NASA-TM-102199

NASA-TM-102199 19910006492

N91-15805

NASA Technical Memorandum 102199

Numerical Solution of the Incompressible Navier-Stokes Equations

Stuart E. Rogers, Ames Research Center, Moffett Field, California

November 1990

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035-1000

NUMERICAL SOLUTION OF THE INCOMPRESSIBLE NAVIER-STOKES EQUATIONS

Stuart E. Rogers

SUMMARY

The current work is initiated in an effort to obtain an efficient, accurate, and robust algorithm for the numerical solution of the incompressible Navier-Stokes equations in two- and three-dimensional generalized curvilinear coordinates for both steady-state and time-dependent flow problems. This is accomplished with the use of the method of artificial compressibility and a high-order flux-difference splitting technique for the differencing of the convective terms. Time accuracy is obtained in the numerical solutions by subiterating the equations in pseudo-time for each physical time step. The system of equations is solved with a line-relaxation scheme which allows the use of very large pseudo-time steps leading to fast convergence for steady-state problems as well as for the subiterations of time-dependent problems. Numerous laminar test flow problems are computed and presented with a comparison against analytically known solutions or experimental results. These include the flow in a driven cavity, the flow over a backward-facing step, the steady and unsteady flow over a circular cylinder, flow over an oscillating plate, flow through a one-dimensional inviscid channel with oscillating back pressure, the steady-state flow through a square duct with a 90° bend, and the flow through an artificial heart configuration with moving boundaries. An adequate comparison with the analytical or experimental results is obtained in all cases. Numerical comparisons of the upwind differencing with central differencing plus artificial dissipation indicates that the upwind differencing provides a much more robust algorithm, which requires significantly less computing time. The time-dependent problems require on the order of 10 to 20 subiterations, indicating that the elliptical nature of the problem does require a substantial amount of computing effort.

PREFACE

The use of computational fluid dynamics to analyze incompressible or very nearly incompressible flows has grown considerably in the last several years. The fact that the incompressible Navier-Stokes equations can be used as a governing approximation in a wide range of disciplines has spawned a need for a generalized flow solver. Generalized in the sense that the entire formulation for the basic code be independent of geometrical approximations. Besides being able to reliably produce trustworthy results, a useful analysis code should not require many different numerical parameters which have to be correctly specified before any answer can be obtained. As an analysis tool such a codes value can be measured by its robustness and its ability to be used easily by people without extensive experience in running the code.

The current work is an effort to improve upon an already successful incompressible flow solver known as INS3D. This code uses a very well known and successful approach, namely that of artificial compressibility, coupled with central differencing plus artificial dissipation, and solved with an approximate factorization scheme. This code, being limited to steady-state problems, also requires the specification of artificial dissipation coefficients, a time-step size, and an artificial compressibility parameter. The result of the current work is a code which takes the artificial compressibility approach a step further, by analyzing the physics of this approach, and applying a more appropriate differencing scheme. With the use of a line-relaxation solution, only one numerical parameter need now be specified, the artificial compressibility constant. The following work is intended to show that the resulting code is a robust means of obtaining steady-state and time-dependent solutions to the incompressible Navier-Stokes equations.

This work was partially funded by NASA Marshall Space Flight Center. In addition, some of the funding for the artificial heart work came from the NASA Technology Utilization Office.

TABLE OF CONTENTS

CHAPTER ONE

INTRODUCTION

1.1 Background and Motivation

Numerical solutions to the incompressible Navier-Stokes equations are in greater demand than ever before as the field of computational fluid dynamics (CFD) increases its impact as an engineering tool. Problems which can be addressed by the incompressible Navier-Stokes equations include low-speed flows in aerodynamics, internal flows in propulsion, liquid flows in hydrodynamics, and problems in biomedical fluid analysis. With this wide range of disciplines come a wide range of possible problem geometries. Thus a useful flow solver will require that no simplifying assumptions be made about the geometries, and that the code easily accept different model configurations. This second criteria will require the use of generalized curvilinear (body-fitted) coordinates in the code formulation. The more efficient and robust that a code can become, the more useful a tool it will be for analysis. Therefore, there is a continuing interest in finding solution methodologies which will produce results using the least amount of computing time and with the least amount of effort by the investigator. This is particularly true for unsteady problems. Time-accurate solutions of the incompressible Navier-Stokes equations are particularly time consuming because of the elliptical nature of the governing equations. A disturbance at one point in space affects the entire flow domain instantaneously. This requires that the numerical algorithm propagates information through the entire flow domain during one discrete time step. Thus some type of iterative scheme is usually required to solve the equations in time.

Another important item which directs the development of software is the current trend of new hardware availability. The newest generation of supercomputers has provided an order of magnitude increase in the available processor memory. This can be used to efficiently implement memory intensive algorithms which would otherwise be too costly.

1.2 Available Solution Methods

The primitive variable formulation of the incompressible Navier-Stokes equations is one in which the system of equations is written with the pressure and velocity components as the dependent variables. This form is perhaps the most widely used as a starting point for the development of a numerical solution method. Several methods do exist in which the incompressible Navier-Stokes equations are formulated into a nonprimitive-variable form. These include a velocity-vorticity method as proposed by Fasel [1], and a vector potential-vorticity method proposed by Aziz and Hellums [2]. Each of these methods require the solution of three Poisson equations at each time level when solving the three-dimensional (3-D) equations, which can become very computationally expensive. Further work on the velocity-vorticity method and the use of a direct solver is being implemented by Hafez *et al.* [3]. These methods currently appear to be too expensive for solving large 3D problems.

This leaves methods formulated in primitive variables. Most methods using primitive variables can be classified into three groups. The first of these, and historically one of the most commonly used primitive variable schemes, is the pressure Poisson method as first introduced by Harlow and Welch [4]. In this method, the velocity field is advanced in time using the momentum equations. Then a Poisson equation in pressure, which is formed by taking the divergence of the momentum equations, is solved for the pressure at the current time level such that the continuity equation will be satisfied at the next time level. This method can be very costly because of the need to solve the Poisson equation at every time iteration, and because the velocity and pressure fields are only indirectly coupled.

The second group of primitive-variable methods which has been used extensively is that of the fractional-step method originally introduced by Chorin [5], and used by Kim and Moin [6], and Rosenfeld *et al.* [7]. This method advances the solution in time using two (or more) steps. First the momentum equations are solved for an intermediate velocity field which will generally not be divergence free. In the next step, a pressure field is obtained which will map the intermediate velocity field into a divergence free field, thus obtaining the solution at the next time level. The second step generally requires the solution of a Poisson equation in pressure. This primitive variable method is similar to the first, and although it relates the changes in the pressure field more directly to the divergence of velocity, the pressure and velocity field solutions are still only indirectly coupled.

The third primitive-variable method is that known as the artificial compressibility approach and was first introduced by Chorin [8]. It has been used with much success by Kwak *et al.* [9] for solving complex incompressible flow problems in generalized coordinates. In this formulation, a time derivative of pressure is added to the continuity equation. Together with the momentum equations, this forms a hyperbolic system of equations which can be marched in pseudo-time to a steady-state solution. The method can also be extended to solve time-dependent problems [10] by using subiterations in pseudo-time at every physical time step. If all that is desired is the steady-state solution the artificial compressibility method can be a very efficient formulation because it does not require that a divergence free velocity field be obtained at each iteration. The artificial mechanism by which this corrects the flow field to satisfy the continuity equation is easy to understand: if in a computational cell the net flux of mass becomes greater than zero, thus making the divergence of velocity positive, the pressure in that cell is decreased and through the action of the pressure gradient increases the force drawing the fluid toward the cell; conversely if the divergence of velocity is negative, the pressure increases, which increases the pressure force pushing the fluid away. In this way the pressure and velocity fields are directly coupled. The addition of the time derivative of pressure to the continuity creates a hyperbolic system of equations complete with artificial pressure waves of finite speed. Since this is the case, many of the well-developed compressible flow algorithms can be utilized for this method.

1.3 Objective and Approach

The goal of the current work was to develop a method to solve the incompressible Navier-Stokes equations for both steady-state and time-dependent problems. This must be done as efficiently as possible so it can be used in solving complex 3-D problems in generalized coordinates which may require a large number (100,000 to 500,000) of grid points. In an effort to develop the methodology for such a code, work was done in both two dimensions (2-D) and 3-D. The reason for developing both a 2-D and a 3-D code is that many aspects of the code can be tested and developed more efficiently in 2-D, particularly when studying time-dependent problems. In addition, while most problems of interest are 3-D in nature, studying a 2-D model of the problem can often provide insight into the physics of the problem.

The artificial compressibility approach was chosen for the current work because of the direct coupling between the velocity and pressure fields and because of the hyperbolic nature of the resulting equations. One previous application of this method by Kwak *et al.*[9] employed the use of central differencing of the convective fluxes, with the addition of artificial dissipation to prevent odd-even decoupling of the grid points. This method requires the specification of artificial dissipation coefficients which can strongly influence the resulting solution, and tends to degrade the accuracy of the solution [11]. In an effort to produce a code with as few numerical parameters as possible, upwind differencing was chosen to difference the convective fluxes. An additional advantage of upwind differencing when using an implicit scheme is that it will generally produce a more diagonally dominant system of equations than central differencing. Since the equations to be solved are hyperbolic, some of the upwind differencing schemes which have recently been developed for the compressible Euler and Navier-Stokes equations by a number of authors [12-15] can be utilized. Using the method of Roe [12] the convective terms are differenced 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. Prior to the current investigation, there was one study into using this form of upwind differencing for the artificial compressibility method by Hartwich and Hsu [16,17]. However, they found it necessary to ensure that the scheme was total-variation diminishing (TVD) by reducing the order of the flux differencing and thereby adding more dissipation near sharp gradients. It was reasoned that since the incompressible Navier-Stokes equations will not produce solutions with strong discontinuities, such as shock waves, that flux-difference splitting could be applied without the use of TVD flux-limiters.

In addition to the use of upwind differencing in the current formulation, an implicit solution algorithm was sought which would enhance the overall robustness of the code. The approximate factorization method of Beam and Warming [18] and Briley and MacDonald [19] has received much use since it's introduction. However, the factorization error introduced can severely limit the time step. In addition, when applied to the method of artificial compressibility, it also limits the magnitude of the artificial compressibility constant [9,11]. In the current formulation the set of numerical equations are solved using a nonfactored line relaxation scheme similar to that employed by MacCormack [20]. This scheme can produce a solution close to the exact solution of a direct solver, without using much more time than a factored scheme. Its drawbacks are that it is not nearly as straight forward in its implementation as approximate factorization, and the line relaxation is inherently recursive, limiting the amount of optimization which can be done on vector processors. The current implementation trys to reduce these drawbacks by implementing the scheme in the most efficient way possible. Since this research was being performed on a Cray 2 computer, ample memory (256 million words of processor memory) was available. In all cases extra memory was used when it enabled a savings in computational time.

In the following chapters, the details of the artificial compressibility scheme and its use in solving the incompressible Navier-Stokes equations for steady-state and time-dependent problems are given. The upwind-differencing scheme is detailed, and then the implicit solution procedure is described. Details of the boundary condition procedures are also included. Numerous computed results are presented with comparisons to analytical solutions and experimental results. The test problems include the flow in a driven cavity, the flow over a backward-facing step, the steady and unsteady flow over a circular cylinder, flow over an oscillating plate, flow through a one-dimensional inviscid channel with oscillating back pressure, the steady-state flow through a square duct with a 90° bend, and the flow through an artificial heart configuration with moving boundaries.

CHAPTER TWO

GOVERNING EQUATIONS AND ARTIFICIAL COMPRESSIBILITY

2.1 Introduction

In this chapter the equations governing constant density, incompressible, viscous, Newtonian fluid flow are presented. The generalized coordinate transformation is presented, and then the artificial compressibility method is introduced. The details of its formulation for the solution of both steady-state and time-dependent problems are given. Finally, the numerical evaluations of the coordinate transformation metrics leading to a free-stream preserving system is detailed. The majority of the material presented here will be for the 3-D formulation only, as the 2-D system is an obvious subset of the 3-D system. The primary exceptions to this are the viscous fluxes and the eigensystem of the Jacobian of the convective flux vectors. The details of these two formulations for both 2-D and 3-D are given in Appendix A and Appendix B, respectively.

2.2 Governing Equations

The equations presented here have first been nondimensionalized using the following quantities

$$\begin{gathered}\tilde{u}_i = \frac{u_i}{u_{ref}}, \quad \tilde{x}_i = \frac{x_i}{x_{ref}}, \quad \tilde{t} = \frac{t u_{ref}}{x_{ref}}, \\ \tilde{p} = \frac{p - p_{ref}}{\rho u_{ref}^2}, \quad \tilde{\tau}_{ij} = \frac{\tau_{ij}}{\rho u_{ref}^2}, \quad \tilde{\nu} = \frac{\nu}{x_{ref} u_{ref}} = Re^{-1}\end{gathered} \tag{2.1}$$

where $u_i = u, v$, or w, the three Cartesian velocity components, for $i = 1, 2, 3$, respectively; where $x_i = x, y$, or z, the three Cartesian spatial coordinates, for $i = 1, 2, 3$, respectively; where p represents pressure, ρ represents density, t represents time, ν is the kinematic viscosity, Re is the Reynolds number, τ_{ij} represents the viscous stress tensor, and the subscript *ref* denotes reference quantities. These reference quantities are generally chosen to be the freestream quantities for external flows, or the quantities at the the inflow boundary for an internal flow. The Navier-Stokes equations which govern incompressible, constant density flow are written in conservative form, with the tildes dropped for convenience, as

$$\begin{aligned}
&\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}=0 \\
&\frac{\partial \vec{u}}{\partial t}+\frac{\partial}{\partial x}(e-e_v)+\frac{\partial}{\partial y}(f-f_v)+\frac{\partial}{\partial z}(g-g_v)=0
\end{aligned} \tag{2.2}$$

where

$$\begin{gathered}
\vec{u}=\begin{bmatrix} u \\ v \\ w \end{bmatrix} \\
e=\begin{bmatrix} u^2+p \\ uv \\ uw \end{bmatrix} \quad f=\begin{bmatrix} vu \\ v^2+p \\ vw \end{bmatrix} \quad g=\begin{bmatrix} wu \\ wv \\ w^2+p \end{bmatrix} \\
e_v=\begin{bmatrix} \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \end{bmatrix} \quad f_v=\begin{bmatrix} \tau_{yx} \\ \tau_{yy} \\ \tau_{yz} \end{bmatrix} \quad g_v=\begin{bmatrix} \tau_{zx} \\ \tau_{zy} \\ \tau_{zz} \end{bmatrix}
\end{gathered} \tag{2.3}$$

For modeling turbulent flows, these equations represent Reynolds averaged quantities, and using the Boussinesq approximation for the Reynolds stress, the viscous stress tensor can be written in the form

$$\tau_{ij}=(\nu+\nu_t)\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right) \tag{2.4}$$

where ν_t is the turbulent eddy viscosity. As the primary thrust of this paper is concerned with algorithm development and testing, only laminar flow calculations will be considered and the determination of ν_t will not be discussed. However, the formulation of the current algorithm will allow the use of a spatially and time varying viscosity. This will make it possible to implement an eddy viscosity turbulence model in the future.

