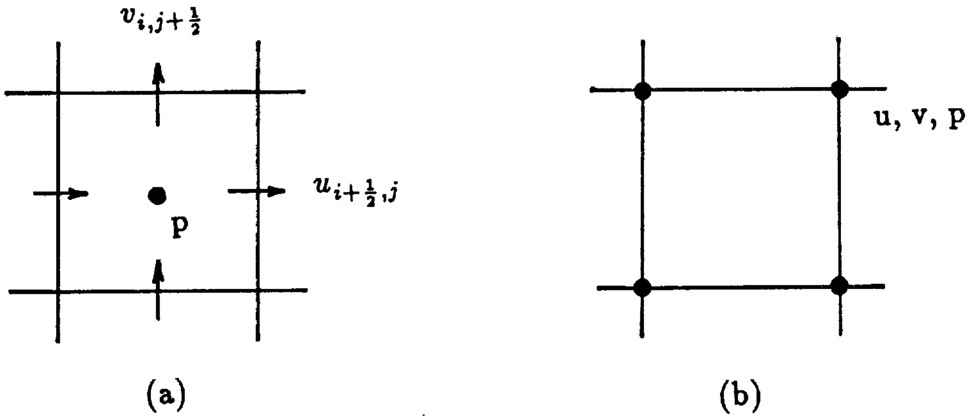


Figure 2.1 Grid arrangement: (a) staggered, (b) regular.

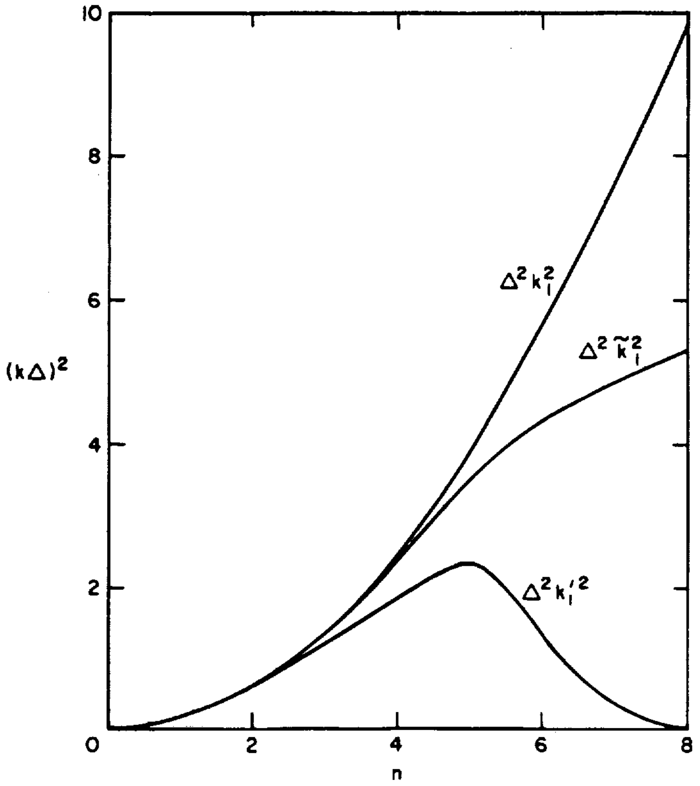


Figure 2.2 Comparison of ∇^2 operators for 16 equally spaced mesh points (see equation 2.10 and 2.11):
$k_1{}^2$ = exact
$\bar{k}_1^2$ = fourth order $\frac{\delta^2}{\delta x_1{}^2}$
$k'_1{}^2$ = fourth order DG operator $(\frac{\delta}{\delta x_1}\frac{\delta}{\delta x_1})$

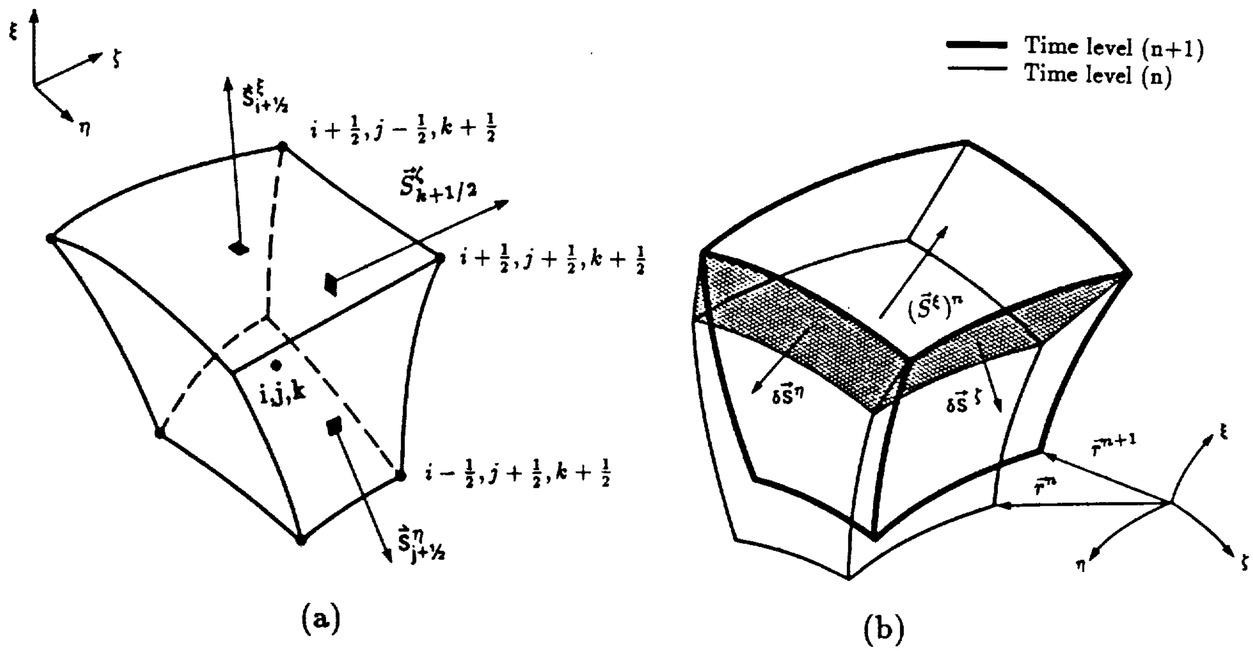


Figure 4.1 Staggered grid in generalized coordinates:
(a) primary cell, (b) moving grid.

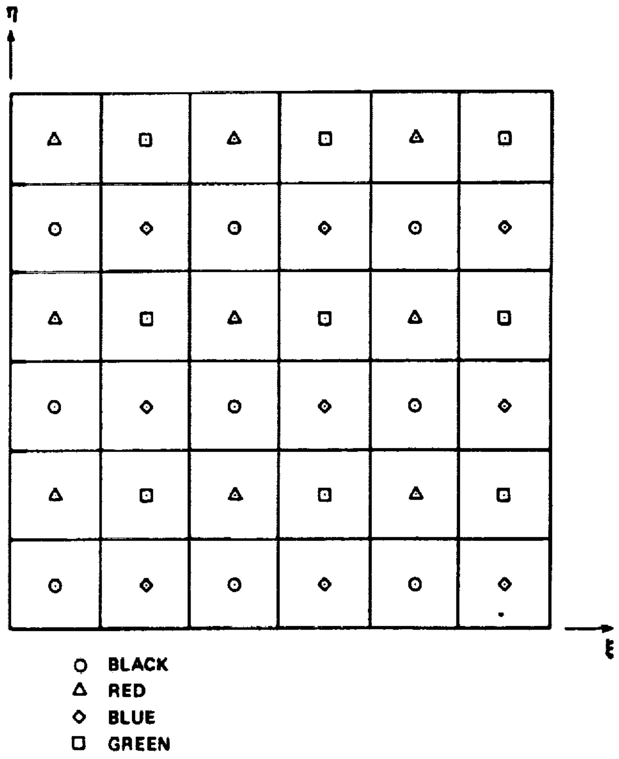


Figure 4.2 Four color arrangement for Poisson solver.

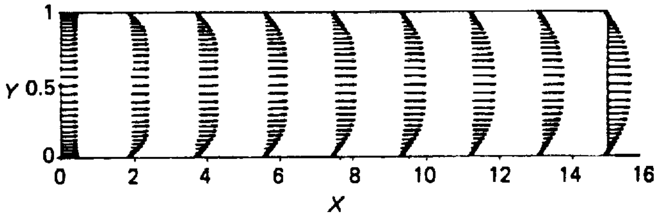


Figure 5.1 Channel flow : velocity profile at Re=1000, β =5, τ=0.1

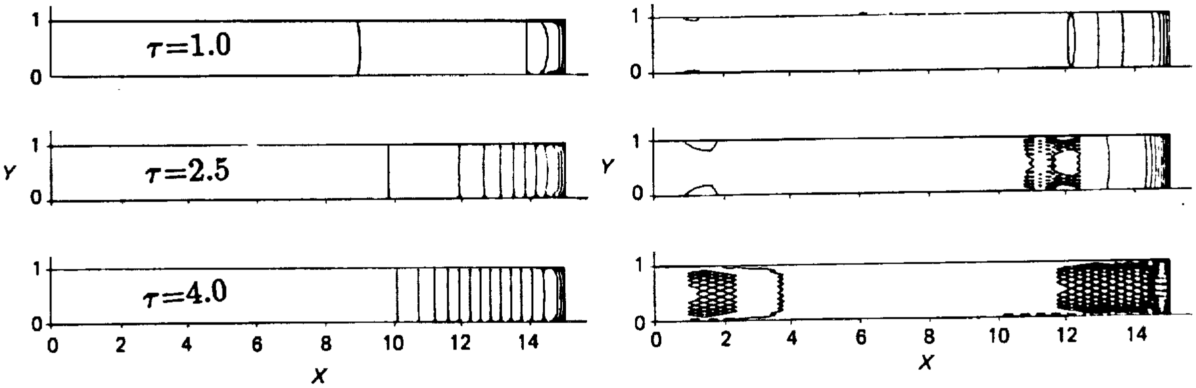


Figure 5.2 Pressure contours for developing channel flow: (a) β=5, (b) β=0.1

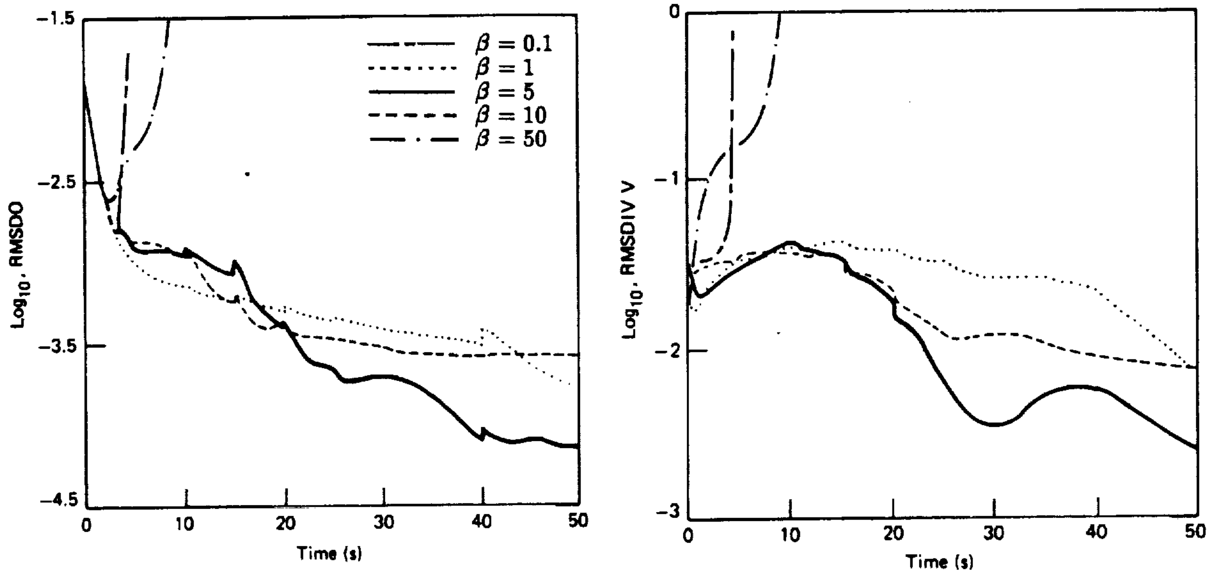


Figure 5.3 Convergence history for channel flow at Re=1000:
(a) RMSDQ, (b) RMSDIV

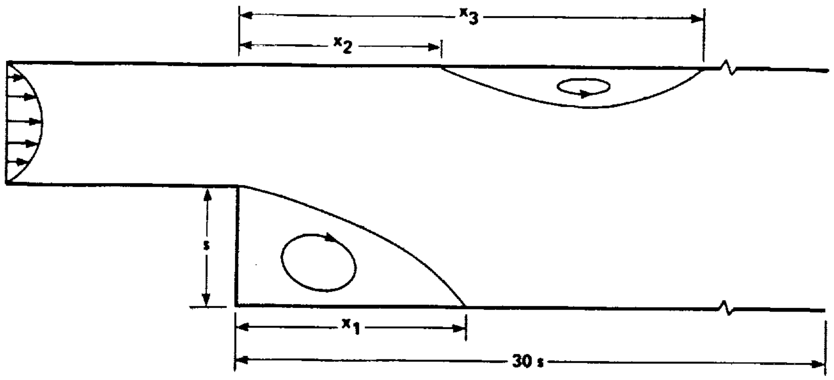


Figure 5.4 Geometry of backward-facing step flow problem.

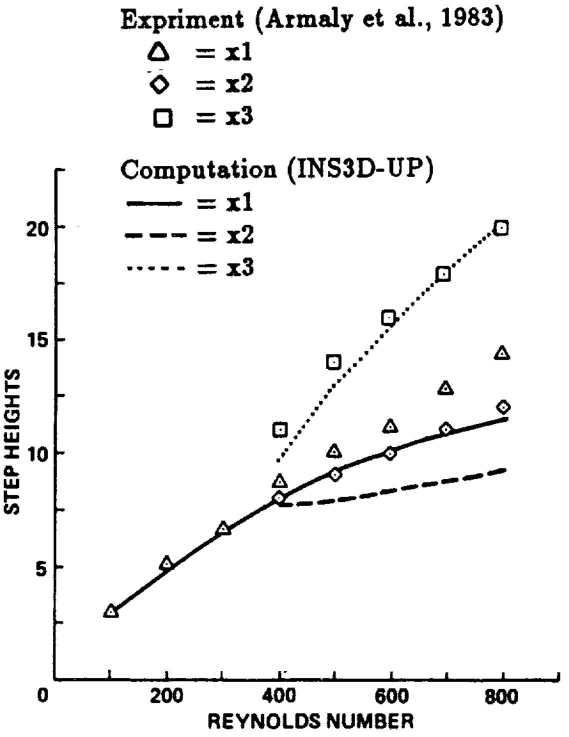


Figure 5.5 Separation and reattachment lengths for the flow over a backward-facing step.

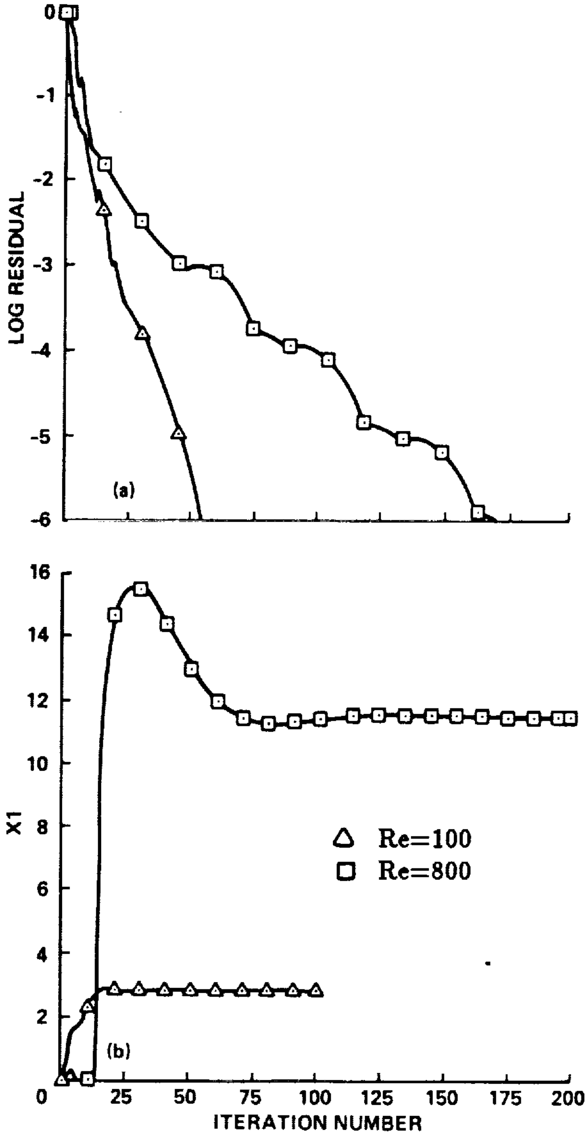


Figure 5.6 Convergence history for the flow over a backward-facing step : (a) residual, (b) primary reattachment length.

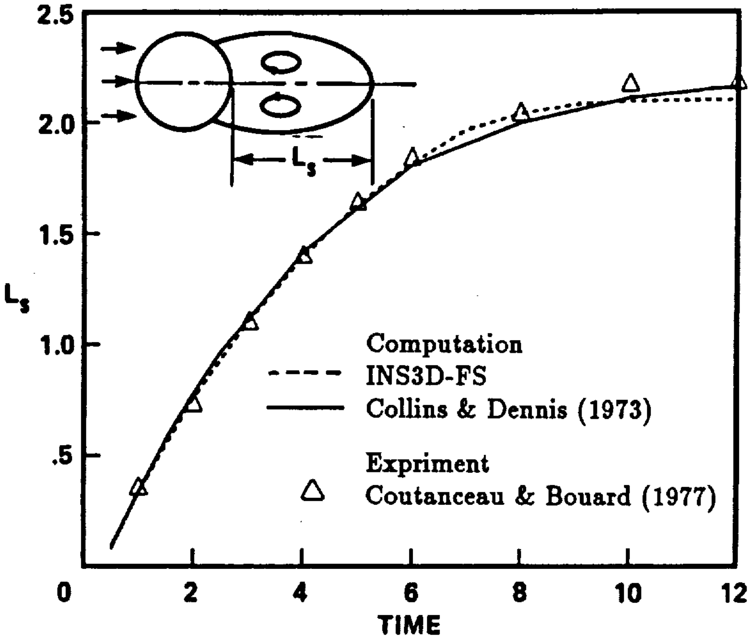


Figure 5.7 Time evolution of separation length for flow over a circular cylinder at Re=40.

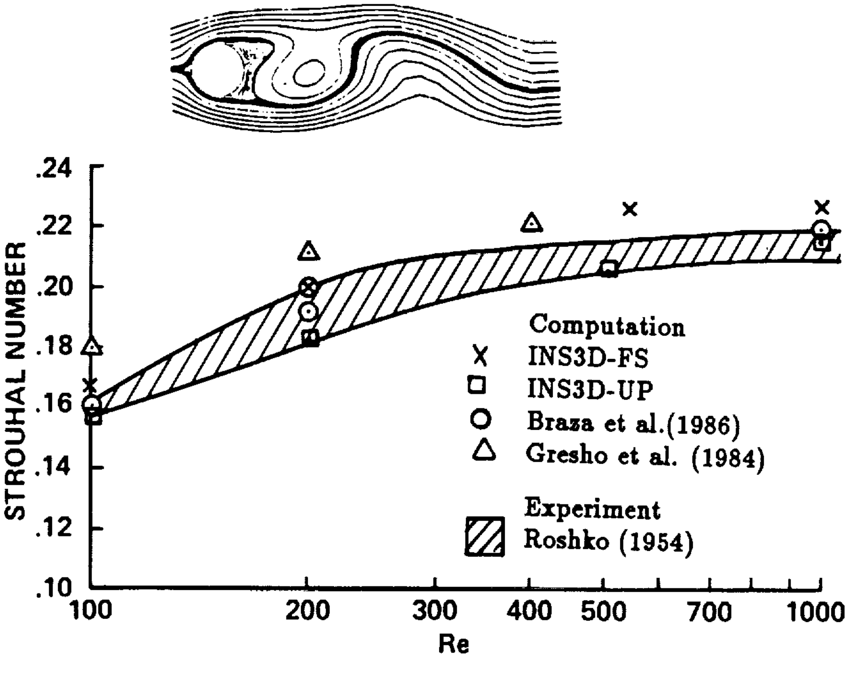


Figure 5.8 Vortex shedding from a circular cylinder.

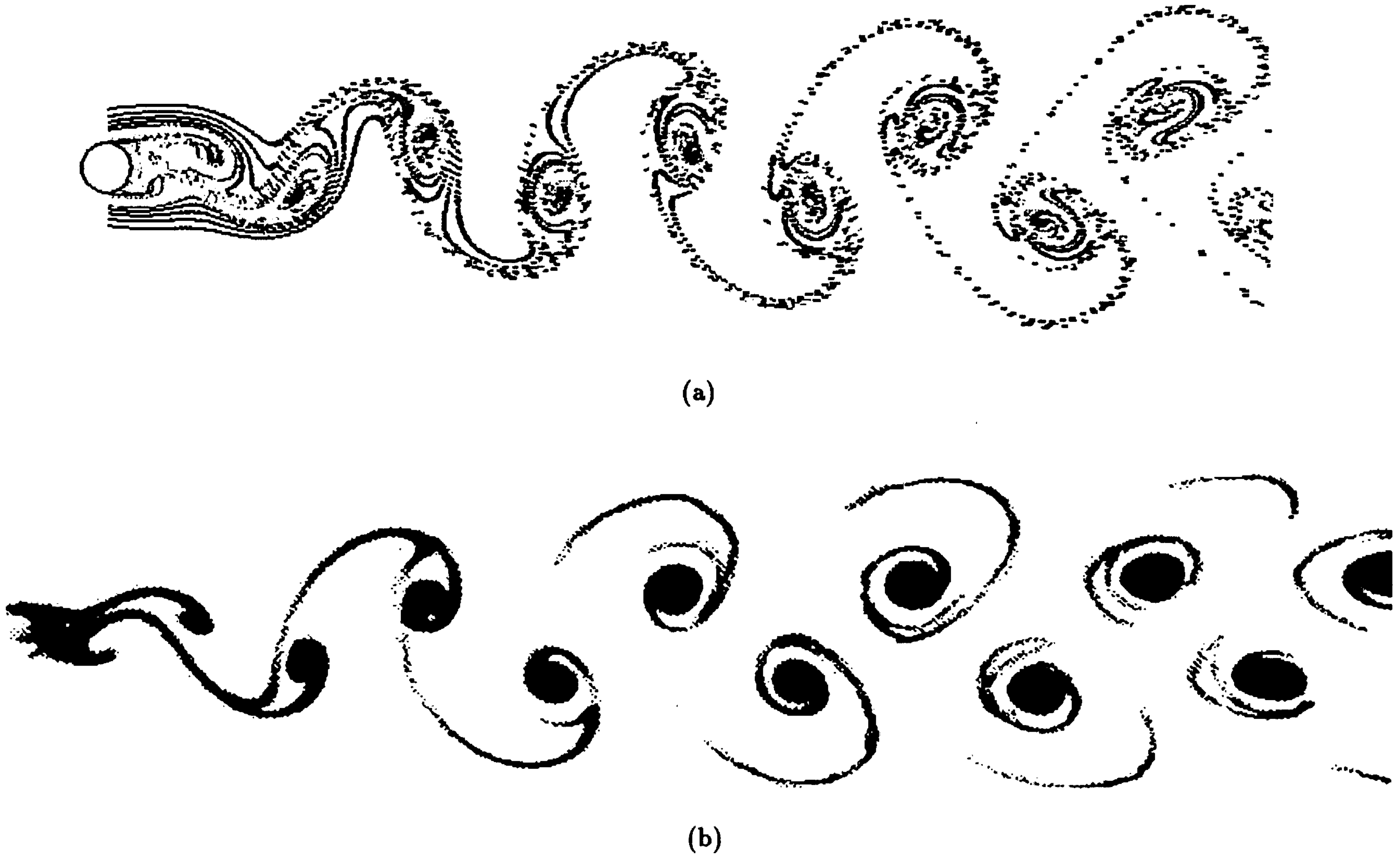


Figure 5.9 Karman vortex street behind a circular cylinder at Re=105: (a) computed (INS3D-UP), (b) experiment (S. Taneda, see Van Dyke, 1982)

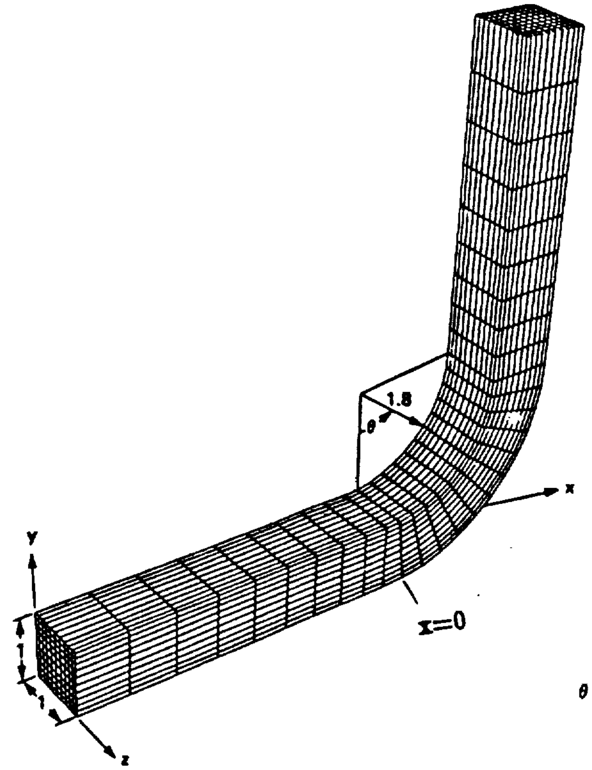


Figure 5.10 Geometry of a square duct with 90° bend.