2.3 Coordinate Transformation

The equations are transformed into generalized coordinates using

$$\begin{aligned} \xi &= \xi(x,y,z,t) \\ \eta &= \eta(x,y,z,t) \\ \zeta &= \zeta(x,y,z,t) \end{aligned} \tag{2.5}$$

in order to facilitate the use of time-varying, body-fitted coordinates for any specific type of geometry to be used in the numerical calculations. This results in the following

$$\begin{aligned} &\frac{\partial}{\partial \xi}\left(\frac{U}{J}\right) + \frac{\partial}{\partial \eta}\left(\frac{V}{J}\right) + \frac{\partial}{\partial \zeta}\left(\frac{W}{J}\right) = 0 \\ &\frac{\partial \hat{u}}{\partial t} = -\frac{\partial}{\partial \xi}(\hat{e} - \hat{e}_v) - \frac{\partial}{\partial \eta}(\hat{f} - \hat{f}_v) - \frac{\partial}{\partial \zeta}(\hat{g} - \hat{g}_v) = -\hat{r} \end{aligned} \tag{2.6}$$

where $\hat{r}$ represents the right-hand side of the momentum equations, J is the Jacobian of the transformation and

$$\begin{aligned} \hat{u} &= \frac{1}{J}\begin{bmatrix} u \\ v \\ w \end{bmatrix} \\ \hat{e} &= \frac{1}{J}\begin{bmatrix} \xi_x p + uU + \xi_t u \\ \xi_y p + vU + \xi_t v \\ \xi_z p + wU + \xi_t w \end{bmatrix} \\ \hat{f} &= \frac{1}{J}\begin{bmatrix} \eta_x p + uV + \eta_t u \\ \eta_y p + vV + \eta_t v \\ \eta_z p + wV + \eta_t w \end{bmatrix} \\ \hat{g} &= \frac{1}{J}\begin{bmatrix} \zeta_x p + uW + \zeta_t u \\ \zeta_y p + vW + \zeta_t v \\ \zeta_z p + wW + \zeta_t w \end{bmatrix} \\ U &= \xi_x u + \xi_y v + \xi_z w \\ V &= \eta_x u + \eta_y v + \eta_z w \\ W &= \zeta_x u + \zeta_y v + \zeta_z w \end{aligned} \tag{2.7}$$

The quantites U, V, and W are contravariant velocity components and the metrics of the coordinate transformation are represented here using

$$\xi_x = \frac{\partial \xi}{\partial x}, \quad \text{etc}$$

The complexity of the differential form of the viscous fluxes varies considerably based on whether either of the following two assumptions can be made: that the flow computations are taking place on an orthogonal mesh; or that the viscosity is spatially constant (the flow is both laminar and assumed to be Newtonian). The viscous fluxes e_v, f_v, and g_v based on any of the four combinations of the two above assumptions are presented in Appendix A.

It is noted here that when using artificial compressibility to solve unsteady-flow problems on a time-varying mesh, it is important to perform this coordinate transformation before introducing the artificial compressibility relation, otherwise an incorrect right-hand side of the continuity equation will result in the divergence of velocity being nonzero.

The following two sections deal with the formulation which makes it possible to solve Eqs. (2.7) numerically using the method of artificial compressibility for either a steady-state or time-dependent problem.

2.4 Steady-State Formulation

The artificial compressibility relation is introduced by adding a time derivative of pressure to the continuity equation

$$\frac{\partial p}{\partial \tau} = -\beta \nabla \cdot \hat{u} \tag{2.8}$$

In the steady-state formulation the equations are to be marched in a time-like fashion until the right-hand side $\hat{r}$ of Eq. (2.6) and the divergence of velocity approach zero. The time variable for this process no longer represents physical time and so in the momentum equations t is replaced with τ, which can be thought of as a pseudo-time or iteration parameter. Combining Eq. (2.8) with the momentum equations gives the following system of equations

$$\frac{\partial \hat{D}}{\partial \tau} = -\frac{\partial}{\partial \xi}(\hat{E} - \hat{E}_v) - \frac{\partial}{\partial \eta}(\hat{F} - \hat{F}_v) - \frac{\partial}{\partial \zeta}(\hat{G} - \hat{G}_v) = -\hat{R} \tag{2.9}$$

where $\hat{R}$ is defined here as the residual vector of these equations and where

$$\begin{aligned}
\hat{D} &= \frac{1}{J}\begin{bmatrix} p \\ u \\ v \\ w \end{bmatrix} \\
\hat{E} &= \frac{1}{J}\begin{bmatrix} \beta U \\ \xi_x p + uU + \xi_t u \\ \xi_y p + vU + \xi_t v \\ \xi_z p + wU + \xi_t w \end{bmatrix} \\
\hat{F} &= \frac{1}{J}\begin{bmatrix} \beta V \\ \eta_x p + uV + \eta_t u \\ \eta_y p + vV + \eta_t v \\ \eta_z p + wV + \eta_t w \end{bmatrix} \\
\hat{G} &= \frac{1}{J}\begin{bmatrix} \beta W \\ \zeta_x p + uW + \zeta_t u \\ \zeta_y p + vW + \zeta_t v \\ \zeta_z p + wW + \zeta_t w \end{bmatrix} \\
\hat{E}_v &= \begin{bmatrix} 0 \\ \hat{e}_v \end{bmatrix} \\
\hat{F}_v &= \begin{bmatrix} 0 \\ \hat{f}_v \end{bmatrix} \\
\hat{G}_v &= \begin{bmatrix} 0 \\ \hat{g}_v \end{bmatrix}
\end{aligned} \tag{2.10}$$

Since fast convergence is desired and accuracy in pseudo-time is not an issue, the pseudo-time derivative is replaced by an implicit Euler finite-difference formula, giving

$$\frac{\hat{D}^{n+1} - \hat{D}^{n}}{\Delta \tau} = -\hat{R}^{n+1}$$

where superscript n denotes quantities at the nth pseudo-time iteration level. The right-hand side is linearized resulting in

$$\left[\frac{1}{J\Delta\tau}I + \left(\frac{\partial\hat{R}}{\partial D}\right)^n\right](D^{n+1} - D^n) = -R^n \tag{2.11}$$

where I is a 4 x 4 identity matrix and where $D = J\hat{D}$. If this equation were solved exactly as it is, then for very large $\Delta\tau$ this would become a Newton iteration for a steady-state solution. However, it is often not feasible to form the exact Jacobian of the residual vector R. Before these details are discussed, however, an equation similar to Eq. (2.11) for time-dependent problems will be developed.

2.5 Time-Accurate Formulation

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

$$\frac{3\hat{u}^{n+1} - 4\hat{u}^n + \hat{u}^{n-1}}{2\Delta t} = -\hat{r}^{n+1} \tag{2.12}$$

where the superscript n denotes the quantities at time $t = n\Delta t$ and $\hat{r}$ is the right-hand side of the momentum equations given in Eq. (2.6). To solve Eq. (2.12) 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:

$$\frac{p^{n+1,m+1} - p^{n+1,m+1}}{\Delta\tau} = -\beta\nabla \cdot \hat{u}^{n+1,m+1} \tag{2.13}$$

where τ denotes pseudo-time and β is an artificial compressibility parameter. In this form it can be seen that the constants β and $\Delta\tau$ are not independent. However, they are kept separate here primarily because of the following reason. In the numerical equation which approximates this partial differential equation, β is moved inside the divergence operator, and the change in pressure becomes a nonlinear function of β because of the use of upwind differencing. Therefore, in the numerical equation, β and $\Delta\tau$ become independent.

Combining Eq. (2.13) with the momentum equations gives

$$\begin{aligned} & I_{t\tau}(\hat{D}^{n+1,m+1} - \hat{D}^{n+1,m}) \\ & = -\hat{R}^{n+1,m+1} - \frac{I_m}{\Delta t}(1.5\hat{D}^{n+1,m} - 2\hat{D}^{n} + 0.5\hat{D}^{n-1}) \end{aligned} \tag{2.14}$$

where $\hat{D}$ is the same vector defined in Eq. (2.10) and $\hat{R}$ is the same residual vector defined in Eq. (2.9). Also appearing in this equation is $I_{t\tau}$ which is a diagonal matrix and I_m which is a modified identity matrix given by

$$I_{t\tau} = \operatorname{diag}\left[\frac{1}{\Delta\tau},\ \frac{1.5}{\Delta t},\ \frac{1.5}{\Delta t},\ \frac{1.5}{\Delta t}\right]$$

$$I_m = \operatorname{diag}[0,\ 1,\ 1,\ 1]$$

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

$$\begin{aligned} & \left[\frac{I_{t\tau}}{J} + \left(\frac{\partial\hat{R}}{\partial D}\right)^{n+1,m}\right](D^{n+1,m+1} - D^{n+1,m}) \\ & = -\hat{R}^{n+1,m} - \frac{I_m}{\Delta t}(1.5\hat{D}^{n+1,m} - 2\hat{D}^{n} + 0.5\hat{D}^{n-1}) \end{aligned} \tag{2.15}$$

As can be seen, Eq. (2.15) is very similar to the steady-state formulation given by Eq. (2.11). In a sense the time-accurate formulation requires the solution of a steady-state problem in order to advance one physical time step. 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 formation of the convective fluxes is not such a simple matter and is the subject of the next chapter.

2.6 Free-Stream Preservation

The metrics of the coordinate transformation are evaluated from the following formulas (Ref. 21)

$$\begin{aligned}
\hat{\xi}_x &= y_\eta z_\zeta - y_\zeta z_\eta & \hat{\zeta}_x &= y_\xi z_\eta - y_\eta z_\xi \\
\hat{\xi}_y &= z_\eta x_\zeta - z_\zeta x_\eta & \hat{\zeta}_y &= z_\xi x_\eta - z_\eta x_\xi \\
\hat{\xi}_z &= x_\eta y_\zeta - x_\zeta y_\eta & \hat{\zeta}_z &= x_\xi y_\eta - x_\eta y_\xi \\
\hat{\eta}_x &= y_\zeta z_\xi - y_\xi z_\zeta & \hat{\xi}_t &= \hat{\xi}_x x_t + \hat{\xi}_y y_t + \hat{\xi}_z z_t \\
\hat{\eta}_y &= z_\zeta x_\xi - z_\xi x_\zeta & \hat{\eta}_t &= \hat{\eta}_x x_t + \hat{\eta}_y y_t + \hat{\eta}_z z_t \\
\hat{\eta}_z &= x_\zeta y_\xi - x_\xi y_\zeta & \hat{\zeta}_t &= \hat{\zeta}_x x_t + \hat{\zeta}_y y_t + \hat{\zeta}_z z_t
\end{aligned}$$

where $\hat{\xi}_x = \xi_x / J$, etc., and where

$$x_\xi = \frac{\partial x}{\partial \xi}, \quad \text{etc.}$$

Special care is taken in evaluating these metrics to ensure free-stream preservation. When the grid is stationary this can be accomplished by evaluating the metrics using the following difference formula

$$\hat{\xi}_x = \mu_\zeta(\delta_\eta y)\mu_\eta(\delta_\zeta z) - \mu_\eta(\delta_\zeta y)\mu_\zeta(\delta_\eta z)$$

where μ represents an average; i.e.,

$$\mu_\xi \alpha_{i,j,k} = \frac{1}{2}(\alpha_{i+1,j,k} + \alpha_{i-1,j,k})$$

and where δ represents a central difference; i.e.,

$$\delta_\xi \alpha_{i,j,k} = \frac{1}{2}(\alpha_{i+1,j,k} - \alpha_{i-1,j,k})$$

Similar formulas are used for the other metric terms.

When the grid is moving with time, an additional constraint must be placed on the evaluation of the metrics to ensure free-stream preservation. The geometric conservation law of Thomas and Lombard [22] gives a formula for evaluating the Jacobian of the transformation at the $n+1$ time level when using a first-order-implicit Euler scheme. A similar formula corresponding to the second-order, three-point, backward-difference scheme used in the current work is given by

$$J^{n+1} = 1.5\left\{\frac{2}{J^n} - \frac{1}{2J^{n-1}} - \Delta t\left[\delta_\xi(\hat{\xi}_t) + \delta_\eta(\hat{\eta}_t) + \delta_\zeta(\hat{\zeta}_t)\right]\right\}^{-1}$$

Using this formula to update the Jacobian will ensure free-stream preservation on a moving grid. This was tested on a number of grids, and it was found that it worked well. In particular, the moving grid for the artificial heart calculations presented in Chapter 6 was found to produce residuals on the order of 10^{-3} for freestream conditions when no correction for the grid velocities was made. However, when the above correction was made, the residuals were on the order of machine zero.

CHAPTER THREE

UPWIND DIFFERENCING

3.1 Introduction

Upwind differencing is used in the present scheme as a means of following the propagation of the artificial waves introduced by the artificial compressibility. The upwind differencing thus provides a dissipative scheme which automatically suppresses any oscillations caused by the nonlinear convective fluxes. In addition, the upwind-differenced flux vector will contribute to terms on the main diagonal of the Jacobian of the residual, whereas a central-differenced flux vector would not. This will help make the implicit scheme nearly diagonally dominant and contribute greatly to the robustness of the code. Even though the upwind flux differences are more costly to form, the speed up in convergence does result in a significant savings in computing-time requirements.

The upwind scheme is derived from one-dimensional (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 [12] as an approximate Riemann solver for the compressible gas dynamics equations.

3.2 Flux-Difference Splitting

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

$$\frac{\partial \hat{E}}{\partial \xi} \approx \frac{\tilde{E}_{i+1/2} - \tilde{E}_{i-1/2}}{\Delta \xi} \tag{3.1}$$

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

The numerical flux is given by

$$\tilde{E}_{i+1/2} = \frac{1}{2}\left[\hat{E}(D_{i+1}) + \hat{E}(D_i) - \phi_{i+1/2}\right] \tag{3.2}$$

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

$$\phi_{i+1/2} = \Delta E^{+}_{i+1/2} - \Delta E^{-}_{i+1/2} \tag{3.3}$$

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

$$\Delta E^{\pm}_{i+1/2} = A^{\pm}(\bar{D})\Delta D_{i+1/2} \tag{3.4}$$

where the Δ operator is given by

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

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

$$A^{\pm} = X_1 \Lambda_1^{\pm} X_1^{-1} \tag{3.5}$$

where $\Lambda_1^{\pm}$ is composed of all the positive (negative) eigenvalues, where each individual positive (negative) eigenvalue is computed as proposed by Buning and Steger [23]

$$\lambda_i^{\pm} = \frac{1}{2}\left(\lambda_i \pm \sqrt{\lambda_i^2 + \epsilon}\right)$$

where ϵ is a small parameter used to avoid a discontinuous transition in the delta fluxes as any of the eigenvalues changes sign. A value of $\epsilon = 0.01$ is used for all of the calculations in this work. In Eq. (3.5) 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. These matrices for both the 2-D and 3-D systems of equations are presented in Appendix B. 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 [12] which are necessary for a conservative scheme, are satisfied if the following averaging procedure is employed

$$\bar{D} = \frac{1}{2}(D_{i+1} + D_i) \tag{3.6}$$

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 difference can be obtained using

$$\phi_{i+1/2} = -\frac{1}{3}\left(\Delta E^{+}_{i-1/2} - \Delta E^{+}_{i+1/2} + \Delta E^{-}_{i+1/2} - \Delta E^{-}_{i+3/2}\right) \tag{3.7}$$

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 [24] 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

$$\phi_{i+1/2} = -\frac{1}{30}[-2\Delta E^{+}_{i-3/2} + 11\Delta E^{+}_{i-1/2} - 6\Delta E^{+}_{i+1/2} - 3\Delta E^{+}_{i+3/2} \\ +2\Delta E^{-}_{i+5/2} - 11\Delta E^{-}_{i+3/2} + 6\Delta E^{-}_{i+1/2} + 3\Delta E^{-}_{i-1/2}] \tag{3.8}$$

Next to the boundary, a scheme that is nearly second-order accurate can be maintained by the third- and fifth-order schemes by using the following

$$\phi_{i+1/2} = \epsilon\left(\Delta E^{+}_{i+1/2} - \Delta E^{-}_{i+1/2}\right) \tag{3.9}$$

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 Eq. (3.3). 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 used for all of the results presented in this paper.

The right and left matrices given in Appendix B clearly show that the artificial compressibility parameter β will affect not only the continuity equation, but the momentum equations as well. An analysis for the Cartesian coordinate case shows that the dissipation terms added to the momentum equations will grow as the square root of β. This indicates that the value of β should be chosen with care when using the upwind-differencing, as extremely large values could cause large errors in the differencing of the momentum equations.

3.3 Central Differencing and Artificial Dissipation

For some of the test cases to be presented in the Computed Results chapter a comparison between the present upwind-difference scheme and a central-differencing plus artificial-dissipation formulation is made. This section presents the artificial terms used in these comparisons.

As stated above, Eq. (3.2) will result in a central-difference scheme if $\phi_{i+1/2} = 0$. If this formula is nondissipative, and if it is used without any additional numerical dissipation, an odd-even grid-point decoupling will develop, and the solution will show wiggles and oscillations from grid point to grid point. This is caused by the nonlinear convective fluxes introducing high frequencies which the natural viscous dissipation will not always damp out. A common method by which artificial dissipation can be added to the differencing is to use a difference approximation to a fourth derivative of the dependent variables on the right-hand side of the equations, and a difference approximation to a second derivative on the left-hand side. For more details, see Pulliam [25]. The right-hand side fourth-order dissipation can be given by

$$\phi_{i+1/2} = -\epsilon_e |A_1|_{i+1/2}(D_{i+2} - 3D_{i+1} + 3D_i - D_{i-1}) \tag{3.10}$$

where ϵ_e is the explicit artificial dissipation coefficient, and $|A_1|_{i+1/2}$ is the spectral radius of the Jacobian matrix of the convective flux vector. The second-order dissipation term used in forming the implicit side of the equations is given by

$$\phi_{i+1/2} = \epsilon_i |A_1|_{i+1/2}(D_{i+1} - D_i) \tag{3.11}$$

where ϵ_i is the implicit artificial dissipation coefficient.

CHAPTER FOUR

IMPLICIT SCHEME

This chapter describes the way in which Eqs. (2.11) and (2.15) are numerically represented and solved for the 3-D equations. The 2-D system is solved in a similar fashion. 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 upwind-difference formula given in Eq. (3.1) 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

$$\begin{aligned}\hat{R}_{ijk} =& \frac{\tilde{E}_{i+1/2,j,k} - \tilde{E}_{i-1/2,j,k}}{\Delta\xi} + \frac{\tilde{F}_{i,j+1/2,k} - \tilde{F}_{i,j-1/2,k}}{\Delta\eta} \\ &+ \frac{\tilde{G}_{i,j,k+1/2} - \tilde{G}_{i,j,k-1/2}}{\Delta\zeta} - \frac{(\hat{E}_v)_{i+1/2,j,k} - (\hat{E}_v)_{i-1/2,j,k}}{\Delta\xi} \\ &- \frac{(\hat{F}_v)_{i,j+1/2,k} - (\hat{F}_v)_{i,j-1/2,k}}{\Delta\eta} - \frac{(\hat{G}_v)_{i,j,k+1/2} - (\hat{G}_v)_{i,j,k-1/2}}{\Delta\eta}\end{aligned} \tag{4.1}$$

The generalized coordinates are chosen so that $\Delta\xi$, $\Delta\eta$, and $\Delta\zeta$ are equal to one. To limit the band-width 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

$$\begin{aligned}\hat{R}_{ijk} = \frac{1}{2}(&\hat{E}_{i+1,j,k} - \hat{E}_{i-1,j,k} + \hat{F}_{i,j+1,k} - \hat{F}_{i,j-1,k} + \hat{G}_{i,j,k+1} - \hat{G}_{i,j,k-1} \\ &-\Delta E^{+}_{i+1/2,j,k} + \Delta E^{-}_{i+1/2,j,k} + \Delta E^{+}_{i-1/2,j,k} - \Delta E^{-}_{i-1/2,j,k} \\ &-\Delta F^{+}_{i,j+1/2,k} + \Delta F^{-}_{i,j+1/2,k} + \Delta F^{+}_{i,j-1/2,k} - \Delta F^{-}_{i,j-1/2,k} \\ &-\Delta G^{+}_{i,j,k+1/2} + \Delta G^{-}_{i,j,k+1/2} + \Delta G^{+}_{i,j,k-1/2} - \Delta G^{-}_{i,j,k-1/2}) \\ &-(\hat{E}_v)_{i+1/2,j,k} + (\hat{E}_v)_{i-1/2,j,k} - (\hat{F}_v)_{i,j+1/2,k} \\ &+(\hat{F}_v)_{i,j-1/2,k} - (\hat{G}_v)_{i,j,k+1/2} + (\hat{G}_v)_{i,j,k-1/2}\end{aligned} \tag{4.2}$$

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

$$\begin{aligned}\frac{\partial \hat{R}}{\partial D} = \mathcal{B}\Bigg[&\frac{\partial \hat{R}_{ijk}}{\partial D_{i,j,k-1}}, 0, ..., 0, \frac{\partial \hat{R}_{ijk}}{\partial D_{i,j-1,k}}, 0, ..., 0, \\ &\frac{\partial \hat{R}_{ijk}}{\partial D_{i-1,j,k}}, \frac{\partial \hat{R}_{ijk}}{\partial D_{i,j,k}}, \frac{\partial \hat{R}_{ijk}}{\partial D_{i+1,j,k}}, \\ &0, ..., 0, \frac{\partial \hat{R}_{ijk}}{\partial D_{i,j+1,k}}, 0, ..., 0, \frac{\partial \hat{R}_{ijk}}{\partial D_{i,j,k+1}}\Bigg]\end{aligned} \tag{4.3}$$

where each entry of the banded matrix represents a vector of 4 x 4 blocks in 3-D (3 x 3 blocks in 2-D) which is aligned along a diagonal of the matrix. These exact Jacobians, however, can be very costly to form. Therefore, approximate Jacobians of the flux differences as derived and analyzed by both Barth [26] and Yee[27] are used. These are given as follows

$$
\begin{aligned}
\frac{\partial \hat{R}_{ijk}}{\partial D_{i,j,k-1}} &\approx \frac{1}{2}(-\hat{C}_{i,j,k-1} - C^{+}_{i,j,k-1/2} + C^{-}_{i,j,k-1/2}) - (\bar{\gamma}_3)_{i,j,k-1/2} \\
\frac{\partial \hat{R}_{ijk}}{\partial D_{i,j-1,k}} &\approx \frac{1}{2}(-\hat{B}_{i,j-1,k} - B^{+}_{i,j-1/2,k} + B^{-}_{i,j-1/2,k}) - (\bar{\gamma}_2)_{i,j-1/2,k} \\
\frac{\partial \hat{R}_{ijk}}{\partial D_{i-1,j,k}} &\approx \frac{1}{2}(-\hat{A}_{i-1,j,k} - A^{+}_{i-1/2,j,k} + A^{-}_{i-1/2,j,k}) - (\bar{\gamma}_1)_{i-1/2,j,k} \\
\frac{\partial \hat{R}_{ijk}}{\partial D_{i,j,k}} &\approx \frac{1}{2}(A^{+}_{i+1/2,j,k} + A^{+}_{i-1/2,j,k} - A^{-}_{i+1/2,j,k} - A^{-}_{i-1/2,j,k} \\
&\quad + B^{+}_{i,j+1/2,k} + B^{+}_{i,j-1/2,k} - B^{-}_{i,j+1/2,k} - B^{-}_{i,j-1/2,k} \\
&\quad + C^{+}_{i,j,k+1/2} + C^{+}_{i,j,k-1/2} - C^{-}_{i,j,k+1/2} - C^{-}_{i,j,k-1/2}) \\
&\quad + (\bar{\gamma}_1)_{i+1/2,j,k} + (\bar{\gamma}_2)_{i,j+1/2,k} + (\bar{\gamma}_3)_{i,j,k+1/2} \\
&\quad + (\bar{\gamma}_1)_{i-1/2,j,k} + (\bar{\gamma}_2)_{i,j-1/2,k} + (\bar{\gamma}_3)_{i,j,k-1/2} \\
\frac{\partial \hat{R}_{ijk}}{\partial D_{i+1,j,k}} &\approx \frac{1}{2}(\hat{A}_{i+1,j,k} - A^{+}_{i+1/2,j,k} + A^{-}_{i+1/2,j,k}) - (\bar{\gamma}_1)_{i+1/2,j,k} \\
\frac{\partial \hat{R}_{ijk}}{\partial D_{i,j+1,k}} &\approx \frac{1}{2}(\hat{B}_{i,j+1,k} - B^{+}_{i,j+1/2,k} + B^{-}_{i,j+1/2,k}) - (\bar{\gamma}_2)_{i,j+1/2,k} \\
\frac{\partial \hat{R}_{ijk}}{\partial D_{i,j,k+1}} &\approx \frac{1}{2}(\hat{C}_{i,j,k+1} - C^{+}_{i,j,k+1/2} + C^{-}_{i,j,k+1/2}) - (\bar{\gamma}_3)_{i,j,k+1/2}
\end{aligned}
\tag{4.4}
$$

where $\hat{A} = \hat{A}_1, \hat{B} = \hat{A}_2$, and $\hat{C} = \hat{A}_3$ as given in Appendix B, and only the orthogonal mesh terms are retained for the implicit viscous terms giving

$$
\begin{aligned}
\gamma_1 &= \frac{1}{ReJ}(\xi_x^2 + \xi_y^2 + \xi_z^2) Im \frac{\partial}{\partial \xi} \\
\gamma_2 &= \frac{1}{ReJ}(\eta_x^2 + \eta_y^2 + \eta_z^2) Im \frac{\partial}{\partial \eta} \\
\gamma_3 &= \frac{1}{ReJ}(\zeta_x^2 + \zeta_y^2 + \zeta_z^2) Im \frac{\partial}{\partial \zeta} \\
(\bar{\gamma}_1)_{i+1/2,j,k} &= \frac{1}{2}\left[(\gamma_1)_{i+1,j,k} + (\gamma_1)_{i-1,j,k}\right] \\
(\bar{\gamma}_2)_{i,j+1/2,k} &= \frac{1}{2}\left[(\gamma_2)_{i,j+1,k} + (\gamma_2)_{i,j-1,k}\right] \\
(\bar{\gamma}_3)_{i,j,k+1/2} &= \frac{1}{2}\left[(\gamma_3)_{i,j,k+1} + (\gamma_3)_{i,j,k-1}\right]
\end{aligned}
\tag{4.5}
$$

The term *Im* is a modified identity matrix with a zero in the first diagonal entry.

The matrix equation is solved using a Gauss-Seidel line-relaxation method similar to that used by MacCormack [20] and by Chakravarthy[28]. To implement this scheme, the entire numerical matrix equation is first formed from values at the *nth* time level. This takes a significant amount of memory to store all of the left-hand side matrices, but the savings in computing time justifies this. At this point the numerical equation is stored as the following banded matrix

$$\mathcal{B}[T,0,...,0,U,0,...,0,X,Y,Z,0,...,0,V,0,...,0,W]\Delta D=\hat{R} \tag{4.6}$$

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 for a forward sweep

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

where l is an iteration index. The number of iterations is generally a small number like two or three. This system can be solved most efficiently by first performing and storing the lower-upper (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.

CHAPTER FIVE

BOUNDARY CONDITIONS

5.1 Method of Characteristics

Implicit boundary conditions are used at all of the boundaries, this helps make possible 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 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[29], 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

$$X^{-1}\frac{\partial \hat{D}}{\partial \tau} = -\Lambda X^{-1}\frac{\partial D}{\partial \xi} \tag{5.1}$$

If one were to move the X^{-1} matrix inside the spatial and time derivatives, the result would be 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 characteristic waves. For a positive (negative) eigenvalue, there corresponds a wave propagating information in the positive (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, we will use these characteristics which bring information from the interior as part of our boundary conditions. The rest of the information should come from outside the computational domain, and so one is free to specify some variables.

There will always be at least one characteristic wave traveling toward the boundary from the interior because there is always at least one positive eigenvalue and one negative eigenvalue in the artificial compressibility equations. To select the proper characteristic waves, Eq. (5.1) is multiplied by a diagonal selection matrix L which has an entry of one in the position of the eigenvalue we wish to select, and zeros elsewhere. Thus

$$LX^{-1}\frac{\partial \hat{D}}{\partial \tau} = -L\Lambda X^{-1}\frac{\partial D}{\partial \xi} \tag{5.2}$$

Replacing the time derivative with an implicit Euler time step gives

$$\left(\frac{LX^{-1}}{J\Delta\tau} + L\Lambda X^{-1}\frac{\partial}{\partial \xi}\right)(D^{n+1} - D^{n}) = -L\Lambda X^{-1}\frac{\partial D^{n}}{\partial \xi} \tag{5.3}$$

The following discussion relates to the number of positive and negative eigenvalues at each specific type of boundary for the 3-D system of equations. Similar results apply to the 2-D system, the only difference being that where there are three eigenvalues of one sign in the 3-D system, there are only two eigenvalues in the 2-D system.

Equation (5.3) gives either one or three relations, depending on the number of nonzero elements in L. To complete the set of equations, some other information must be supplied. Here is defined a vector Ω that contains a boundary condition for each zero element on the diagonal of the matrix L. The other elements of Ω are set to zero. Since these boundary conditions are held constant in time, the following holds true

$$\frac{\partial \Omega}{\partial \tau} = 0 \longrightarrow \frac{\partial \Omega}{\partial D}\frac{\partial D}{\partial \tau} = 0 \longrightarrow \frac{\partial \Omega}{\partial D} = 0 \tag{5.4}$$

Combining Eqs. (5.3) and (5.4) gives

$$\left[\frac{LX^{-1}}{J\Delta\tau} + L\Lambda X^{-1}\frac{\partial}{\partial \xi} + \frac{\partial \Omega}{\partial D}\right](D^{n+1} - D^{n}) = -L\Lambda X^{-1}\frac{\partial D^{n}}{\partial \xi} \tag{5.5}$$

Equation (5.5) can be used to update the variables implicitly at any of the inflow or outflow boundaries with the proper choice of L and Ω.

5.2 Inflow Boundary

At the inflow, there will be one characteristic wave traveling out of the computational domain and three traveling in since fluid is traveling into the domain. If the incoming fluid is traveling in the positive ξ direction, then the signs of the eigenvalues are given by

$$\lambda_1, \lambda_2 > 0$$

$$\lambda_3 > 0$$

$$\lambda_4 < 0$$

This fourth eigenvalue corresponds to the only wave carrying information out of the computational domain. So L will have a one for the fourth diagonal entry. If the incoming fluid is traveling in the negative ξ direction, then

$$\lambda_1, \lambda_2 < 0$$

$$\lambda_3 > 0$$

$$\lambda_4 < 0$$

For the same reasoning a one in the third diagonal entry of L is required for this boundary.