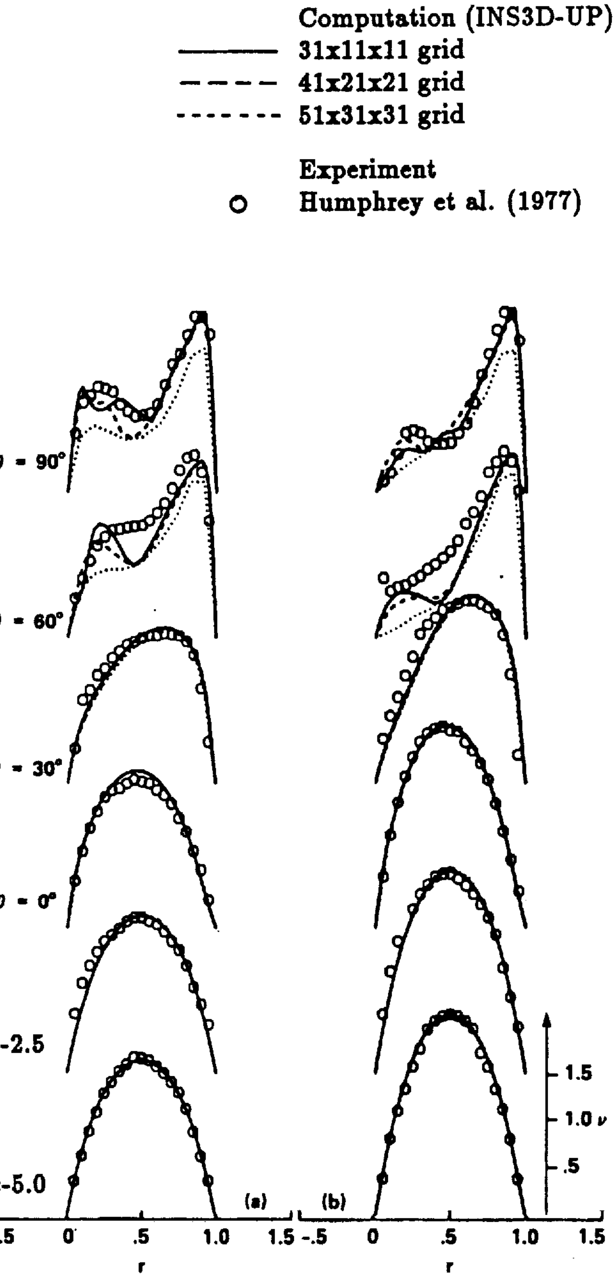


Figure 5.11 Streamwise velocity profile for flow through a 90° bend: (a) xy plane at z=0.25, (b) xy plane at z=0.5.

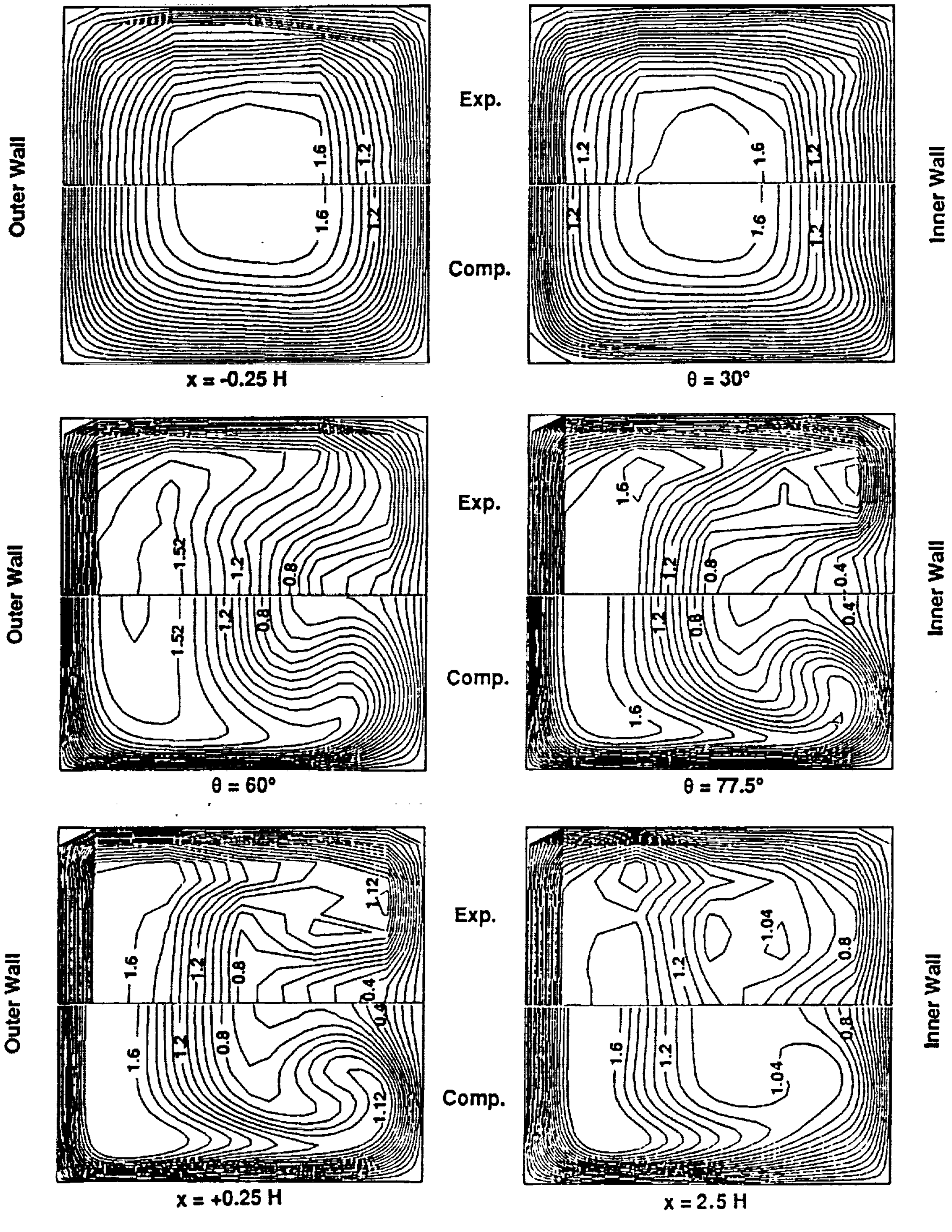
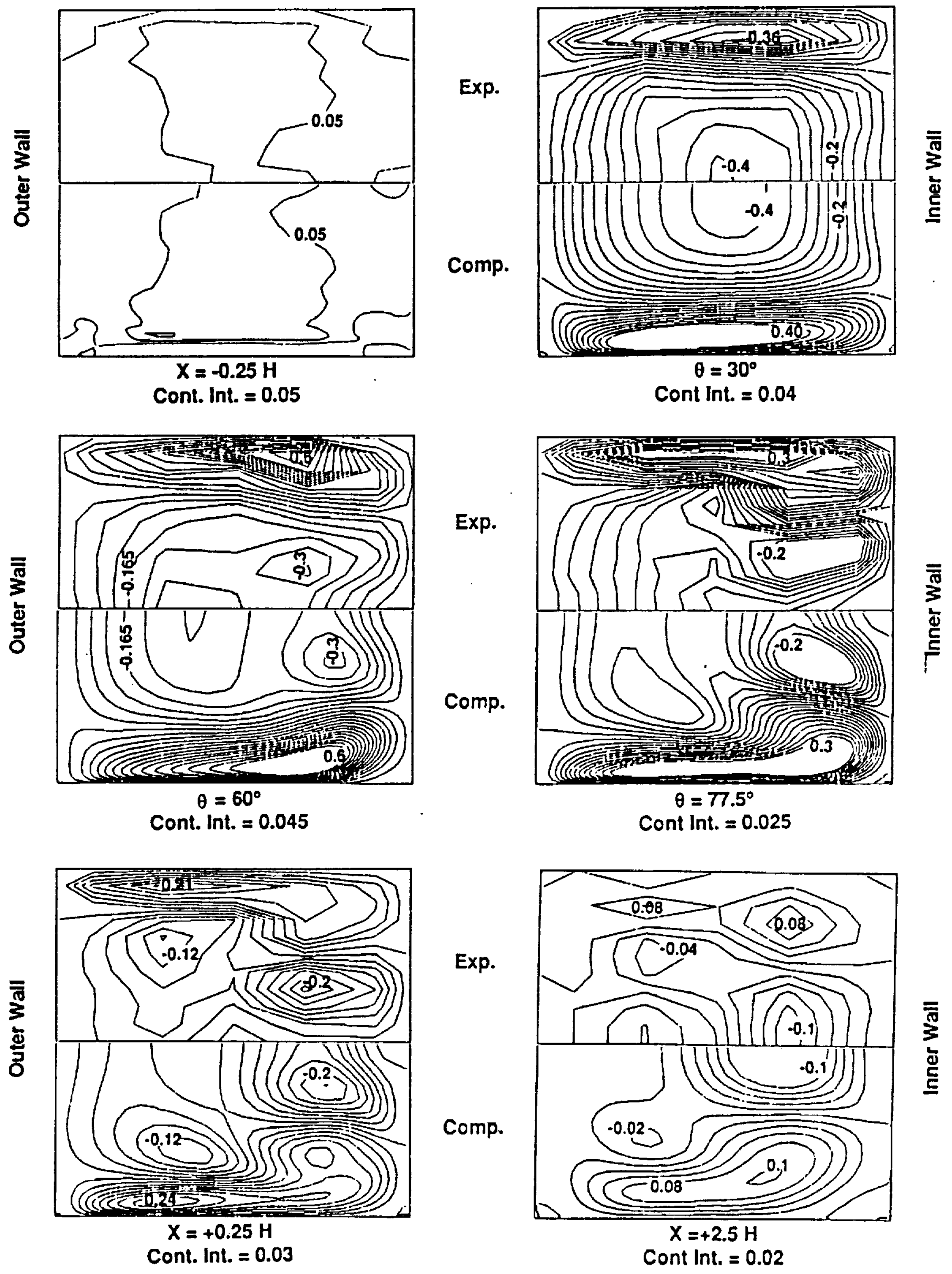


Figure 5.12 Predicted axial flow in 90° bend (McConnaughey et al., 1989, using INS3D) compared with the data of Taylor et al. (1981).



X = -0.25 H
Cont. Int. = 0.05

θ = 30°
Cont Int. = 0.04

θ = 60°
Cont. Int. = 0.045

θ = 77.5°
Cont Int. = 0.025

X = +0.25 H
Cont. Int. = 0.03

X =+2.5 H
Cont Int. = 0.02

Figure 5.13 Predicted radial flow in 90° bend (McConnaughey et al., 1989, using INS3D) compared with the data of Taylor et al. (1982).

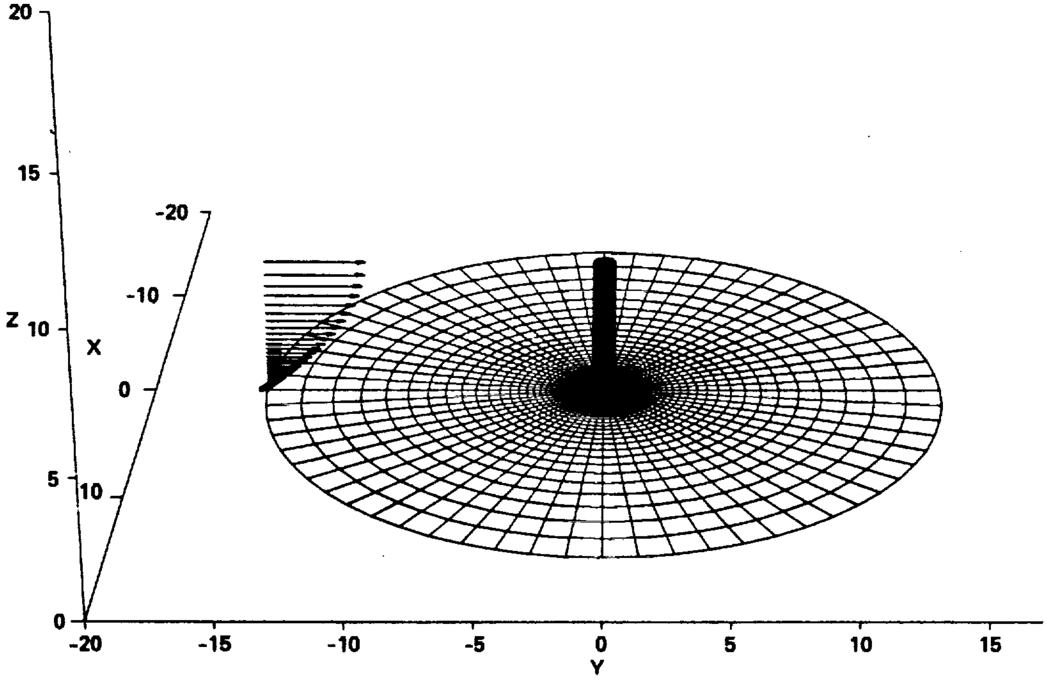


Figure 5.14 Grid for a single post on flat plate.

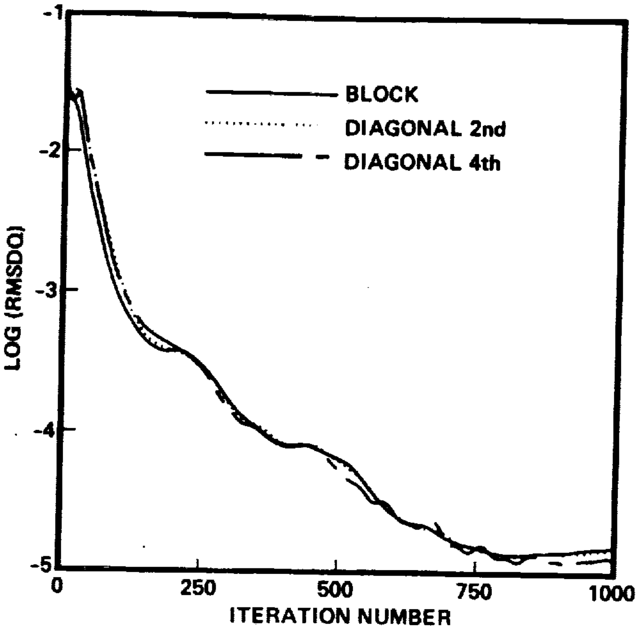


Figure 5.15 Convergence history for flow around a single post on flat plate at Re=1000 (INS3D).