Two different sets of specified variables have been used successfully for inflow boundaries. 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 this set and for the first boundary mentioned previously, the Ω vector is

$$\Omega = \begin{bmatrix} p + \frac{1}{2}(u^2 + v^2 + w^2) \\ v \\ w \\ 0 \end{bmatrix} ; \quad \frac{\partial \Omega}{\partial D} = \begin{bmatrix} 1 & u & v & w \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

The second possible set of specified variables consists of the velocity components. This is useful for problems in which a specific velocity profile is desired at the inflow. The Ω vector for this is

$$\Omega = \begin{bmatrix} u \\ v \\ w \\ 0 \end{bmatrix} ; \quad \frac{\partial \Omega}{\partial D} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

5.3 Outflow Boundary

At the outflow boundary there are three characteristic waves traveling out of the computational domain and one traveling in since fluid is also leaving the domain. If the fluid is traveling along the positive ξ direction then

$$\lambda_1, \lambda_2 > 0$$

$$\lambda_3 > 0$$

$$\lambda_4 < 0$$

and ones are required in the first three diagonal entries of the L matrix. If the fluid is traveling in the negative ξ direction then

$$\lambda_1, \lambda_2 < 0$$

$$\lambda_3 > 0$$

$$\lambda_4 < 0$$

and ones are required in the first, second, and fourth diagonal entries of the L matrix.

For all of the test problems presented in this paper static pressure was specified at the outflow boundary. For the first outflow boundary mentioned previously this results in

$$\Omega = \begin{bmatrix} 0 \\ 0 \\ 0 \\ p \end{bmatrix}; \quad \frac{\partial \Omega}{\partial D} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

CHAPTER SIX

COMPUTED RESULTS

6.1 Introduction

The 2-D and 3-D codes have been run for a large number of laminar flow test cases. Eight of these cases are presented here as examples of the codes capability to efficiently and accurately predict flow physics. These cases are as follows: a driven square cavity flow, which has become a standard validation test case due to its simple boundary conditions; the flow over a backward facing step, which has also become a popular validation case as it is an example of an internal flow with a recirculating region; the steady flow over a circular cylinder, which has become an extensively validated external flow problem; the flow on top of an oscillating plate, known as Stoke's second problem; the flow through an inviscid one dimensional (1-D) channel with an oscillating back pressure; vortex shedding behind a 2-D circular cylinder; the flow through a 3-D square duct with a 90° bend; and the flow through an artificial heart.

The computing times reported here are the CPU seconds used on a Cray 2. For comparison, these times are nearly the same as obtained running on a Cray XMP-48. For steady-state problems, the computations are run until the maximum residual has converged between 4 to 6 orders of magnitude, the maximum divergence of velocity over all the points is less than 10^{-4}, and the flow quantities being measured have approached a steady-state value in at least 4 significant digits. For unsteady-flow problems the sub-iterations were continued until the maximum residual and the maximum divergence of velocity was less than 10^{-4}, except for the artificial heart case, in which this number was increased to 10^{-3} due to computing time limitations. This convergence is usually obtained in 100 to 200 iterations for steady-state cases, and in 10 to 20 subiterations for unsteady problems.

For each of the test cases presented, the larger the time step $\Delta\tau$, the better the convergence was. In all of the upwind-difference cases presented here the solution remained stable no matter how large a time step was used, so the time step was set to 10^{12} which effectively reduced the $1/\Delta\tau$ term to zero. The choice of β for each case was determined through numerical convergence tests. It was found that the convergence was quite sensitive to the value of β, and in some cases, the choice of β could cause the scheme to become unstable. For most cases, however, it was easy to find a range of β for which the code would converge very quickly. This process usually required running the code for a small number of iterations using values of β equal to 0.1, 1, 10, 100, etc., and isolating the range of β which gave the best convergence rates. The convergence of the current formulation is degraded by the errors introduced by the approximate Jacobians on the left-hand side of the equations and by the fact that the whole system of equations is not exactly solved by the line-relaxation process. If it were possible to use the exact Jacobians and solve the system exactly, then this would be a Newton iteration, in which case one would expect to have quadratic convergence when using a very large time step for any value of β. In all cases, the time quantities given in the following flow problems refer to dimensionless time.

6.2 Driven Cavity Flow

The two-dimensional flow in a driven square cavity whose top wall moves with a uniform velocity has been used rather extensively as a validation test case by several authors in the recent past. It provides a good test case in that there is no primary flow direction and the boundary conditions are very simple to employ. Ghia *et al.* [30] presented extensive numerical data obtained from their multigrid vorticity-stream function formulation using very fine grids. Although there was no experimental results for this problem, they reported results which agreed quite well with other computational efforts, particularly at the lower range of Reynolds numbers. Other recent computational work involving this particular geometry include Schreiber and Keller [31] who use a vorticity-stream function formulation; Kim and Moin [6] who use a fractional-step method in primitive variables in conjunction with approximate factorization; Vanka [32] who uses a multigrid technique in primitive variables; and Benjamin and Denny [33] who use a centrally-differenced vorticity-stream function formulation in conjunction with an ADI scheme.

The current calculations attempt to maintain the accuracy of these authors while using fewer grid points. First, the results of a grid resolution study are reported here. This study was done for a Reynolds number of 10,000, which was the highest Reynolds number tested and had by far the most complex flow pattern. The grids for this study were obtained by first computing a grid with 161 by 161 points, which were clustered near the walls. Subsequently coarser grids of dimensions 81 by 81, 41 by 41, and 21 by 21 were obtained by removing every other grid point in both directions from the previous grid. An example of one of these grids, the 81 by 81 grid, is shown in Fig. 1. These calculations used 11 implicit line relaxation sweeps in the ξ-direction for each iteration. The parameters used in these calculations, as well as several flow quantities are shown in Table 1. The value of β, the number of iterations required for 6 orders of magnitude convergence (nt_c), and the computing time, comprise the first entries in this table. This shows how the value of the near optimum β for this problem increases with the grid-point density. The convergence does slow somewhat with increasing grid-point density, but it is only by a small amount for the denser grids. The other entries in Table 1 are several of the flow quantities which were computed, namely the maximum and minimum vertical velocity component along the centerline of the cavity, and the streamfunction, ψ_{min}, and vorticity values, $\omega_{v.c.}$, at the center of the primary vortex.

Table 1 Numerical Parameters and Flow Quantities for
Grid-Resolution Study of Driven-Cavity Flow

Grid	β	nt_c	Time	v_{max}	v_{min}	ψ_{min}	$\omega_{v.c.}$
161	1	440	680	0.458	-0.573	-0.1204	-1.880
81	1	390	154	0.446	-0.562	-0.1174	-1.817
41	0.1	380	42	0.370	-0.473	-0.1015	-1.621
21	0.01	170	6	0.059	-0.070	-0.0222	-0.790

Table 2 shows the relative error in each of these flow quantities for the coarser grids as compared to the 161 by 161 grid results. Also included in this table are the ratios of the error from one grid to the next. These values show the order of the convergence toward a grid-independent solution. Since these values are all greater than

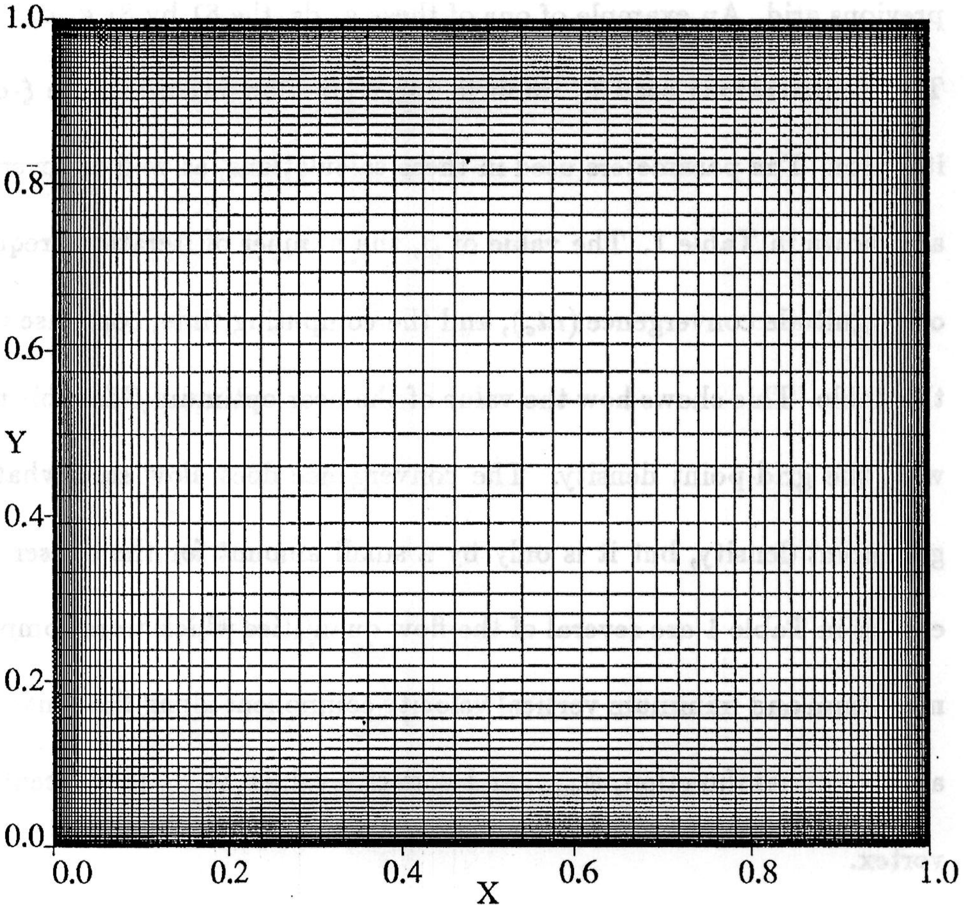


Fig. 1. Grid with 81x81 points used for computing the driven cavity flow.

4, this indicates that accuracy of the overall spatial differencing is between second and third order. Thus the effect of the fifth-order upwind differencing is to help increase the overall order above the second-order accuracy of the viscous terms and the near-boundary differencing. These numbers indicate that the difference between the 161 by 161 grid and a finer grid would be very slight, and that the solution given by this grid is essentially grid independent. This also shows that the results from the 81 by 81 grid are only in error by 2 or 3 percent, and that the results from this grid are close enough to a grid-independent solution that this grid could be used for the rest of this study.

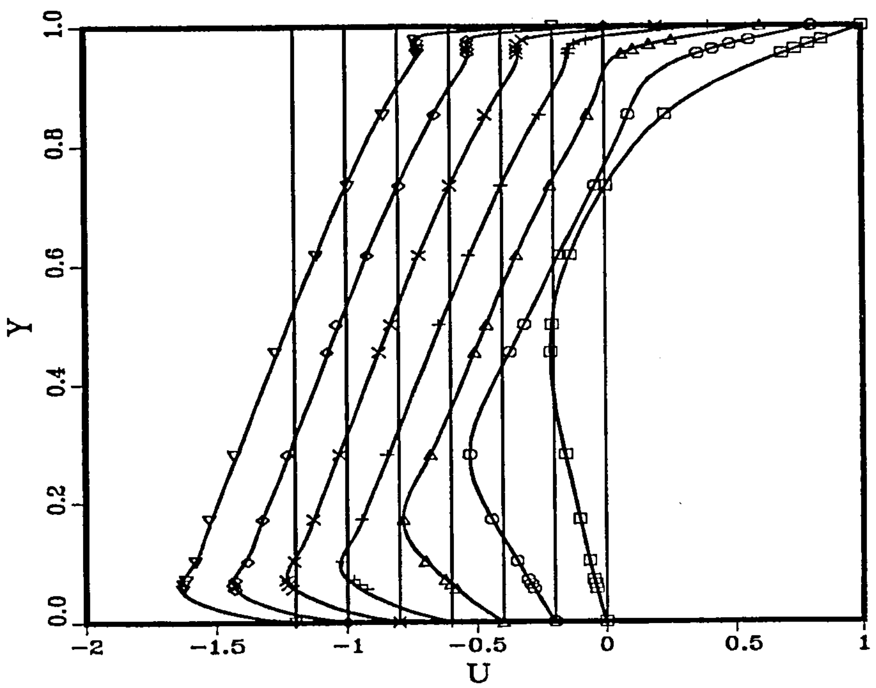
The remaining flow calculations for this problem were computed using the 81 by

Table 2 Relative Errors in Flow Quantities for Grid-Resolution Study of Driven-Cavity Flow

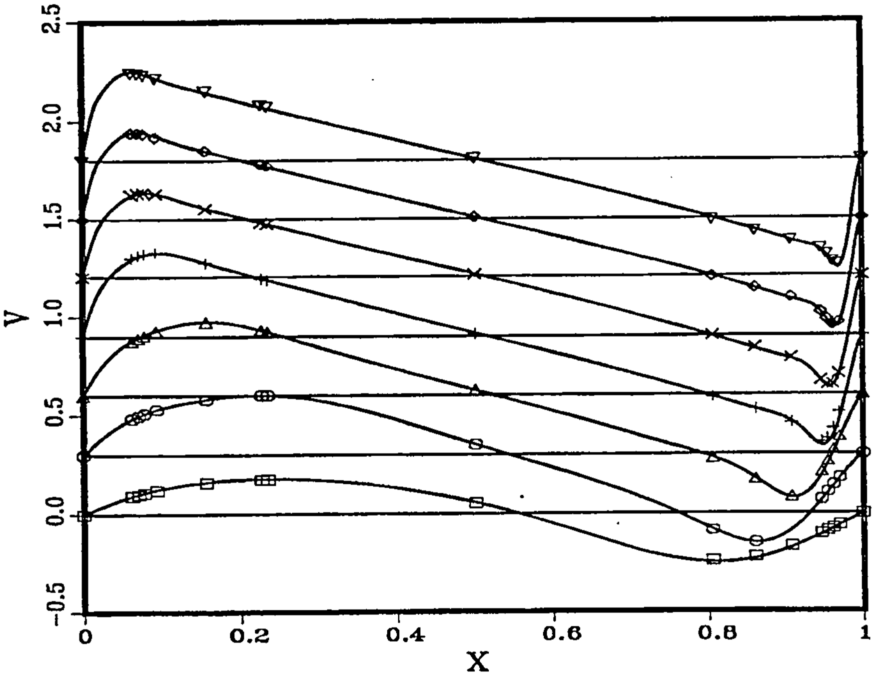
Grid	v_{max}	v_{min}	ψ_{min}	$\omega_{v.c.}$
81	0.024	0.019	0.025	0.034
41	0.191	0.173	0.157	0.138
21	0.872	0.878	0.816	0.580
er_{41}/er_{81}	7.96	9.10	6.30	4.06
er_{21}/er_{41}	4.57	5.08	5.20	4.20

81 grid for Reynolds numbers of 100, 400, 1000, 3200, 5000, 7500, and 10,000. The value of the artificial compressibility β was set to 20 for the Reynolds number of 100, to 10 for the 400 Reynolds number case, to 2 for a Reynolds number of 1000, and was set to 1 for the higher Reynolds numbers. This problem indicates that there is an inverse relationship between the Reynolds number and the optimum β.

The velocity components on the lines passing through the geometric center of the cavity are compared to the results of Ghia *et al.* [30] in Fig. 2. The u-velocity component is plotted along the y-axis for the different Reynolds numbers in Fig. 2a. The origins of the plots has been shifted to the left for each successive Reynolds number case. The data of Ghia was obtained from a uniform grid of 129 by 129 points for Reynolds numbers up to 3200, and a uniform grid of 257 by 257 points for Reynolds numbers 5000 and greater. It is seen that these two computed results agree very well with each other. In Fig. 2b, the v-velocity component is plotted along the x-axis passing through the geometric center for the different Reynolds numbers. The origins of these plots are shifted up for each successive Reynolds number case. Again, good agreement is seen between the two computed results.



2a. U-velocity component versus y on the vertical centerline.



2b. V-velocity component versus x on the horizontal centerline.

Fig. 2. Comparison between present results (solid line) and computations of Ghia *et al.*[30] (symbols). □ : Re=100, ○: Re=400, △: Re=1000, +: Re=3200, ×: Re=5000, ◇: Re=7500, and ▽: Re=10,000.

In Table 3, the stream function and vorticity quantities are given for the core

Table 3 Stream-Function and Vorticity at the Center of the Primary Vortex for Different Reynolds Numbers

Re	Present ψ_{min} ($\omega_{v.c.}$)	Ghia *et al.* [24] ψ_{min} ($\omega_{v.c.}$)	Schreiber *et al.* [25] ψ_{min} ($\omega_{v.c.}$)	Kim *et al.* [6] ψ_{min} ($\omega_{v.c.}$)
100	-0.1030(-3.104) 81x81	-0.1034(-3.166) 129x129	-0.1033(-3.182) 121x121	-0.1030(-3.177) 65x65
400	-0.1131(-2.296) 81x81	-0.1139(-2.294) 129x129	-0.1130(-2.281) 141x141	-0.1120(-2.260) 65x65
1000	-0.1171(-2.044) 81x81	-0.1179(-2.050) 129x129	-0.1160(-2.026) 141x141	-0.1160(-2.026) 97x97
3200	-0.1195(-1.904) 81x81	-0.1204(-1.989) 129x129	–	-0.1150(-1.901) 97x97
5000	-0.1192(-1.846) 81x81	-0.1190(-1.860) 257x257	–	-0.1120(-1.812) 97x97
7500	-0.1186(-1.846) 81x81	-0.1200(-1.880) 257x257	–	–
10000	-0.1177(-1.826) 81x81	-0.1197(-1.881) 257x257	-0.1028(-1.622) 180x180	–

of the primary vortex for all Reynolds numbers tested. Included with the present results are the results of Ghia *et al.* [30], Schreiber and Keller [31], and Kim and Moin [6]. Listed below the flow quantities is the grid size used for the calculation. Good agreement among all calculations is seen in the lower Reynolds number cases. The discrepancies between different solutions increase at the higher Reynolds numbers, although the same general trend of a leveling off and then a slight decrease in the value of the stream-function is seen.

To study the 10,000 Reynolds number case in more detail, the streamlines are plotted in Fig. 3. The values of the stream-function contours for this plot are given in Table 4. The contour levels plotted correspond with the values plotted by Ghia *et al.* [30] for this case. Qualitatively, the plots appear to be identical. They each

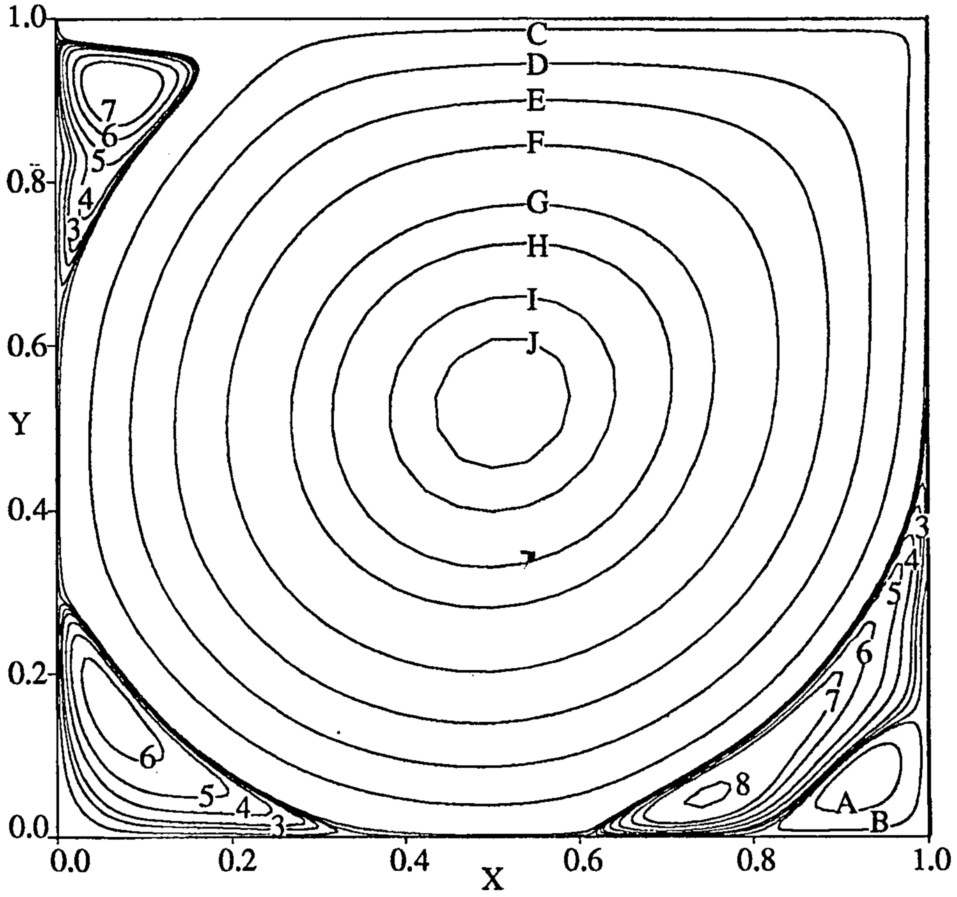


Fig. 3. Streamlines showing the driven cavity flow at Re = 10,000.

Table 4 Values for Streamline Contours in Fig. 3

Contour number	Value of ψ	Contour letter	Value of ψ
3	1.0×10^{-4}	A	-1.0×10^{-5}
4	2.5×10^{-4}	B	-1.0×10^{-4}
5	5.0×10^{-4}	C	-0.01
6	1.0×10^{-3}	D	-0.03
7	1.5×10^{-3}	E	-0.05
8	3.0×10^{-3}	F	-0.07
		G	-0.09
		H	-0.1
		I	-0.11
		J	-0.115

show secondary vortices of the same size and shape in the lower corners and the upper left corner. In Table 5, the stream-function $\psi_{v.c.}$, vorticity $\omega_{v.c.}$, and location of the vortex core $(x_{v.c.}, y_{v.c.})$ for all the secondary vortices for this 10,000 Reynolds number case are given for both the present results and the results of Ghia *et al.* [30]. In this table, the initial T stands for top, B stands for bottom, R stands for right, L stands for left, and the superscript number corresponds to the level of the secondary vortex. Thus BR^3 refers to the third and smallest secondary vortex found in the bottom right corner of the cavity, which is not seen in Fig. 3 because it is too small. Quite good agreement between the two computations is seen for this case, especially considering that the results of Ghia *et al.* [30] use over 10 times as many grid points (66,049 versus 6561).

Table 5 Properties of the Secondary Vortices for the Driven Cavity at Re = 10,000

Vortex	Results	$\psi_{v.c.}$	$\omega_{v.c.}$	$x_{v.c.}$	$y_{v.c.}$
TL	Present	2.418×10^{-3}	2.191	0.0723	0.9117
	Ghia *et al.* [30]	2.420×10^{-3}	2.183	0.0703	0.9141
BL		1.434×10^{-3}	2.097	0.0585	0.1686
		1.518×10^{-3}	2.086	0.0586	0.1641
BR		3.227×10^{-3}	4.163	0.7619	0.0585
		3.418×10^{-3}	4.053	0.7656	0.0586
BL^2		-5.120×10^{-7}	-0.02207	0.1416	0.0172
		-7.757×10^{-7}	-0.02754	0.1560	0.0195
BR^2		-2.103×10^{-4}	0.3726	0.9277	0.0729
		-1.313×10^{-4}	0.3126	0.9336	0.0625
BR^3		-4.267×10^{-7}	-2.956×10^{-3}	0.9981	0.0087
		-5.668×10^{-9}	–	0.9961	0.0039

The convergence toward a steady-state for this problem was very good for the

3 lowest Reynolds number cases, which required less than 150 iterations and only 50 seconds of computing time. The higher Reynolds number cases were slower to converge, the 10,000 case took 390 iterations and 150 seconds of computing time. The average of the computing requirements for all seven cases came out to 200 iterations and 75 seconds of computing time.

6.3 Flow Over a Backward Facing Step

A second two-dimensional problem which has been used as a validation case is the flow over a backward-facing step. This case was run using both upwind differencing and central differencing. The challenge in modeling this problem comes from the fact that the size of the separation bubbles downstream of the step are very sensitive to the pressure gradient in the flow. The geometry used in these calculations is shown in Fig. 4. At the inflow boundary, a parabolic profile is prescribed throughout the calculation, and the static pressure is allowed to change. Two step-heights downstream from the inflow a one-to-two expansion is encountered. The outflow boundary extends to 30 step heights downstream of the step, where characteristic boundary conditions are used along with the specification that the static pressure remain constant. The ability of the flow code to predict the reattachment length, x1, of the primary separation bubble behind the step as well as the separation and reattachment locations, x2 and x3, of the secondary separation bubble on the opposite wall was tested by comparing the computed results from both the upwind- and central-differencing schemes to experimental values given by Armaly *et al.* [34].

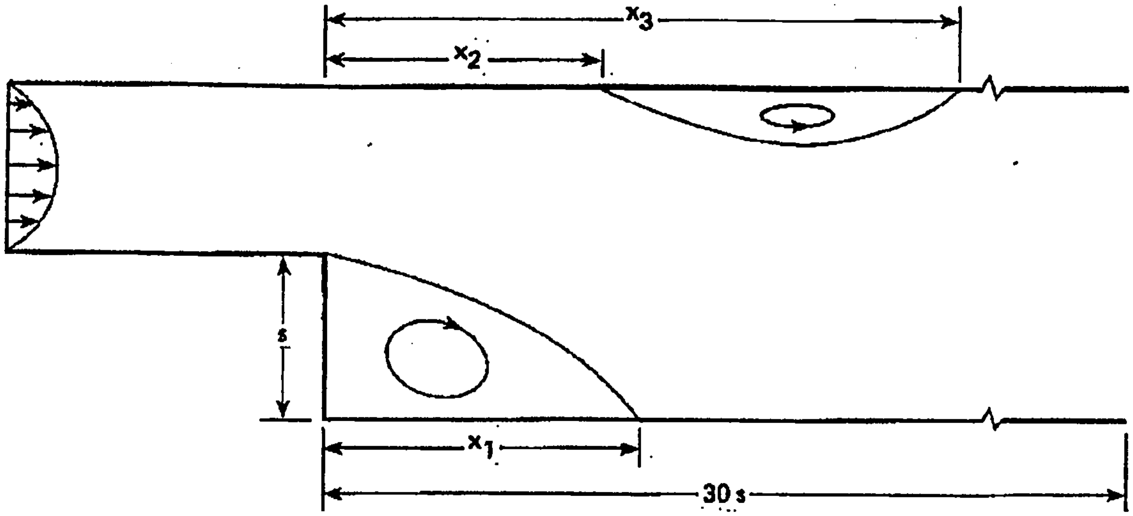


Fig. 4. Geometry of backward-facing step flow problem.

6.3.1 Upwind-Differencing Calculations

The upwind difference scheme was used to compute these quantities for the laminar range of Reynolds numbers from 100 to 800, which are based on the average inflow velocity and twice the step height. The flow was calculated using a grid of 100 points in the streamwise direction and 53 points in the cross-flow direction downstream of the step. The streamwise points were clustered toward the vertical-step face, and the cross-flow points were clustered at the walls. The initial conditions were specified to be freestream velocity at the interior points with uniform pressure everywhere. For the Reynolds numbers of 100 and 200, β was set to 1, for the Reynolds number of 300 case, 0.5 was used, and for the Reynolds numbers of 400 through 800, β was set to 0.1. The implicit line-relaxation process used 2 sweeps along the primary flow direction.

In Fig. 5, the quantities x1, x2, and x3 are plotted versus Reynolds number for both the present computed results and the experimental results of Armaly *et al.* [34]. Good agreement is seen between the two for the value of x1 at the lower Reynolds

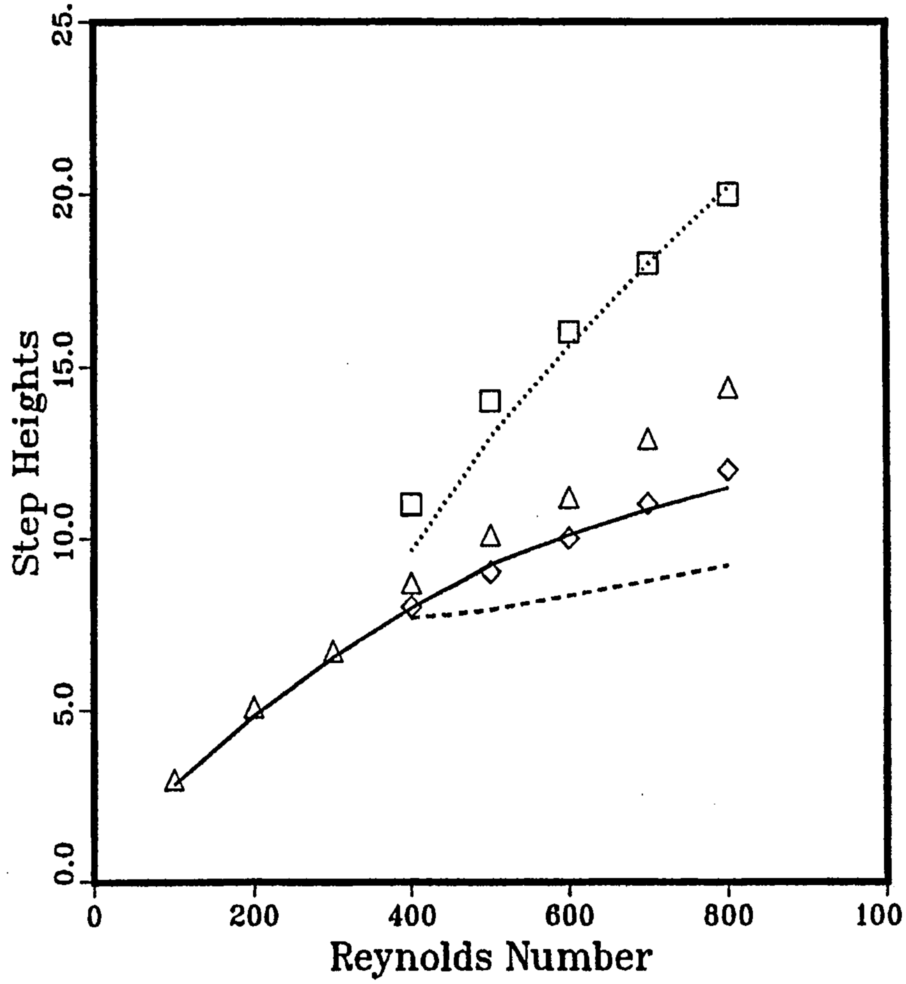
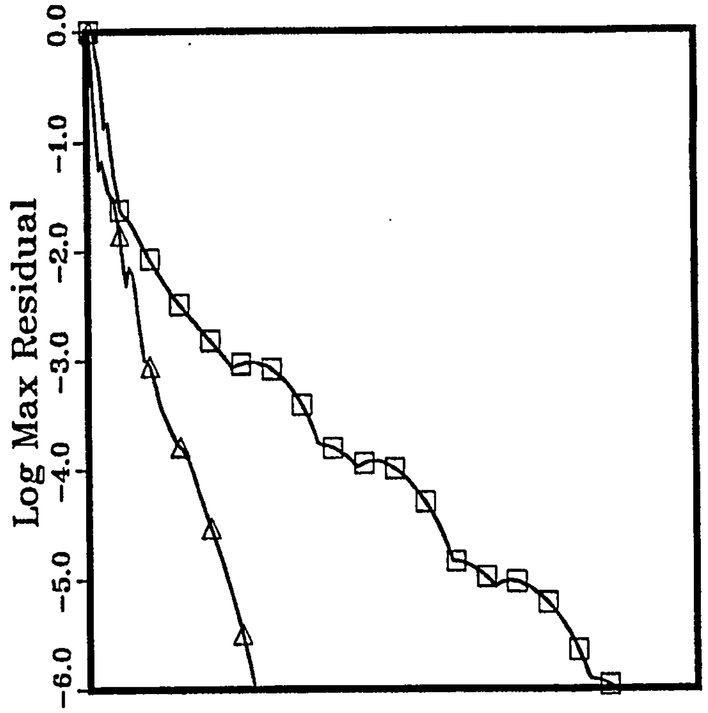


Fig. 5. Separation length versus Reynolds number for the flow over a backward facing step. Solid line: computed x1, dashed line: computed x2, dotted line: computed x3, △: experimental x1, ◇ : experimental x2, and □ : experimental x3.