C - 2

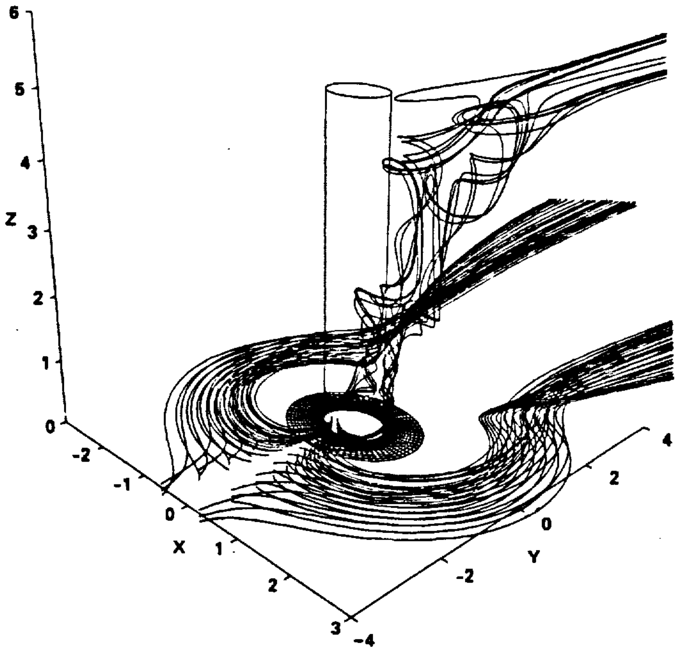


Figure 5.16 Particle traces for a single post at Re=1000.

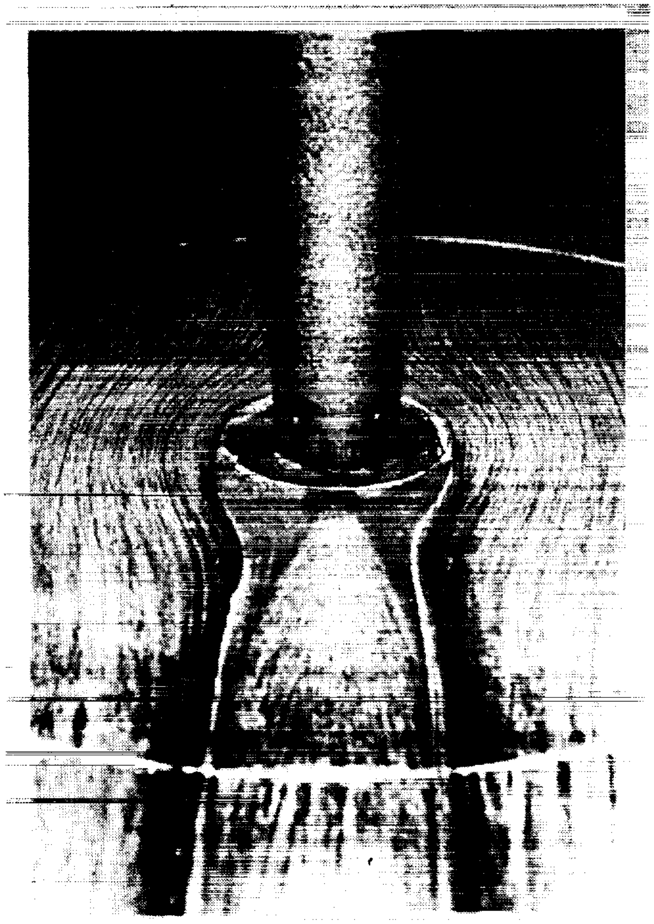


Figure 5.17 Oil flow visualization around a single post at Re=1.85x10^5
(by G. Schewe, DFVLR, 1985).

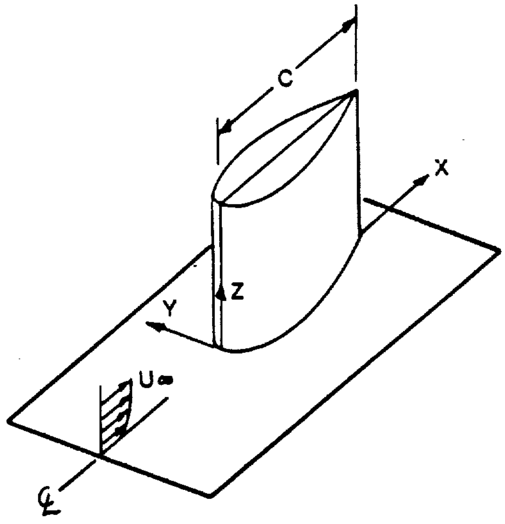


Figure 5.18 Wing on falt plate.

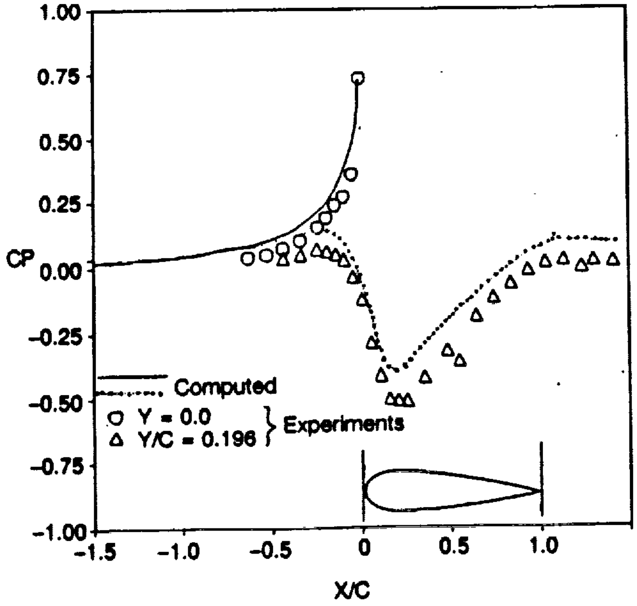


Figure 5.19 Pressure on flat plate along y=constant at Re=5.x10^5.

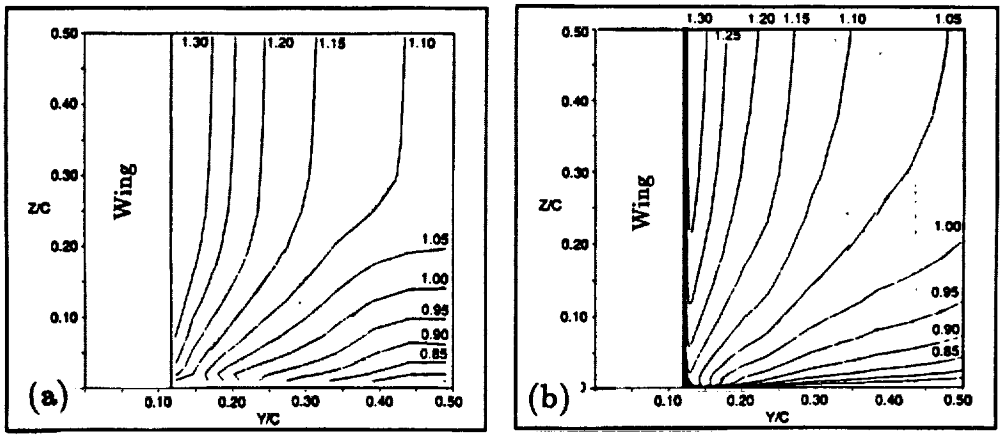


Figure 5.20 Velocity contours for wing-flat plate at x/c=0.18 and Re=5x10^5 : (a) experiment (Dickinson, 1986), (b) computation (Burke, 1989; INS3D)

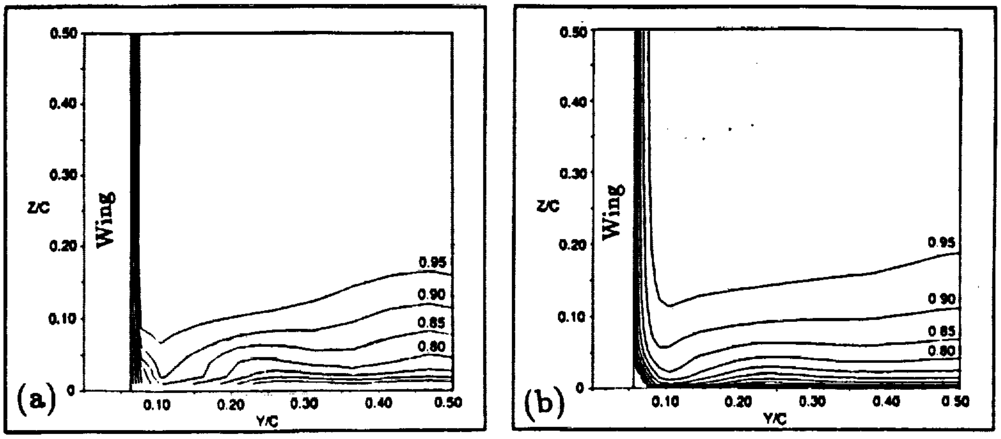


Figure 5.21 Velocity contours for wing-flat plate at x/c=0.75 and Re=5x10^5 : (a) experiment (Dickinson, 1986), (b) computation (Burke, 1989; INS3D)