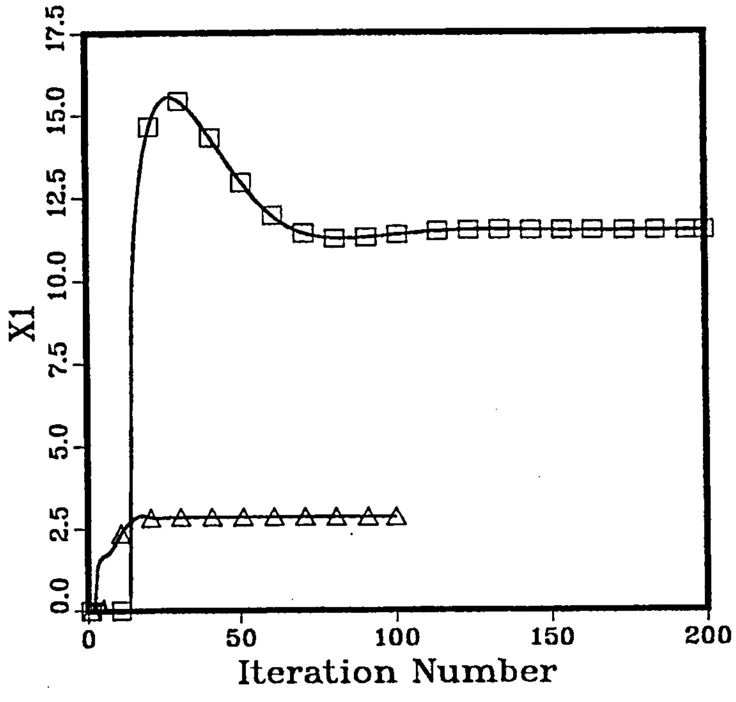
numbers before the secondary separation appears. At a Reynolds number of 400, the secondary separation bubble is present, and the computed primary reattachment length begins to fall off of the experimental curve. Similarly, the computed secondary separation distance is shorter than the experimental values, although they do agree in that the secondary separation point is upstream of the primary reattachment point. The computed secondary reattachment point is seen to be close the experimental values. In their experiment, Armaly *et al.* [34] reported that the flow was found to be three-dimensional near the step for Reynolds numbers greater than 400, and that the three-dimensional effects were negligible for lower Reynolds numbers. These three-dimensional effects could explain the discrepancies between the calculations and experiment.

Results similar to the present results were reported by Kim and Moin [6]. They reported a primary reattachment length of just under 12 step-heights for a Reynolds number of 800, and the present result for this Reynolds number is 11.48. They reported a secondary separation bubble size of 7.8 and 11.5 step-heights for Reynolds numbers of 600 and 800, respectively. The present results show secondary separation bubble sizes of 7.34 and 11.07 step-heights for these two Reynolds numbers. The similarities between these computational results give additional credence to the idea that the three-dimensionality of the experiment of Armaly *et al.* [34] was the reason for most of the discrepancies exhibited in Fig. 5.

The convergence characteristics of the upwind code for this problem are very good. In Fig. 6 the convergence histories of the Re = 100 and 800 cases are plotted. Figure 6a plots the log of the maximum residual normalized by the maximum residual at the first time step versus iteration number for the Re = 100 and 800 cases. Figure 6b plots the primary reattachment length x1 versus iteration number. The Re=100 case converges within 55 iterations and the Re=800 case converges within 165 iterations. The average number of iterations required for the 8 different Reynolds number cases is 104 and the average required computing time is just under 11.5 seconds.



6a. Log of the maximum residual versus iteration number.



6b. Primary reattachment length versus iteration number.

Fig. 6. Convergence history for the flow over a backward facing-step. △: Re = 100, □ : Re = 800.

6.3.2 Central-Difference Calculations

The central-difference scheme was used to compute the flow over a backward-facing step for a Reynolds number of 100, using the same grid, and the same initial and boundary conditions as the upwind-difference calculations. The quantity β was set to 2, and the pseudo-time step $\Delta\tau$ was set to 0.11. This was the largest value of $\Delta\tau$ for which the code would remain stable. The artificial dissipation coefficients ϵ_e and ϵ_i were set to 0.04 and 0.12, respectively. The separation bubble length was computed to be 2.84, which is less than 2% off the upwind-calculation value of 2.89. In general, the computational results from both methods agreed very well.

Although the final solution was nearly the same as the upwind case, it took 1675 iterations to converge, which took 164 seconds of computing time. So although the central difference flux vectors save about 15% of the total computing time cost per iteration, the overall computing costs are much greater. In addition, running this case took significantly more time by the investigator because of the need to specify the smoothing coefficients, as well as the maximum allowable time step, through trial and error testing. Since central differencing leads to equations which are far less diagonally dominant, the system is far less robust. It also means that a line-relaxation scheme may not be the best way to solve the system of equations.

Previous work by the author [11] in computing flow through this same geometry with the INS3D code gave similar convergence results as those seen here, and slightly better computing time requirements. The INS3D code uses central differences and approximate factorization, which indicates that approximate factorization may be a more appropriate solution scheme in conjunction with central differencing.

6.4 Steady Flow Over a Circular Cylinder

As an example of an external flow problem, the flow over a two-dimensional circular cylinder was calculated using both upwind and central differencing. The grid was an algebraically generated O-grid with 100 points in the circumferential direction and 60 points radially. The grid points were clustered in the radial direction toward the body, and the outer boundary was placed 20 diameters from the cylinder. The code was run and steady-state solutions were obtained for Reynolds numbers of 5, 10, 20, and 40, based on the freestream velocity and the cylinder diameter. At the outer boundary method of characteristic boundary conditions were employed. In addition, where the fluid was entering the domain, the velocity was held constant, and where fluid was leaving the domain, the static pressure was held constant.

For each case, several flow quantities were computed. Fig. 7 shows a schematic diagram of the geometry for this flow problem along with several of these flow quantities. These quantities are the flow separation length measured from the rear of the cylinder in cylinder diameters (L_{sep}), the angle which defines the point of separation from the body (θ_{sep}), the coefficient of drag (C_D), the coefficient of pressure drag (C_{Dp}), and the coefficient of pressure at the front (C_{pf}) and rear (C_{pr}) stagnation points.

6.4.1 Upwind-Differencing Calculations

In using the upwind-differencing scheme, the value of the artificial compressibil-

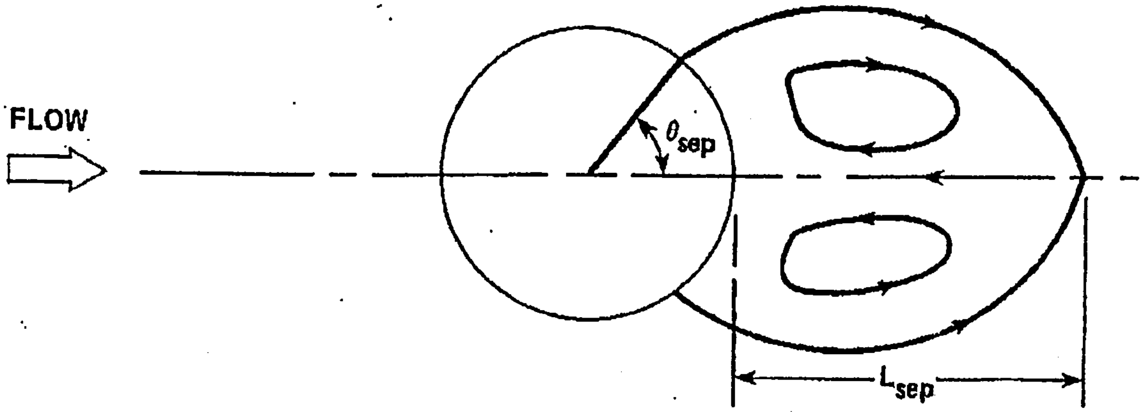


Fig. 7. Schematic diagram showing flow quantities for the circular cylinder flow computations.

ity constant β was set to 50 for all the cases. The line relaxation scheme used 4 sweeps in both coordinate directions, which seemed to be the best trade-off of convergence versus computing time. In Table 6, the flow quantities are presented for the present calculations as well as the numerical results of Takami and Keller [35], Dennis and Chang [36], Tuann and Olson [37], Braza *et al.* [38], and the experimental results of Coutanceau and Bouard [39] and of Tritton [40]. The comparisons show that there is very good agreement among nearly all the results. The results of the experiment, however, do not agree well with the computations. This could be because it is difficult to obtain accurate experimental results at such low Reynolds numbers. The convergence of the code toward a steady-state solution for this case was found to be quite good. All four Reynolds number cases converged in less than 70 iterations, requiring an average of 21 seconds of computing time.

6.4.2 Central-Differencing Calculations

The Reynolds number 5 case was run using the central-differencing scheme. The

Table 6 Flow Quantities for a Circular Cylinder

Source	Reynolds Number			
	5	10	20	40
L_{sep}				
Present	0	0.254	0.932	2.29
Ref. 35	0	0.249	0.935	2.32
Ref. 36	0	0.252	0.94	2.35
Ref. 37	0	0.25	0.9	2.1
Ref. 39 (exp)	–	0.34	0.93	2.13
θ_{sep} (degrees)				
Present	0	28.8	43.1	53.0
Ref. 35	–	29.3	43.7	53.6
Ref. 36	0	29.6	43.7	53.8
Ref. 37	> 6	29.7	44.1	54.8
Ref. 38	–	–	43.6	54.5
Ref. 39 (exp)	–	32.5	44.8	53.5
C_D (C_{Dp})				
Present	4.18 (2.19)	2.89 (1.602)	2.08 (1.242)	1.549 (1.011)
Ref. 35	–	2.80	2.01	1.536
Ref. 36	4.12 (2.20)	2.85 (1.600)	2.05 (1.233)	1.522 (0.998)
Ref. 37	4.66 (2.48)	3.18 (1.775)	2.25 (1.35)	1.675 (1.095)
Ref. 38	–	–	2.18	1.60
Ref. 40 (exp)	4.16	3.06	2.02	1.65
C_{pf} ($-C_{pr}$)				
Present	1.847 (1.067)	1.476 (0.755)	1.265 (0.615)	1.147 (0.536)
Ref. 35	–	1.474 (0.670)	1.261 (0.537)	1.141 (0.512)
Ref. 36	1.872 (1.044)	1.489 (0.742)	1.269 (0.589)	1.144 (0.509)
Ref. 37	2.23 (1.081)	1.744 (0.773)	1.457 (0.614)	1.312 (0.543)

artificial dissipation coefficients ϵ_e and ϵ_i were set to 0.01 and 0.03, respectively. The line-relaxation scheme used 2 sweeps in the circumferential direction, and the artificial compressibility constant β was set to 50. The largest pseudo-time step $\Delta\tau$ for which the code remained stable was 0.01. This lead to very slow convergence. After 2000 iterations the solution was very nearly converged. Again, the solution was very close to that of the upwind-differencing calculations. The computing time was about 260 seconds, close to 10 times that required by the upwind scheme for this case.

6.5 Oscillating Plate

As an initial unsteady test case, the motion of fluid on top of an infinite plate which oscillates back and forth its own plane was calculated. This problem, also known as Stoke's second problem (see for example White [41]), was set up with the x-axis along the plate, and the y-axis normal to it. The velocity of the plate is set to

$$u_{plate} = u_0 \cos\omega t$$

where u_0 is the maximum plate speed and ω is the frequency. The exact solution for this problem is given by

$$u(y,t) = u_0 \exp\left(-y\sqrt{\frac{\omega}{2\nu}}\right) \cos\left(\omega t - y\sqrt{\frac{\omega}{2\nu}}\right) \tag{6.1}$$

where ν is the kinematic viscosity. This is a 1-D problem in that there are no gradients in the direction parallel to the plate and so only η gradient terms were used. The mesh extended to a height where the exact solution for the velocity is less than $5\text{x}10^{-4}u_0$ and consisted of 50 points in the y-direction stretched such that the spacing at the plate was 0.002 of the total height. The velocity u_0 was set to unity, the frequency was set to 2π, and the square root term was set to unity. Thus

$$\sqrt{\frac{\omega}{2\nu}} = 1 \longrightarrow \nu = \pi$$

The physical time step Δt was set to 0.01. Since the vertical velocity component v is always zero, and there are no gradients in the x-direction, the divergence of velocity is always zero, and thus only one iteration is required to advance the equations one step in time. In fact, after one iteration, the maximum value of the right-hand side of Eq. (2.15) was reduced to 10^{-10}.

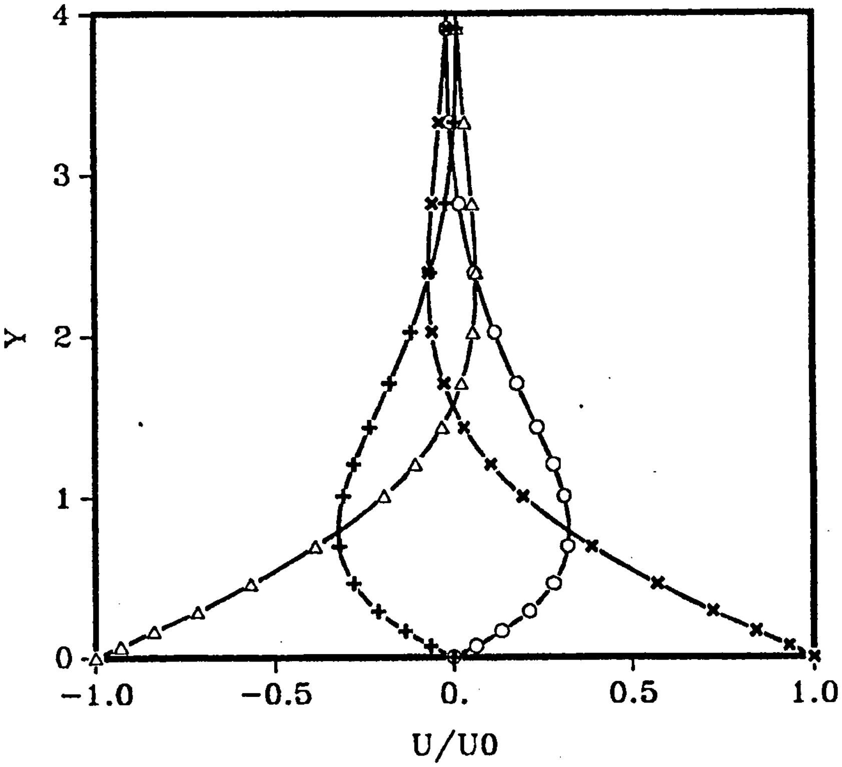


Fig. 8. Velocity profiles for flow over an oscillating plate. Solid line: computed solution; Symbols: analytical solution; o: t=8.25; △: t=8.5; +: t=8.75; ×: t=9.

The problem was run over nine full periods after which the transient solution had died off and the solution was fully periodic in time. In Fig. 8 velocity profiles at different times during the cycle are plotted for both the computed solution and the exact analytical solution. The computed solution is designated by the solid lines and the exact solution by the symbols. Excellent comparison is seen between the two.

6.6 One-Dimensional Channel

The flow through an inviscid 1-D channel with an oscillating back pressure was calculated. This problem was also used by Merkle and Athavale[5] to verify their method. The present work used the 2-D code by setting all gradients in the y-direction to zero and using an equally spaced grid of 21 points in the x-direction. At the inflow constant total pressure was specified, and the static pressure at the outflow was set to be

$$p_{exit} = p_0 + p_e \sin\omega t$$

where p_e is the amplitude of the pressure oscillations and ω is the frequency. An exact analytical solution exists for this as long as the magnitude of the back-pressure oscillation p_e remains small compared to the mean back pressure p_0. For this problem, the ratio of p_e to p_0 is taken to be 0.1. The flow was calculated for a channel length of unity and a mean velocity of unity. In this case the solution is given by

$$\begin{aligned} u(t) &= 1 - \frac{p_e}{1+\omega^2}\left(\sin\omega t - \omega\cos\omega t - \omega e^{-t}\right) \\ p(x,t) &= p_0 + p_e \sin\omega t + (x-1)\frac{p_e\omega}{1+\omega^2}\left(\cos\omega t + \omega\sin\omega t + e^{-t}\right) \end{aligned} \tag{6.2}$$

Both the pressure and velocity have exponentially decaying terms corresponding to the initial transient, as well as the periodic sine and cosine terms. The velocity is a function of time only, which is a direct effect of the incompressible continuity condition in a constant area duct.

The initial conditions for the problem were taken to be the velocity and pressure given by Eqs. (6.2) evaluated at t=0. The problem was calculated for six different frequencies ranging from 0.01 to 100.0. The physical time step was taken so that 30 steps would be used for one period of the back-pressure oscillation. Thus

$$\Delta t = \frac{\pi}{15\omega}$$

The artificial compressibility constant β was set to 10. Each sub-iteration in pseudo-time was converged until the maximum divergence of velocity at any point was less than 10^{-6} and the maximum element of the right-hand side of Eq. (2.15) was less than 10^{-6}. This required five to eight subiterations to obtain this convergence for all cases calculated.

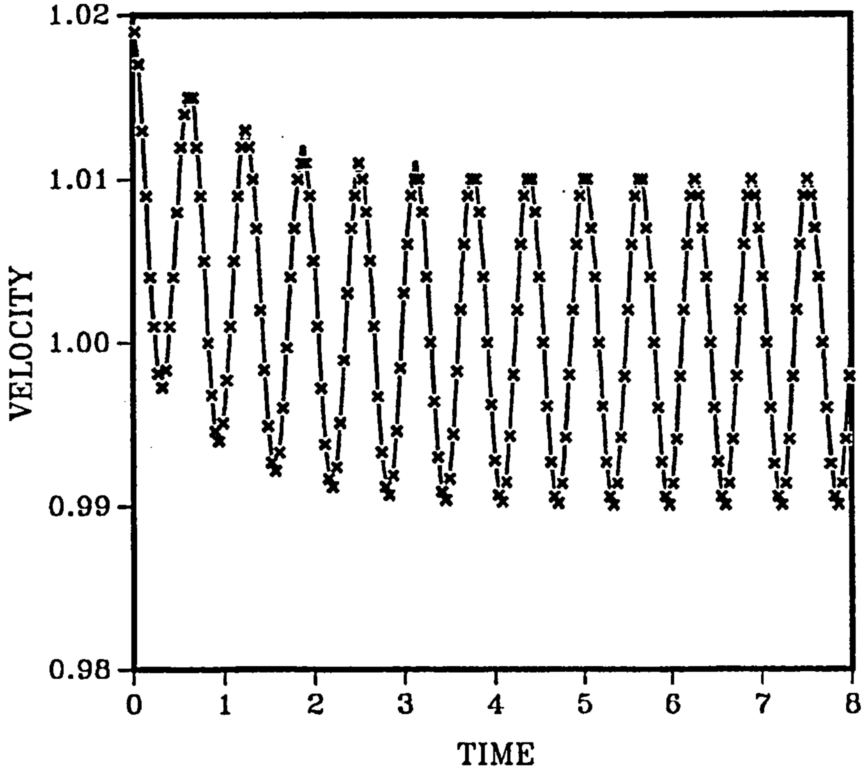


Fig. 9. Velocity for the flow through an inviscid channel with oscillating back pressure. Solid line: computed solution; ×: analytical solution.

The velocity is plotted as a function of time in Fig. 9. Shown are the computed values (solid line), and the exact values (× symbol) for a frequency of $\omega = 10$. The transient is seen to die out after about five periods, after which simple harmonic motion occurs. The velocity results compare very well with each other. For example, the computed mean velocity for the periodic motion is equal to 1.00035, and the exact mean velocity is 1. The values of the computed pressure are seen to be quite accurate as well. The maximum error of the computed pressure was measured for all cases run and was found to average less than 10^{-4}, which corresponds to an error of about 0.01 percent. This error is proably due to truncation.

6.7 Vortex Shedding Behind a Circular Cylinder

The flow over a circular cylinder was computed in an effort to induce vortex shedding in the wake. Computations were carried out at two different Reynolds numbers, 105 and 200, which are discussed as separate cases.

6.7.1 Reynolds Number 200

The Reynolds number 200 case was run first and quantitative measurements of the flow field were taken. This case was run using a 60 by 100 O-grid with points clustered near the body such that the spacing next to the surface was 0.0035 diameters. The outer boundary extended to 10 diameters from the cylinder, a physical time step Δt of 0.025 was used, and the artificial compressibility constant β was set to 2500. Both the fifth- and third-order accurate convective flux differences were used.

At each time step the subiterations were carried out until both the maximum divergence of velocity at any point and the maximum residual at any point were less than 10^{-4}. During the initial transient, 20 to 30 subiterations were required for each time step, but this quickly decreased to an average of 5 subiterations per time step. Once an asymmetric wake developed, 12 to 14 subiterations were required.

The flow was started impulsively from free-stream conditions and run until a periodic shedding of vortices occurred. The vortex shedding developed without any type of artificial disturbance. This is most probably due to the roundoff error adding some very small asymmetric disturbances which will trigger the asymmetric motion. However, the fifth-order differencing calculations started to develop an asymmetric wake within a nondimensional time of 50 and was completely periodic by a time of 100, whereas the additional dissipation of the third-order scheme delayed the formation of a completely periodic solution until a nondimensional time of 180.

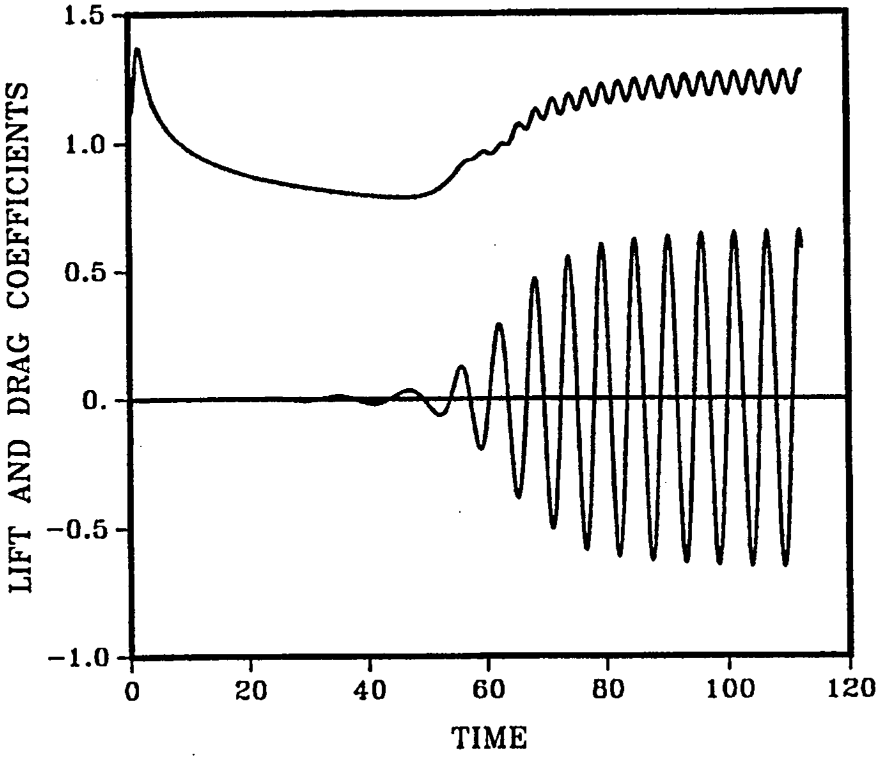


Fig. 10. Lift and drag coefficients versus time for flow over a circular cylinder at a Reynolds number of 200.

The lift and drag coefficients for the fifth-order calculations are plotted versus time in Fig. 10. The time history for the third-order calculations is quite similar to this plot. However the quantitative results do vary. This is shown in Table 7 where the values of the lift and drag coefficients for the periodic state are listed, as well as the Strouhal number. Also listed in this table are the results from several other calculations by Rosenfeld [42], Lecointe and Piquet [43], Martinez [44], Lin *et al.* [45], and Thoman and Szewczyk [46]. Experimental results of Wille [47], Kovasznay [48], and Roshko [49] are included as well. The present fifth-order results are in general agreement with most of the other results. Values for the Strouhal number near 0.19 appear to be consistent, and so the Strouhal number of the present result for the third-order differencing is to be too low. The difference between the third- and fifth-order schemes is attributed to the difference in the amount of numerical dissipation which is added by each scheme.

Table 7 Lift and Drag Coefficients and Strouhal Numbers for Circular Cylinder Flow at Reynolds Number 200

	C_D	C_L	St
Present 3rd order	1.29 ± 0.05	± 0.75	0.16
Present 5th order	1.23 ± 0.05	± 0.65	0.185
Rosenfeld [42]	1.46 ± 0.05	± 0.69	0.211
Lecointe & Piquet [43]			
2nd order	1.46 ± 0.04	± 0.70	0.227
4th order	1.58 ± 0.0035	± 0.50	0.194
Martinez [44]	1.27 ± 0.0035		
Lin *et al.* [45]	1.17		
Thoman & Szewczyk [46]	1.17 ± 0.005		
Wille [47] (exp)	1.3		
Kovasznay [48] (exp)			0.19
Roshko [49] (exp)			0.19

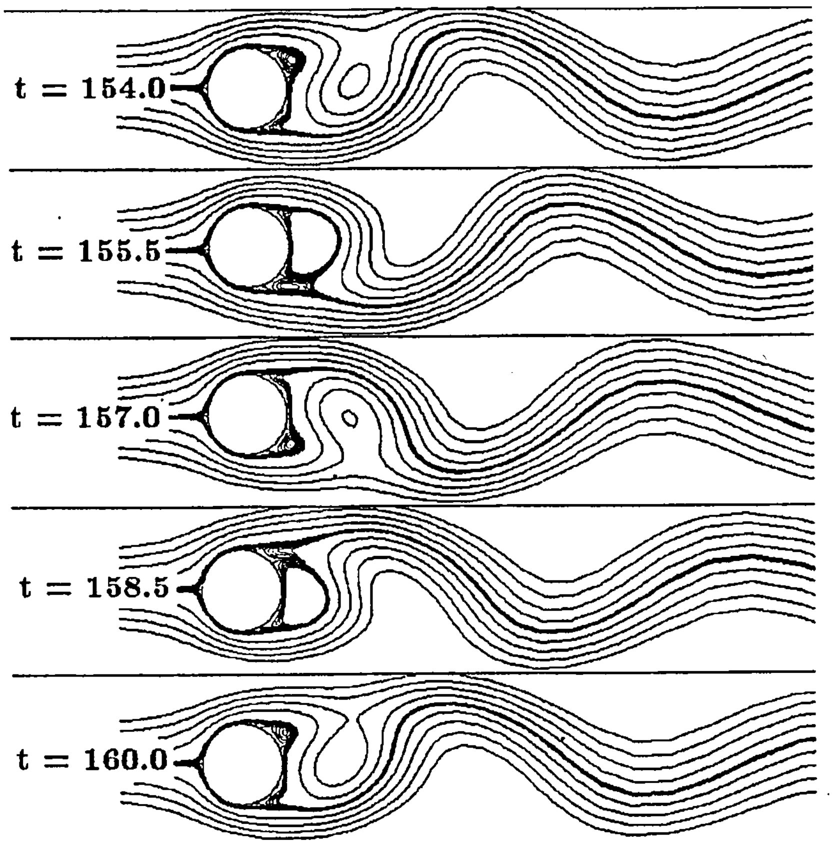


Fig. 11. Streamlines for flow over a circular cylinder at a Reynolds number of 200 at various times during the vortex shedding cycle for the third-order differencing scheme.

In Fig. 11, the streamlines at various stages during one period are plotted for the third-order calculations. The first plot shows the flow when the drag is at a minimum and the lift is zero. It can be seen that a new vortex is forming on the top half, while the low-pressure center of the previous vortex has pulled away from the body causing the drop in drag. The second plot shows that the top vortex has now extended itself across the entire back side of the body, causing this point in time to have a maximum in drag and lift. The next two plots correspond to another minimum in drag with zero lift, and a maximum in drag with a minimum in lift, respectively. These two are mirror images of the first two plots. Finally, the last plot is nearly identical to the first one, and is only different because it is taken at a time of 6 after the first plot, whereas the actual period for this flow is 6.25. The main qualitative flow features shown here are the same as are seen in the fifth-order calculations.

6.7.2 Reynolds Number 105

The code was next run for a circular cylinder at a Reynolds number of 105 in an effort to see how well the Kármán vortex street is captured by the upwind differencing scheme. The grid dimensions were increased to 120 by 240 points in the radial and circumferential directions, respectively, and the outer boundary was extended to 25 diameters away from the cylinder. A physical time step of 0.05 was used, and the value of β was set to 1100. The fifth-order convective flux differences were used. In order to reduce the computing time required to see the vortex shedding, an artificial velocity disturbance along one side of the cylinder was added. The surface velocity on one side was prescribed to be sinusoidal in time with a non-dimensional frequency of 0.16 and a magnitude of one quarter of the freestream velocity. This triggered shedding right away, and the disturbance was turned off after a non-dimensional time of 15.7, after which time the vortex shedding continued and a completely periodic solution was obtained within several cycles. With this finer grid, an average of 30 subiterations were required at each time step. The computing time requirements for this case were about 45 minutes per periodic cycle.