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

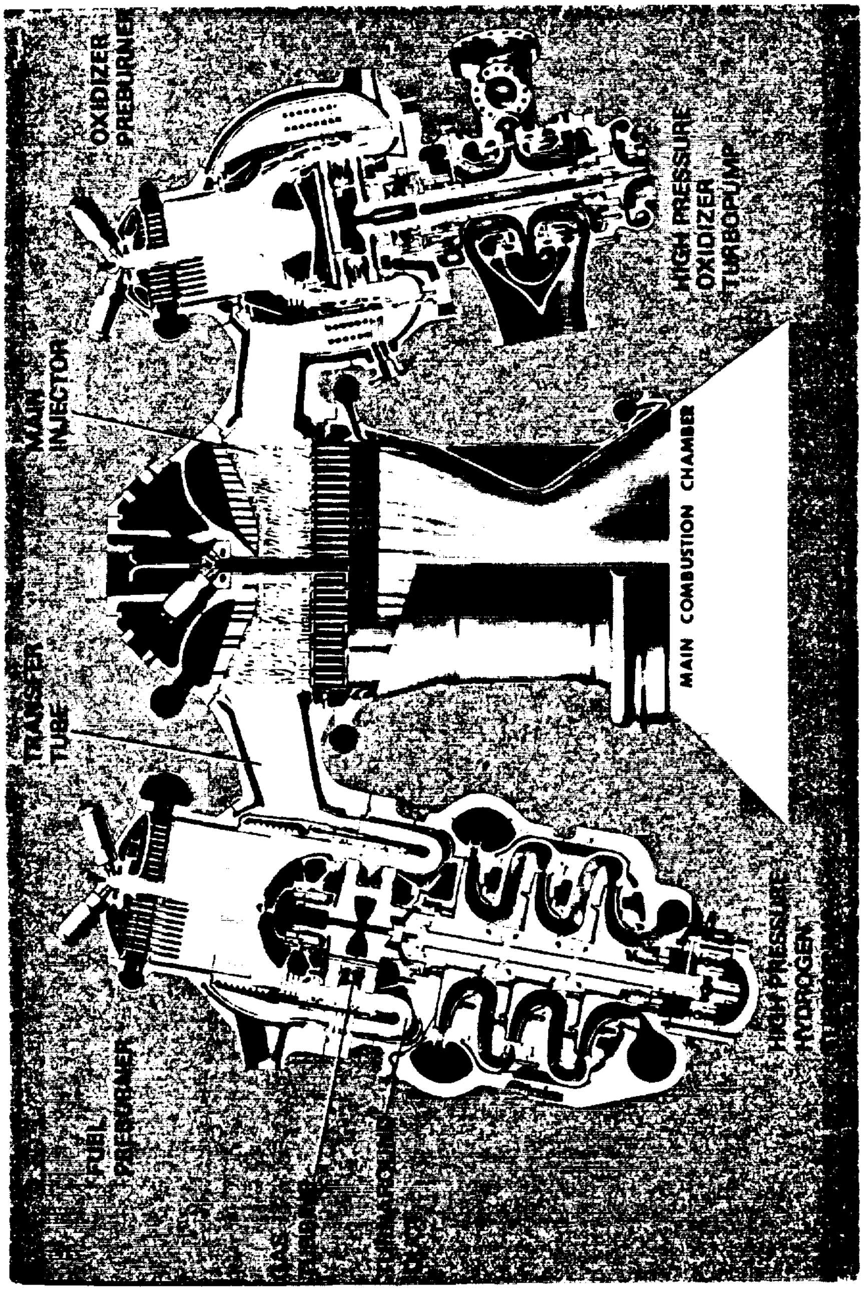


Figure 6.1. SSME power head component arrangement

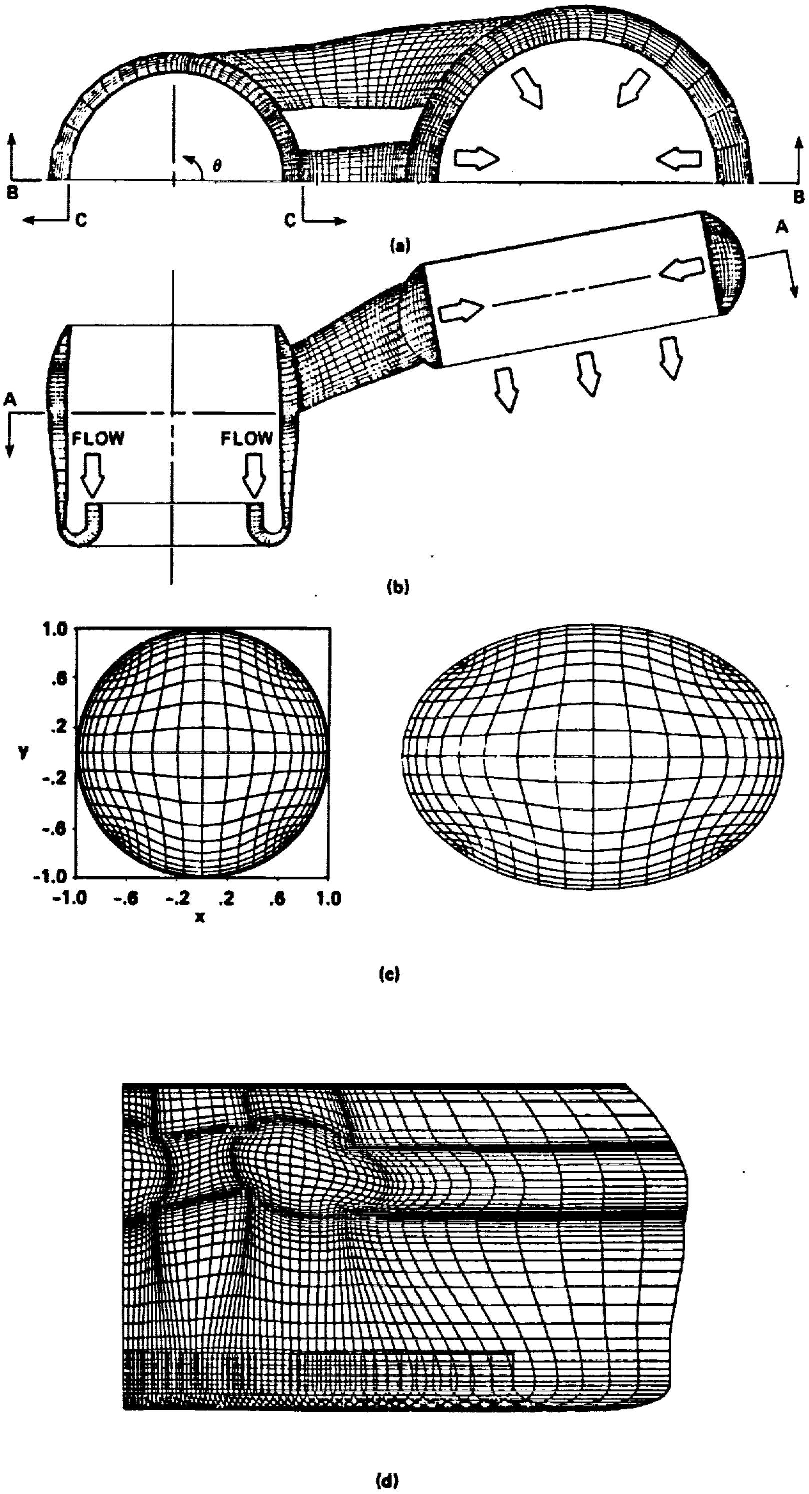


Figure 6.2. Grid of the SSME hot-gas manifold.

a) Horizontal view (cross section A-A); b) vertical view (cross section B-B); c) H-grid for circular and elliptic cross section of transfer duct; d) unwrapped surface of annular fuel bowl (cross section C-C).

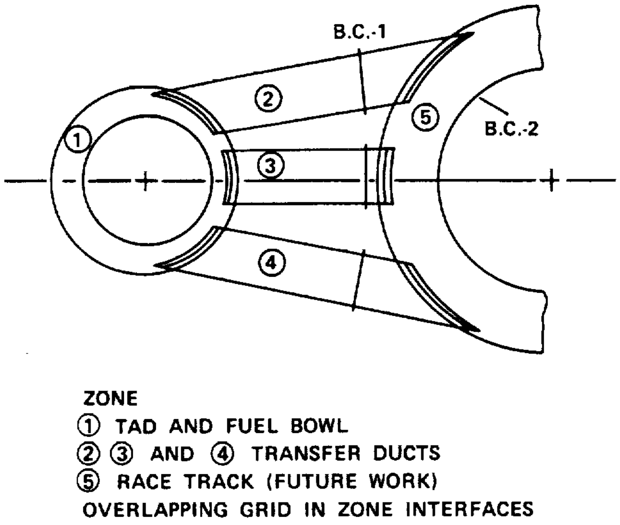


Figure 6.3. Multiple-zone arrangement

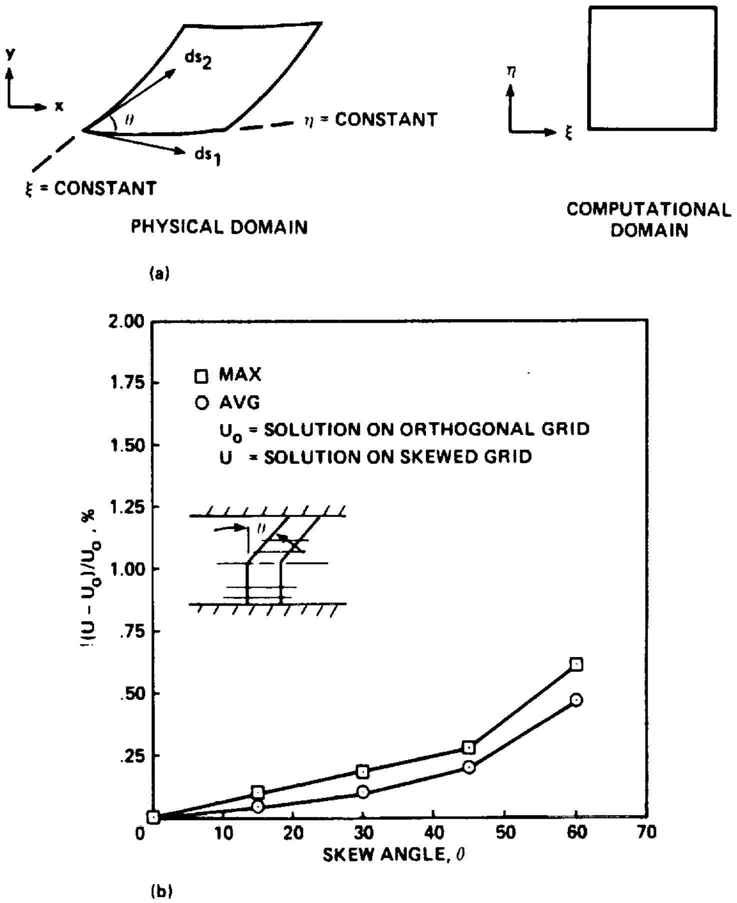


Figure 6.4. Grid effect on channel flow solution at Re=1000.
a) Definition of grid skewness; b) Relative error due to skewness.

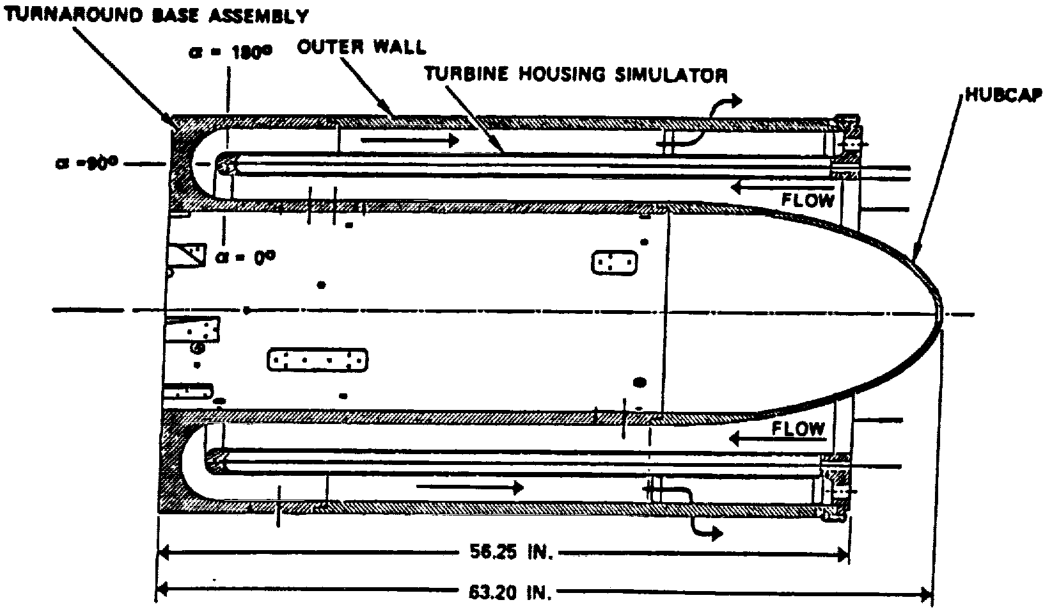
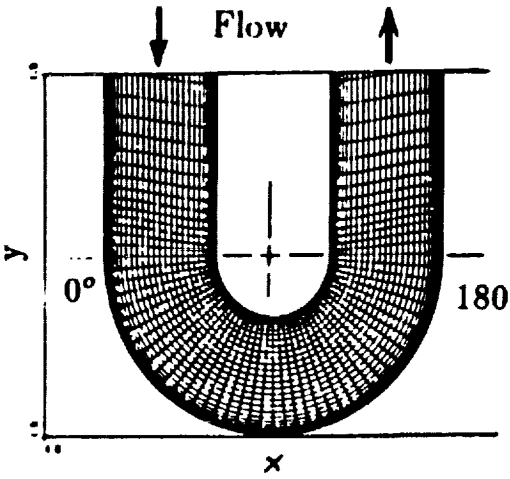


Figure 6.5 A schematic of experimental turn-around duct model.



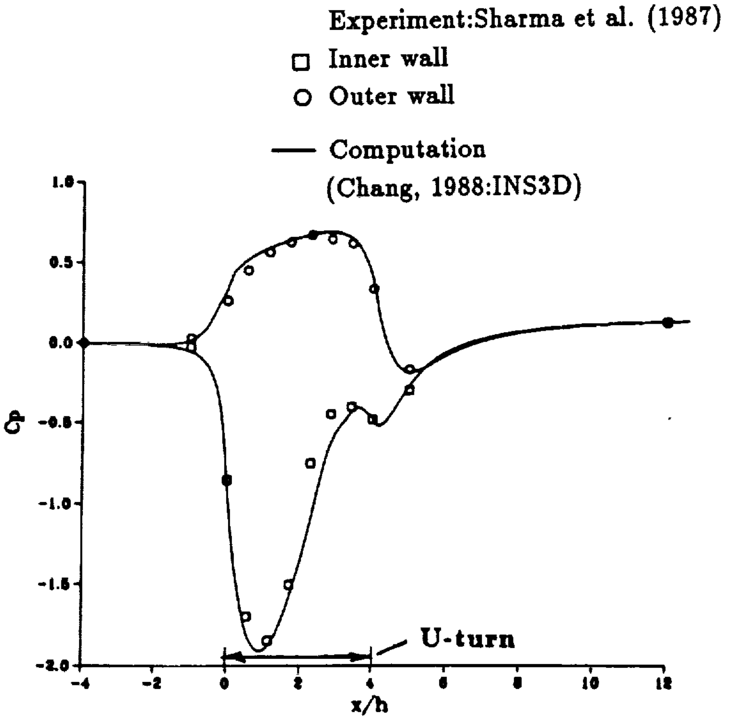


Figure 6.6 Grid in U-turn region. Figure 6.7 Static pressure distribution at Re=10^5.

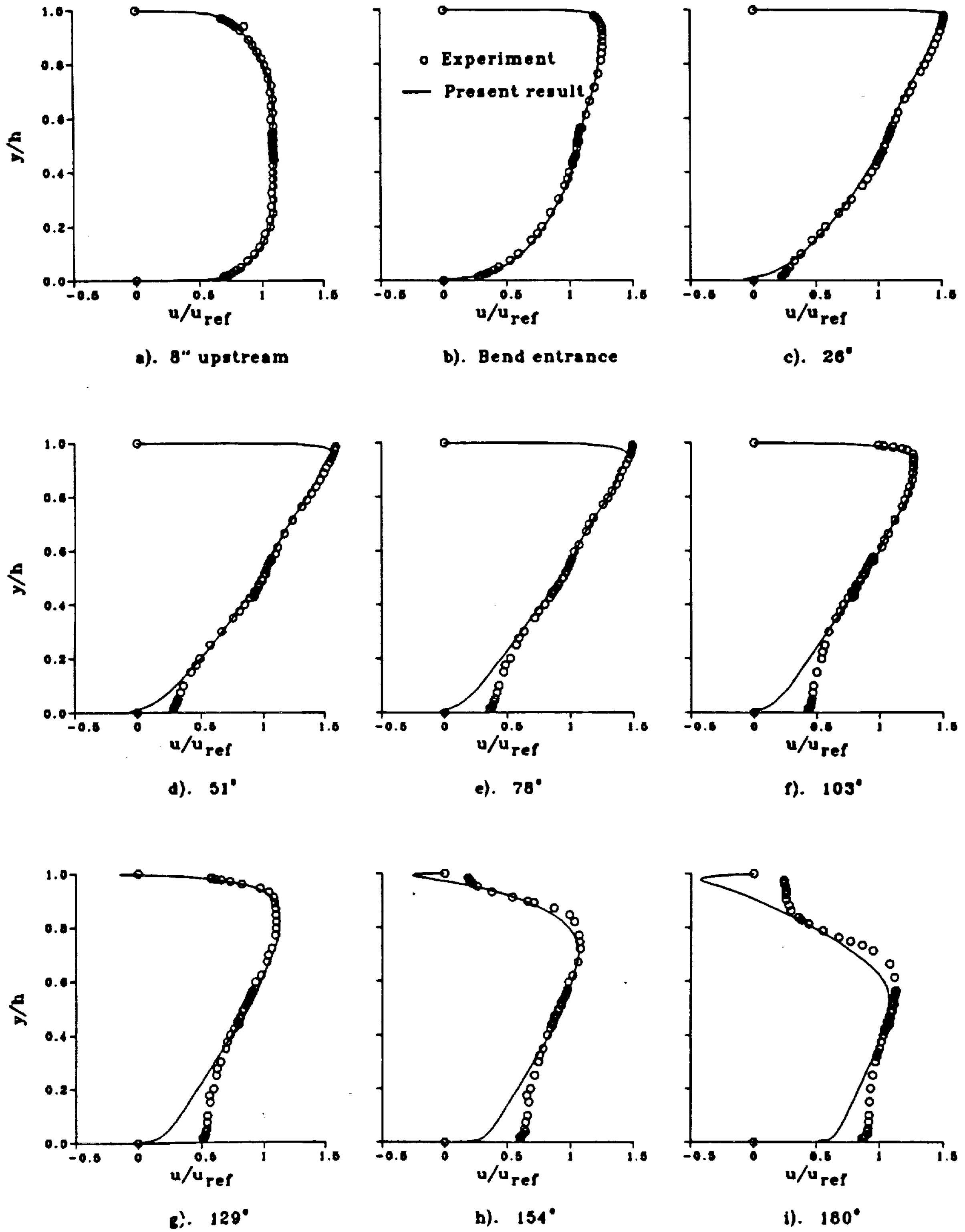


Figure 6.8 Comparison of normalized streamwise velocities.

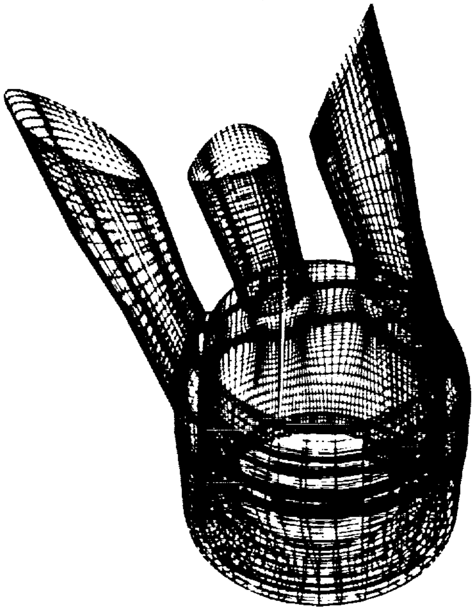
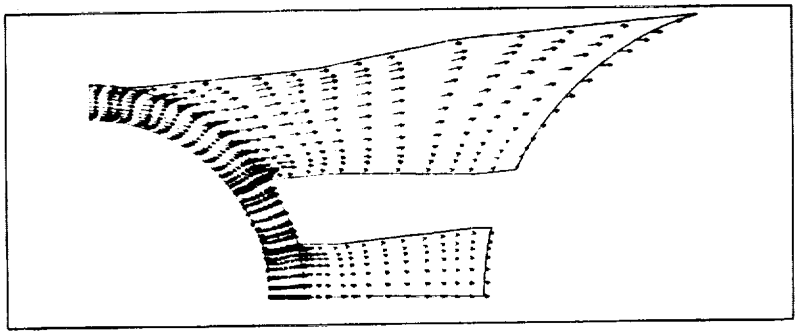
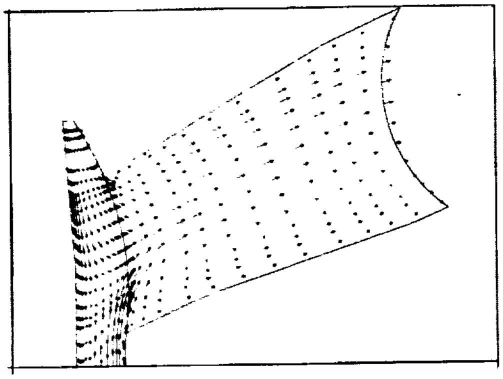


Figure 6.9 Inner and outer surface grid for a three-duct HGM.



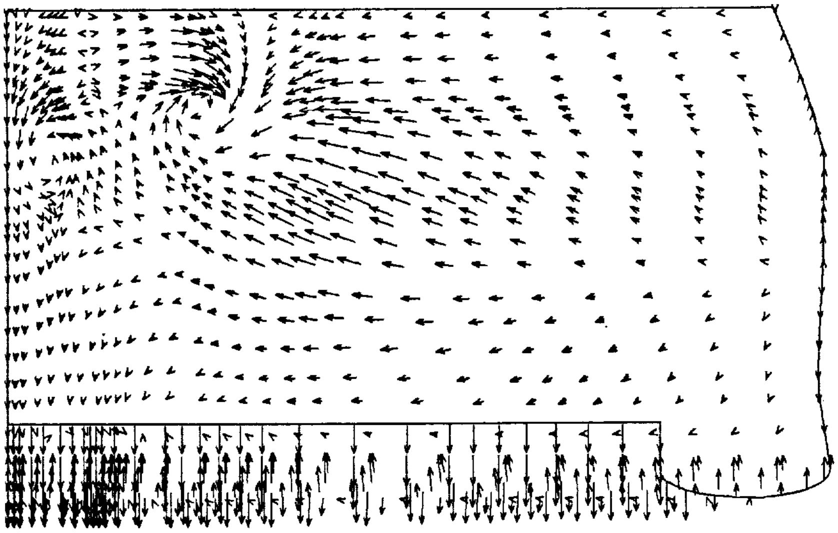
(a)



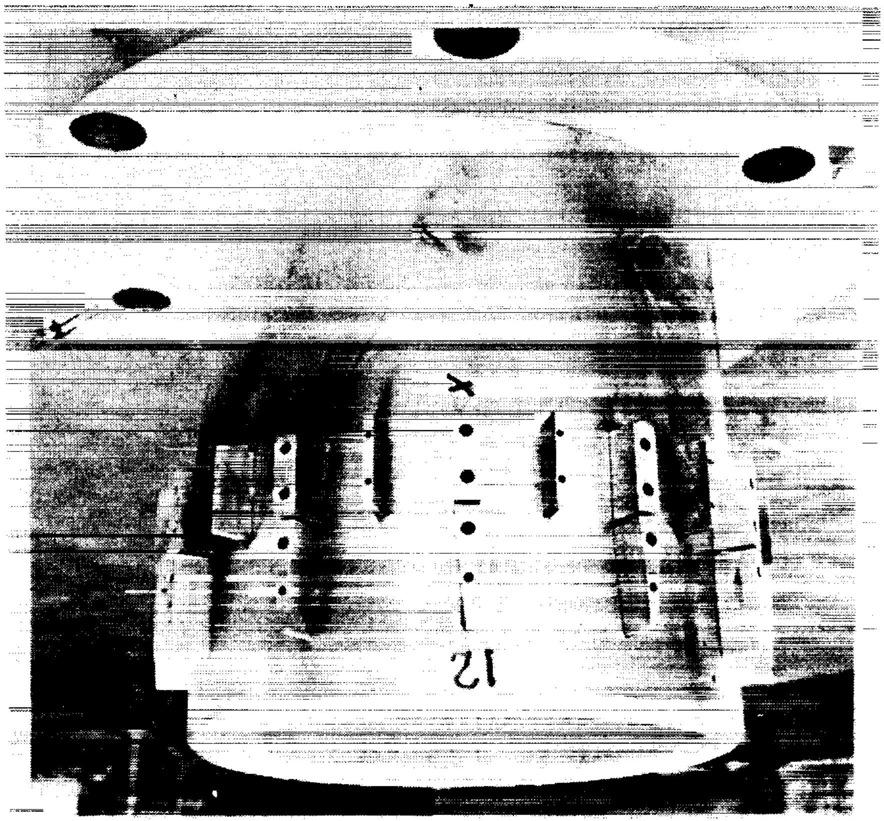
(b)

Figure 6.10 Computed velocity distribution at Re=1000: (a) top view, (b) vertical cross section of center transfer duct.

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH



(a)



(b)

Figure 6.11 Velocity vectors on unwrapped surfaces:
(a) unwrapped inner wall, (b) experiment.

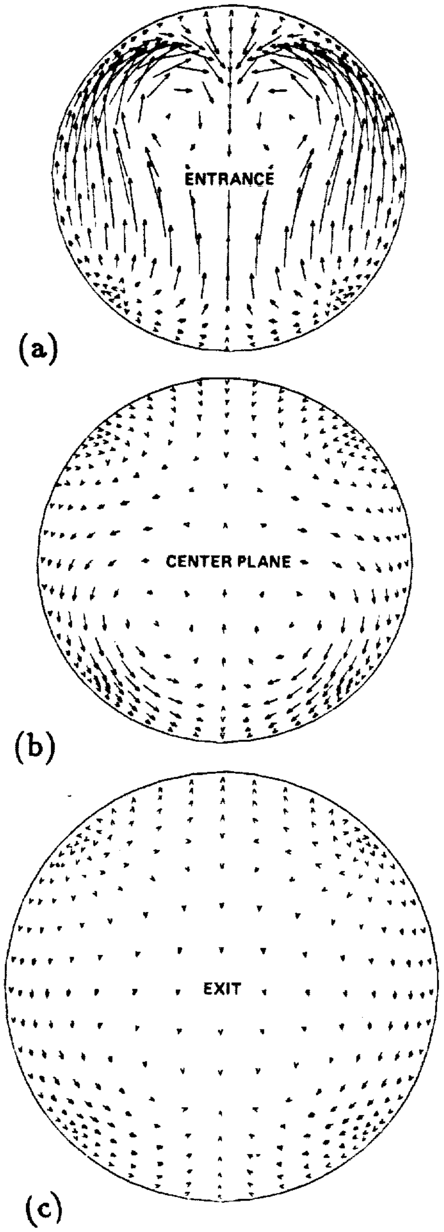


Figure 6.12 Velocity vectors at cross sections of center duct in three-duct Hot Gas Manifold.

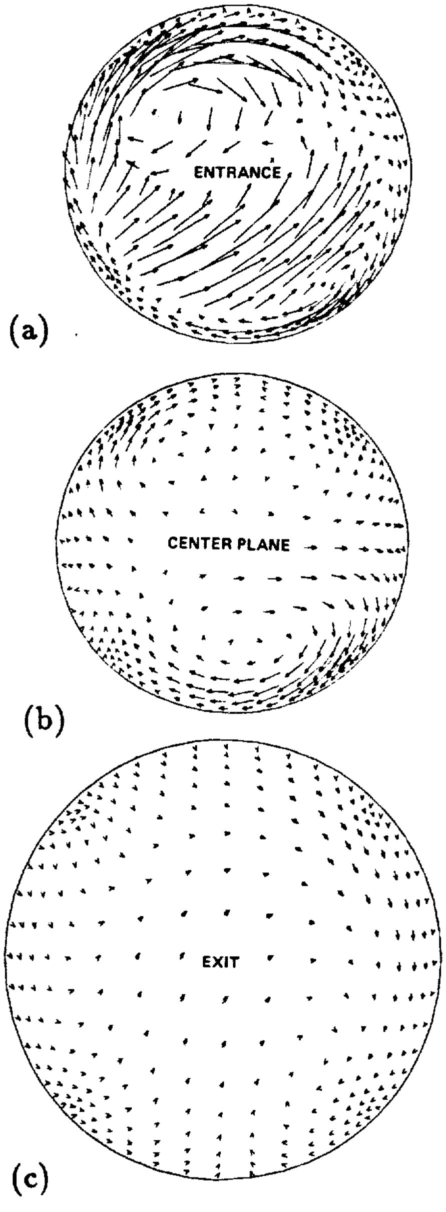


Figure 6.13 Velocity vectors at cross sections of outer duct in three-duct Hot Gas Manifold.

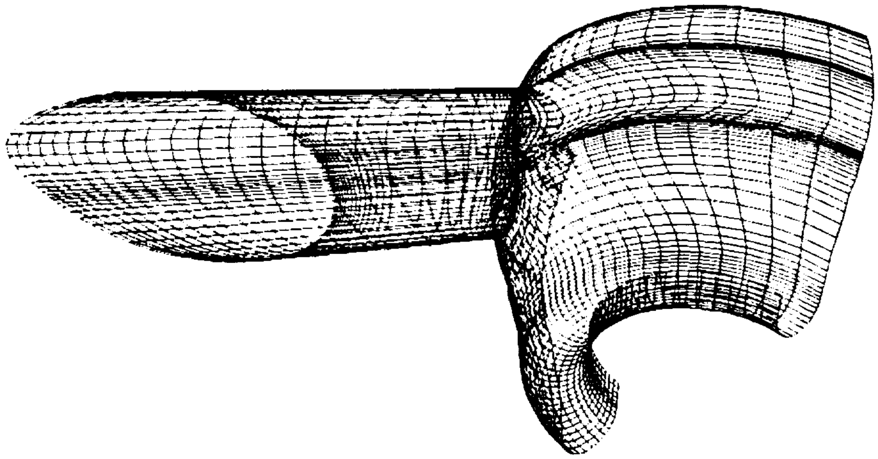


Figure 6.14 Outer-surface grid for a two-duct HGM.

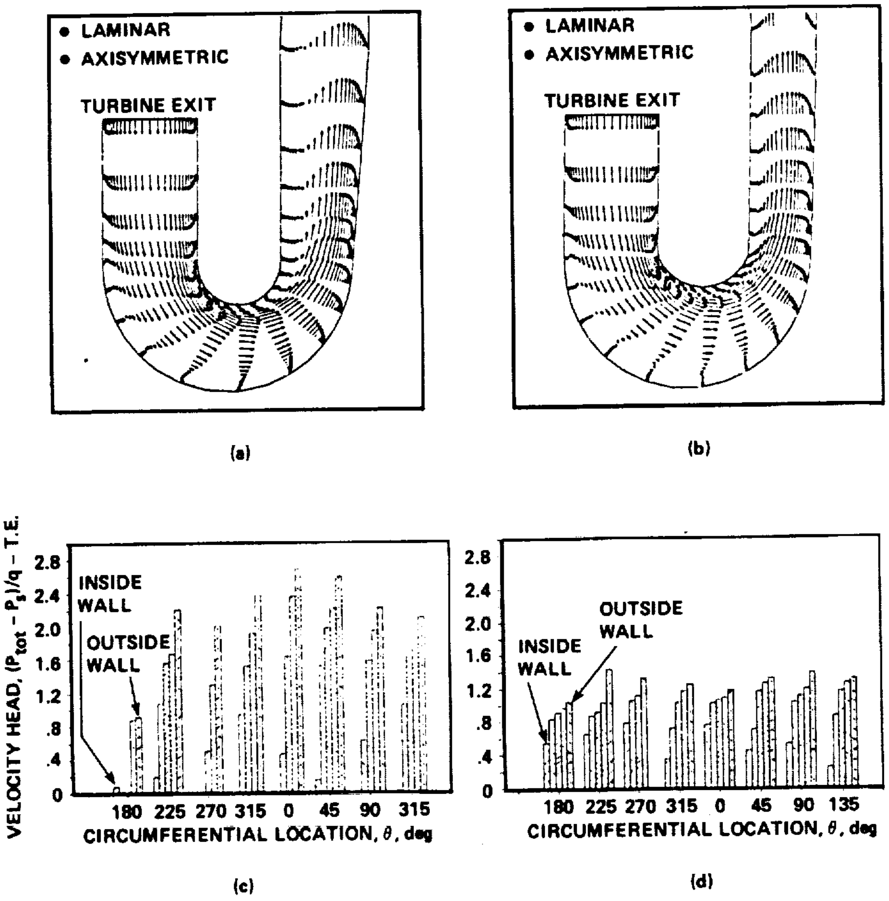


Figure 6.15 Turn-Around-Duct redesign:

a) Computed velocity vectors for the current design;
b) Computed velocity vectors for the new design;
c) Experimental velocity head measurements for the current design (θ defined in Figure 6.2a);
d) Experimental velocity head measurements for the new design.

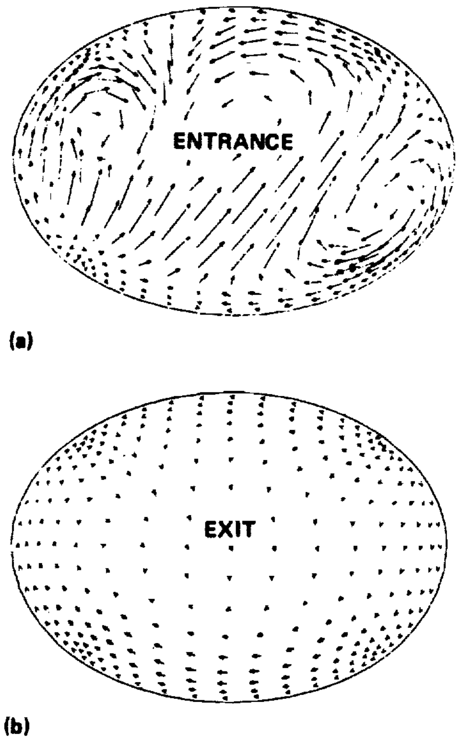


Figure 6.16 Velocity vectors at cross sections of transfer duct in two-duct HGM (laminar:Re=10^3).

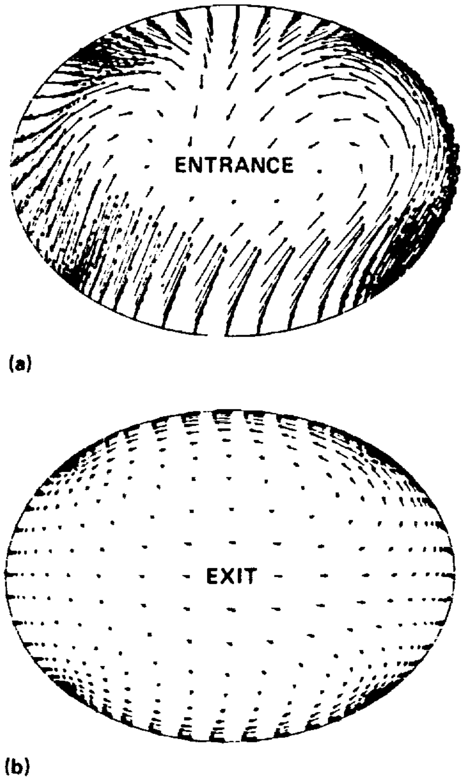


Figure 6.17 Velocity vectors at cross sections of transfer duct in two-duct HGM (turbulent::Re=10^5).

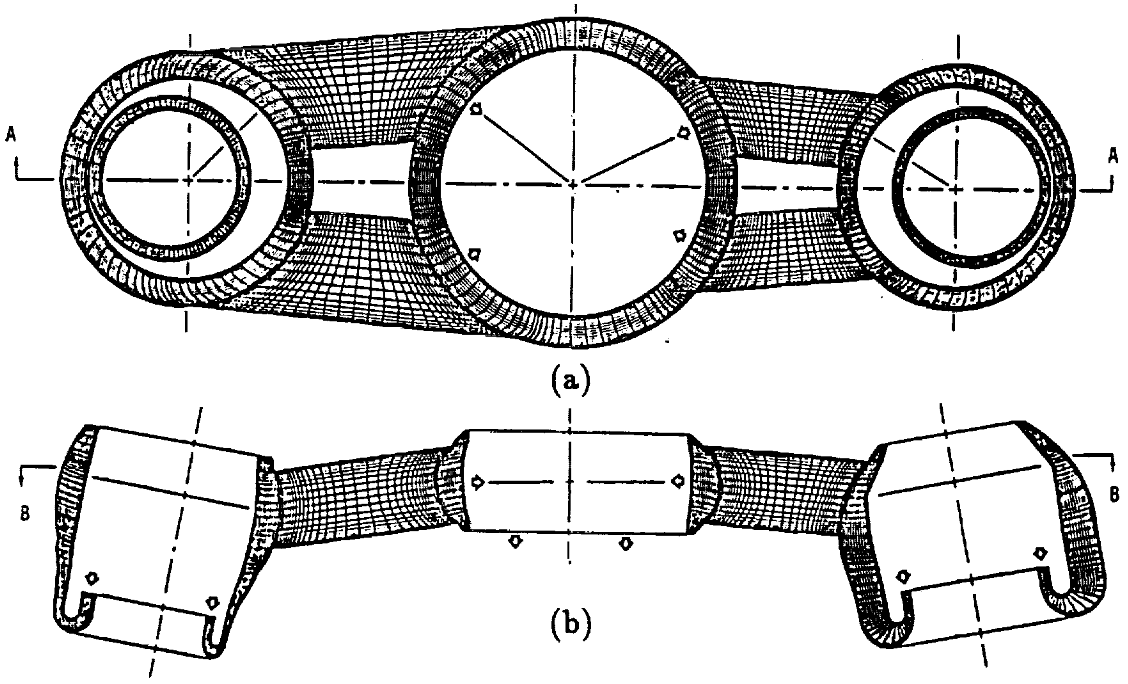


Figure 6.18 Computer model of the new two-duct SSME power head :
(a) vertical cross-section (B-B), (b) horizontal cross-section (A-A).

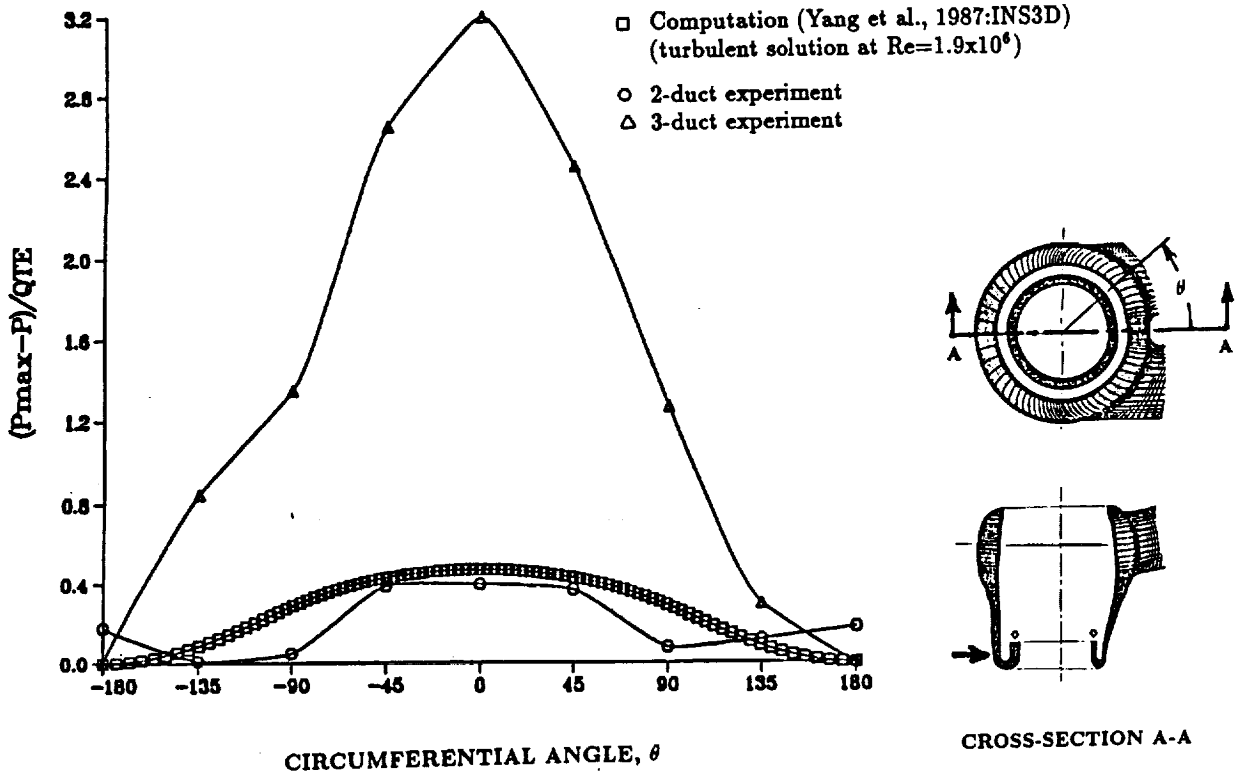


Figure 6.19 Pressure coefficient on fuel bowl outer surface after 180° turn.

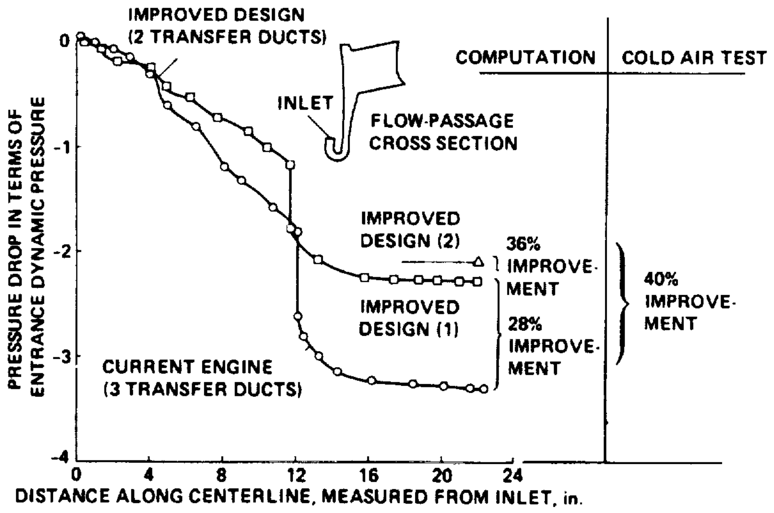


Figure 6.20 Pressure losses in three-duct and two-duct HGM.

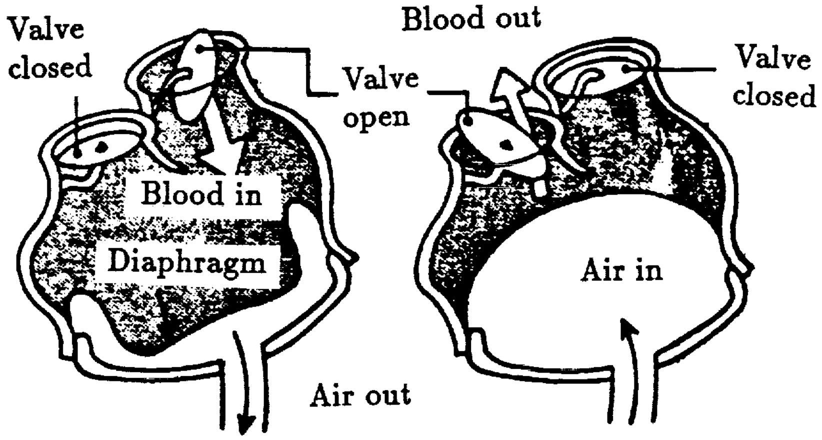
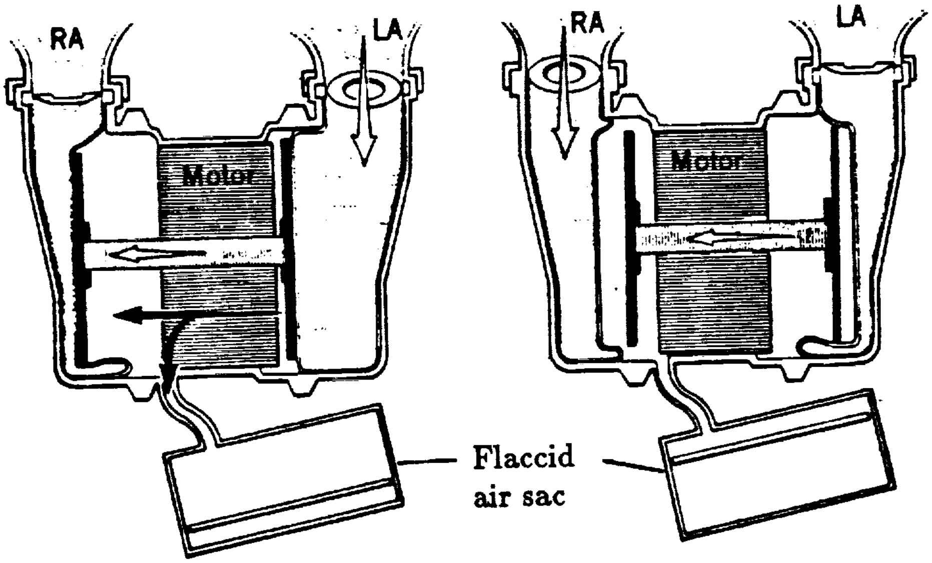


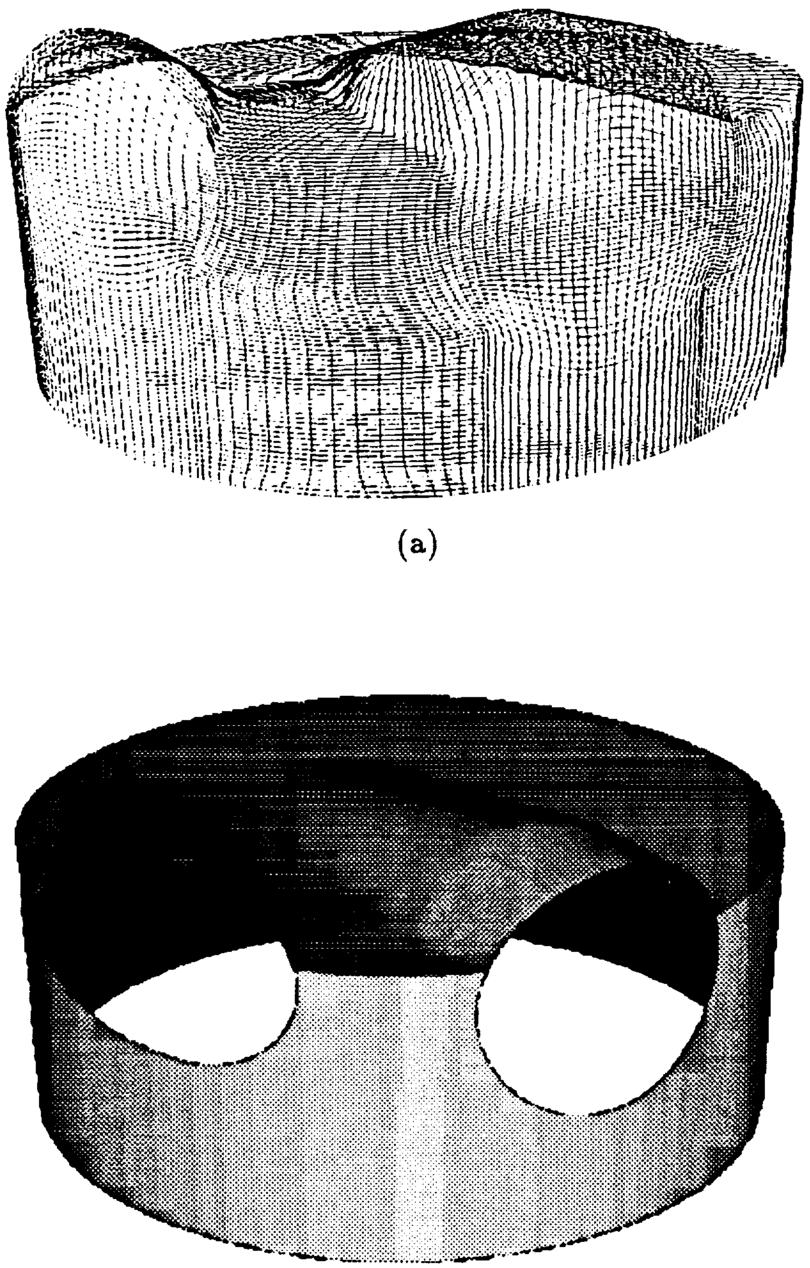
Figure 6.21 Schematic of an artificial heart operated by air pressure.



Left diastole

Left systole completed

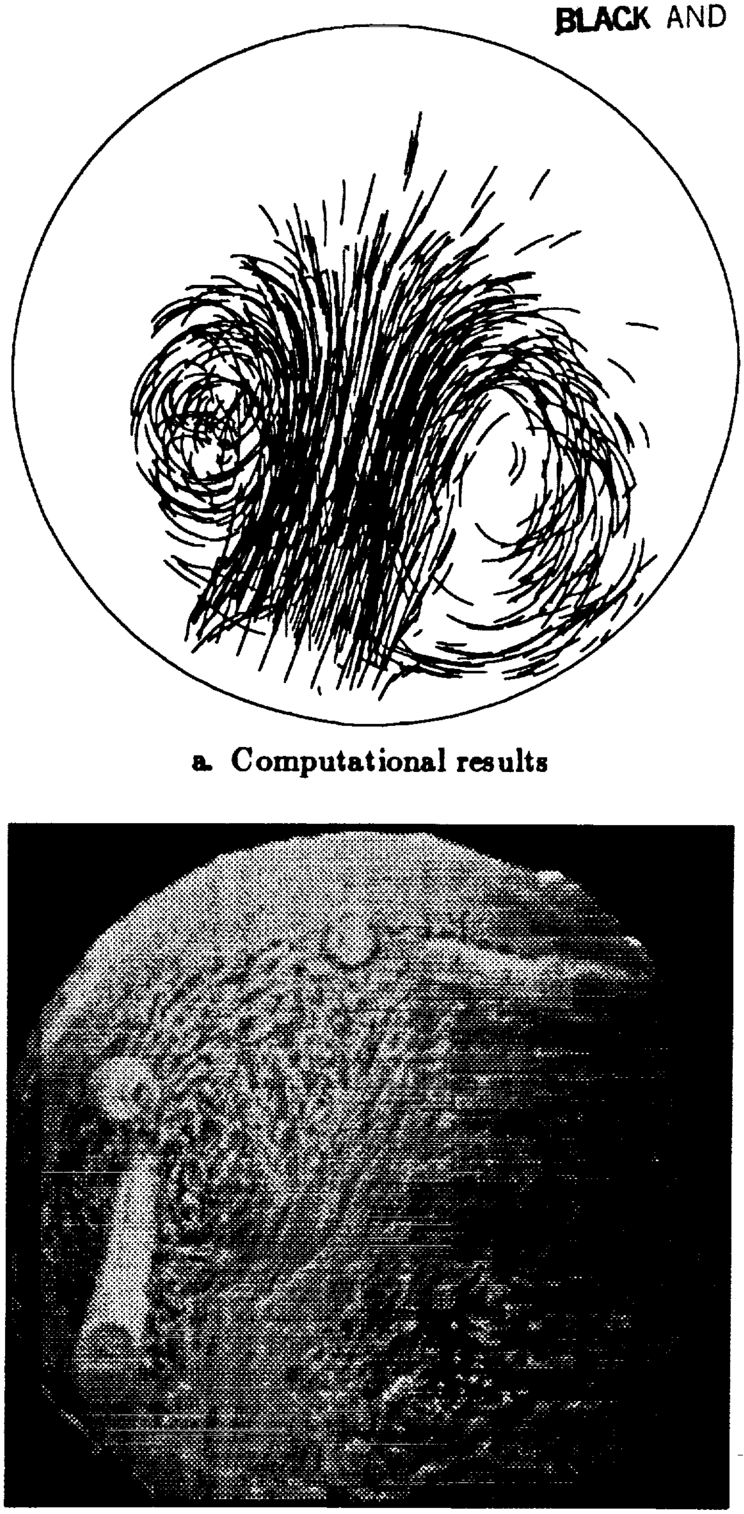
Figure 6.22 Schematic of the electric motor driven Penn State artificial heart.



(a)

(b)

Figure 6.23 A computer model of the Penn State artificial heart: (a) surace grid, (b) main chamber showing valve openings.



a. Computational results

b. Experimental results

Figure 6.24 Incomping particle traces as the piston nears the bottom position: (a) computation (Rogers et al., 1989), (b) experiment (J.M. Tarbell, Penn State Univ, private communication, 1988).

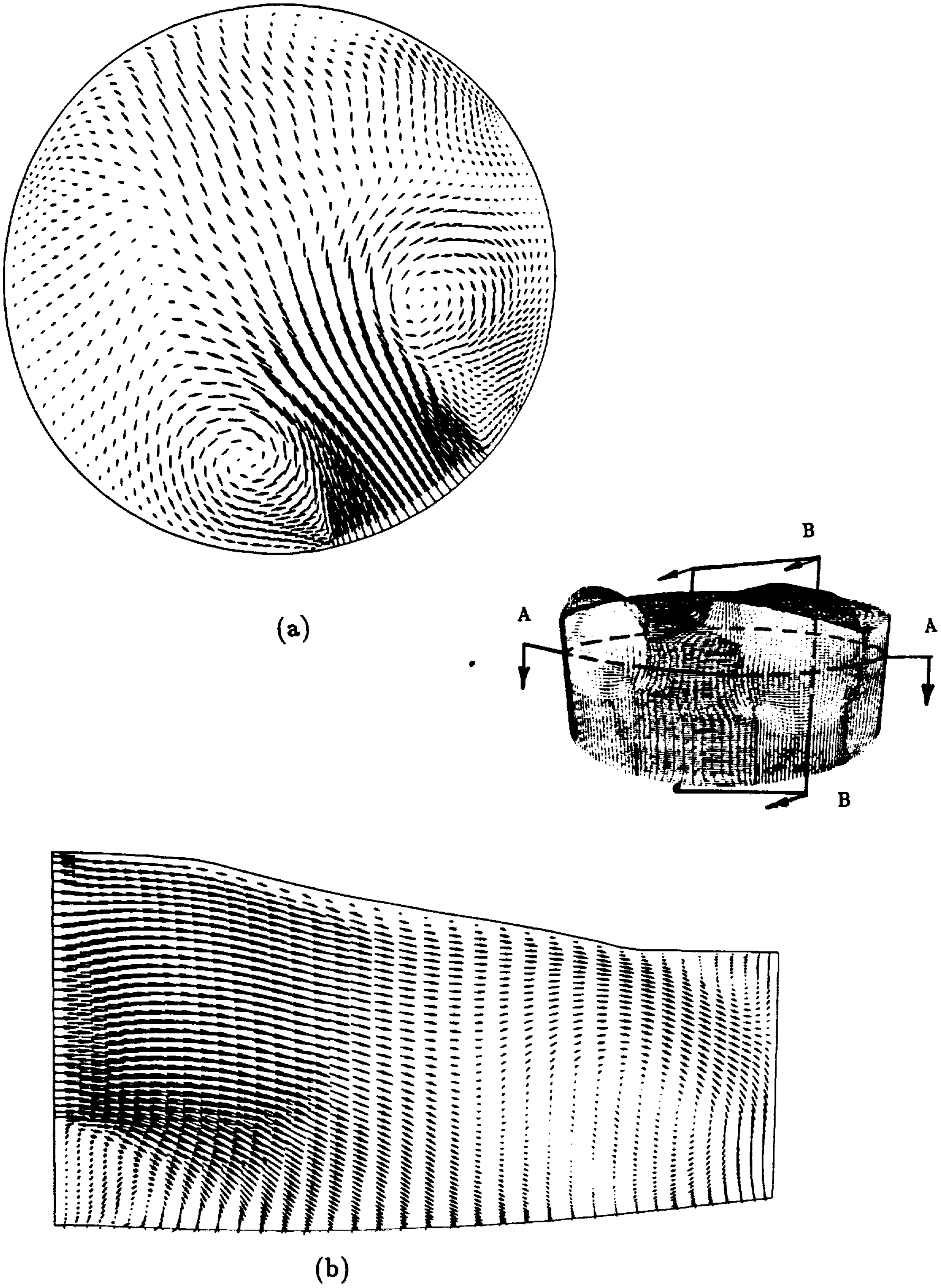


Figure 6.25 Velocity vectors of incoming fluid: (a) top view in horizontal plane through center of inflow valve (cross-section A-A), (b) side view in vertical plane through center of inflow valve (cross-section B-B).

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No. NASA TM-101090	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle
Computation of Viscous Incompressible Flows

5. Report Date
March 1989

6. Performing Organization Code

7. Author(s)
Dochan Kwak

8. Performing Organization Report No.
A-89094

10. Work Unit No.
505-60-01

9. Performing Organization Name and Address
Ames Research Center
Moffett Field, CA 94035

11. Contract or Grant No.

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Washington, DC 20546-0001

13. Type of Report and Period Covered
Technical Memorandum

14. Sponsoring Agency Code

15. Supplementary Notes

Point of Contact: Dochan Kwak, Ames Research Center, MS 258-1, Moffett Field, CA 94035
(415) 694-6743 or FTS 464-6743

Presented at the von Karman Institute for Fluid Dynamics Lecture Series 1989-04, March 6-10, 1989.

16. Abstract

This paper discusses incompressible Navier-Stokes solution methods and their applications to three-dimensional flows. A brief review on existing methods is given followed by a detailed description of recent progress on development of three-dimensional generalized flow solvers. Emphasis is placed on primitive variable formulations which are most promising and flexible for general three-dimensional computations of viscous incompressible flows. Both steady- and unsteady-solution algorithms and their salient features are discussed. Finally, examples of real world applications of these flow solvers are given.

17. Key Words (Suggested by Author(s))
Incompressible flow, Navier-Stokes equations, Computational fluid dynamics, Space shuttle main engine, Artificial heart simulation

18. Distribution Statement
Unclassified-Unlimited

Subject Category - 02

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	118	A06

For sale by the National Technical Information Service, Springfield, Virginia 22161