Streaklines were computed for this periodic solution at a non-dimensional time of 58 and are shown in Fig. 12. The lower half of this figure shows an experimental picture of the same flow conditions by Sadatoshi Taneda reproduced from Van Dyke

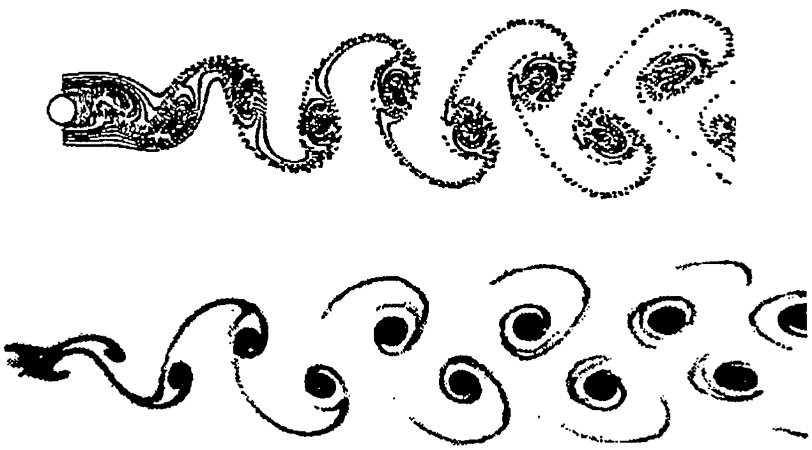


Fig. 12. Comparison between computational and experimental streaklines for the flow over a circular cylinder at a Reynolds number of 105.

[50]. The streaklines in the experiment are shown by electrolytic precipitation in water. The vortex structures are seen to be very similar.

6.8 Square Duct With 90° Bend

The flow through a square duct with a 90° bend was used as a steady-state 3-D test case. This particular geometry was studied experimentally by Humphrey *et al.* [51] which enables comparisons to be made with the current computed results. Four different grids were used whose dimensions are 31 x 11 x 11, 41 x 21 x 21, 51 x 31 x 31, and 61 x 41 x 41. The problem was non-dimensionalized using the side of the square cross-section as the unit length, and the average inflow velocity as the unit velocity. The Reynolds number was 790 based on the unit length and velocity, and the artificial compressibility parameter β was set to one for all cases. The grid for the 31 x 11 x 11 case is shown in Fig. 13. The straight inflow section before the bend was set to a length of five and the outflow section downstream of the bend was also set to a length of five. The radius of curvature of the inner wall in the curved section was 1.8 units

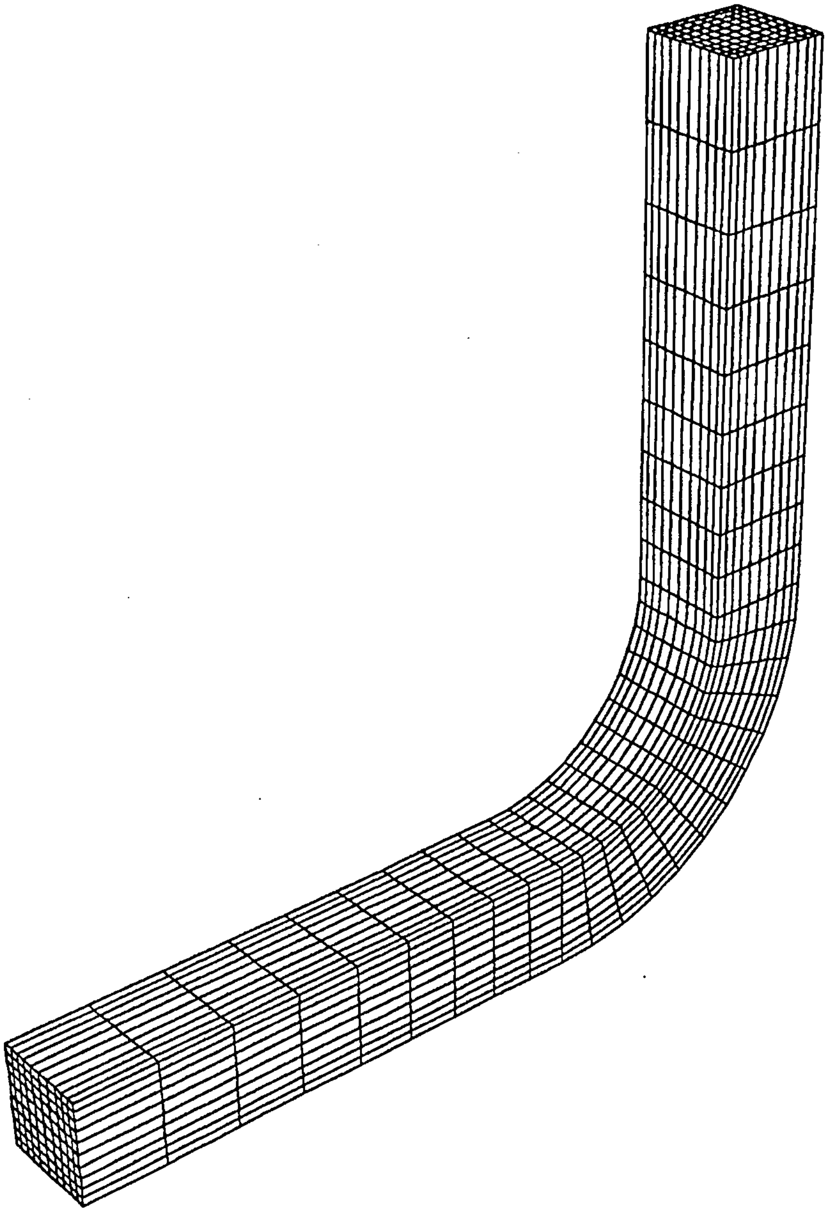


Fig. 13. Geometry and grid (31 x 11 x 11) for computation of flow through a square duct with a 90° bend.

in length. The inflow velocity profile was prescribed to be that of a fully developed laminar straight square duct as given by White [52].

The convergence history for this steady-state problem is shown in Fig. 14. The log of the maximum residual over all the grid points is plotted versus iteration number for all four grid cases. The convergence is shown to be very fast. Although not shown

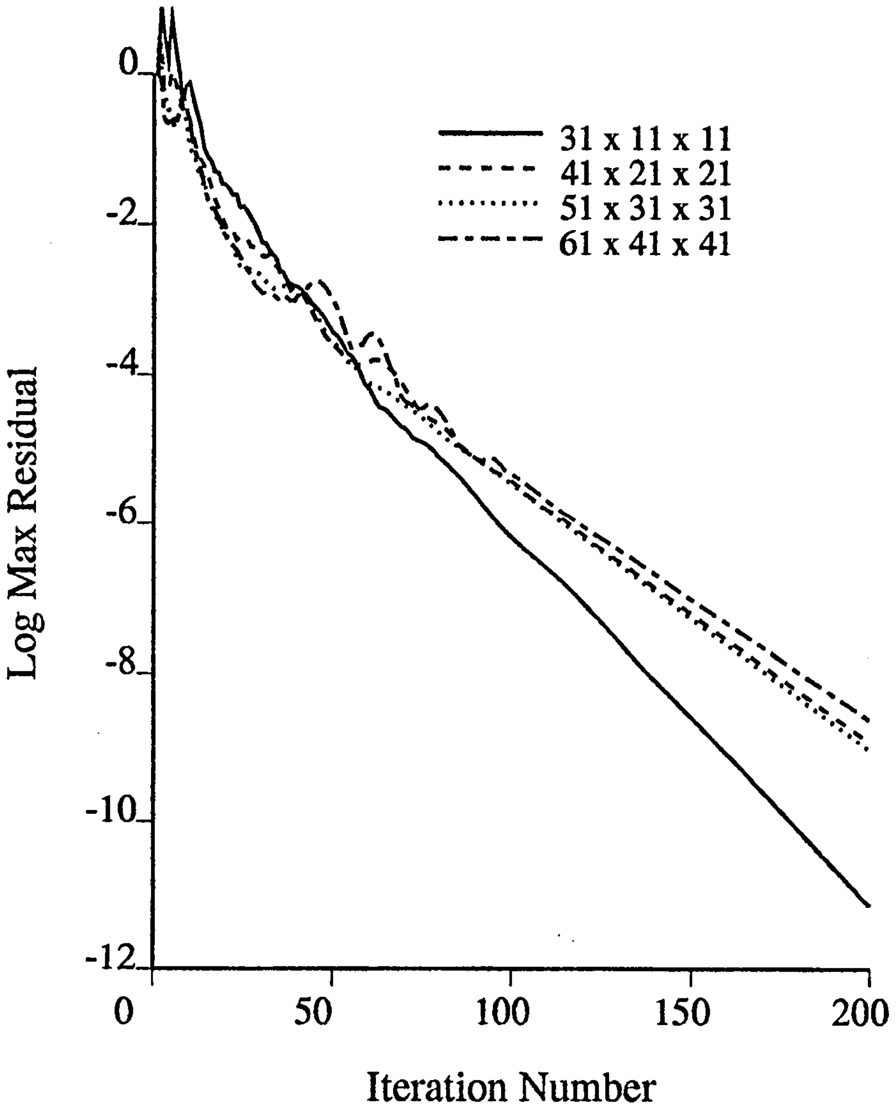


Fig. 14. Maximum residual over all points versus iteration number showing convergence history for the flow through a square duct with a 90° bend.

in the figure, machine zero is reached in less than 300 iterations for all four cases. The solution is considered converged if the maximum residual has converged at least four orders of magnitude and the maximum divergence of velocity is less then 10^{-4}. This is obtained in less than 115 iterations for all cases. The computing times required for convergence for the four cases are 32, 196, 550, and 1067 sec, respectively.

The computed results are compared to the experimental results of Humphrey *et al.* [51] in Fig. 15. Shown are the longitudinal velocity profiles at various streamwise stations for two different cross-flow locations. In Fig. 15a, the velocity profiles are taken at z=0.25, that is half way between the x-y plane wall and the x-y symmetry plane. The second location, shown in Fig. 15b, is from the x-y plane at z=0.5, or the x-y symmetry plane. In each of these figures, the profiles are shown at x=0 (the inflow boundary), at x=2.5 (half-way between the inflow boundary and the start of the curved section), and at four positions in the curved section corresponding to θ equal to 0, 30, 60, and 90.° The symbols represent the experimental results and the lines represent the computed solutions. The computations for the two finest grid cases are seen to be in very good agreement, indicating that these represent a grid-independent solution. Good comparison is seen between the computation and the experiment, particularly for the first four streamwise stations. However, at the latter part of the bend, the fine-grid computations begin to predict two relative minima in the velocity profiles, which is not evident in the experimental results. It is seen that the computations are able to predict the maximum longitudinal velocity very well.

Figure 16 shows the cross-stream velocities for the 51 x 31 x 31 grid case at the streamwise points of $\theta = 30°, \theta = 60°$, and $\theta = 90°$. This figures shows how a pair of secondary vortices are generated by the large values of static pressure on the outside wall, which arises when the flow starts to negotiate the sharp bend. The center of these vortices is seen to move towards the inside wall between the $\theta = 30°$ station and the $\theta = 60°$ position. The vortices tend to center again further downstream, and at the

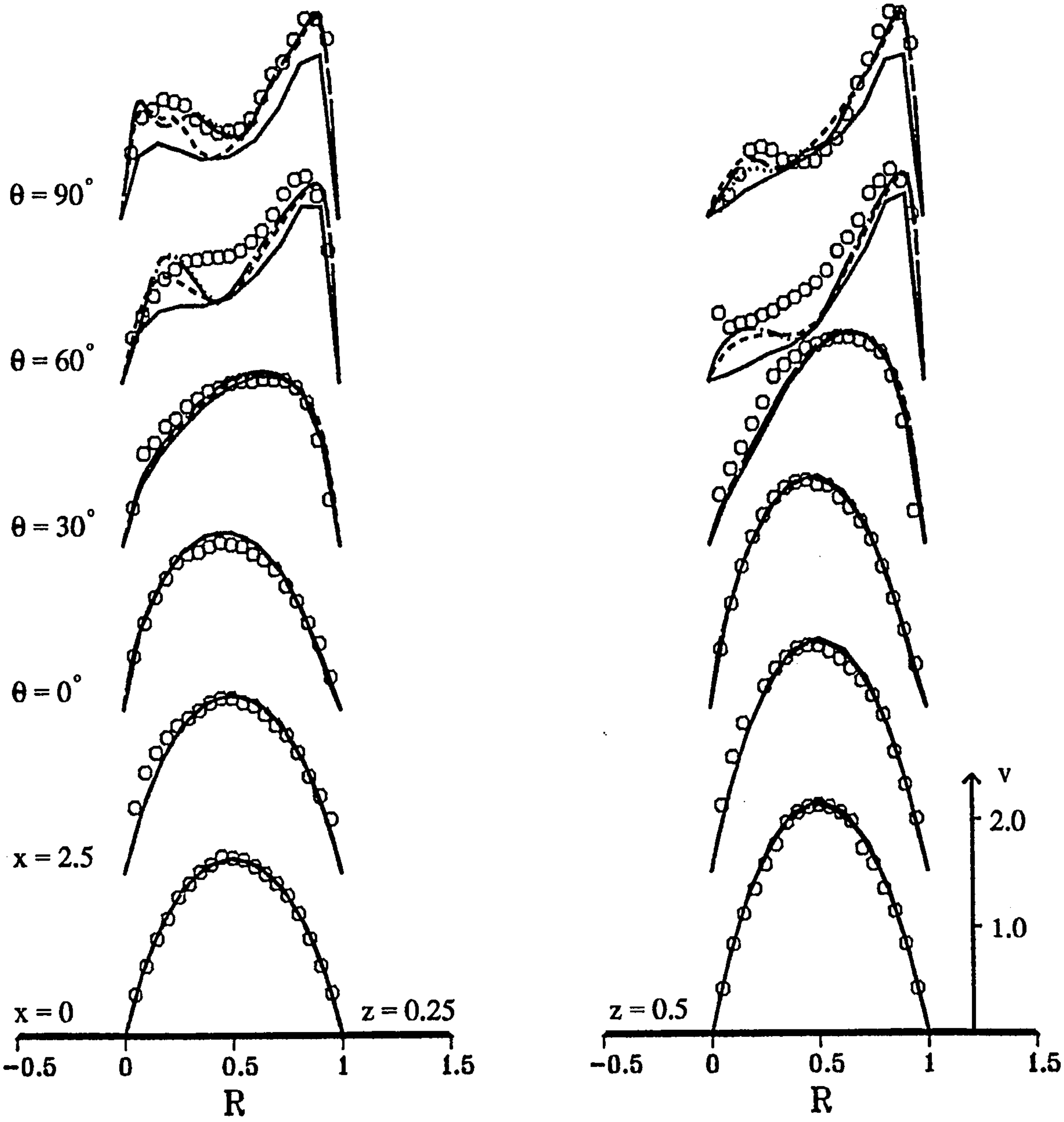


Fig. 15. Streamwise-velocity profiles at the streamwise stations given by x=0, x=2.5, $\theta = 0°, \theta = 30°, \theta = 60°$, and $\theta = 90°$. Solid line = 31 x 11 x 11 grid, dash line = 41 x 21 x 21 grid, dotted line = 51 x 31 x 31 grid, chain-dot line = 61 x 41 x 41 grid, and o = experiment [51].

same time a secondary pair of vortices are seen to appear. This agrees qualitatively with the observations of Humphrey *et al.* [51].

Figure 17 shows velocity vectors for the 51 x 31 x 31 grid case at the k=2 grid plane next to the outer-radius wall between the streamwise positions of $\theta = 0$ and $\theta =$

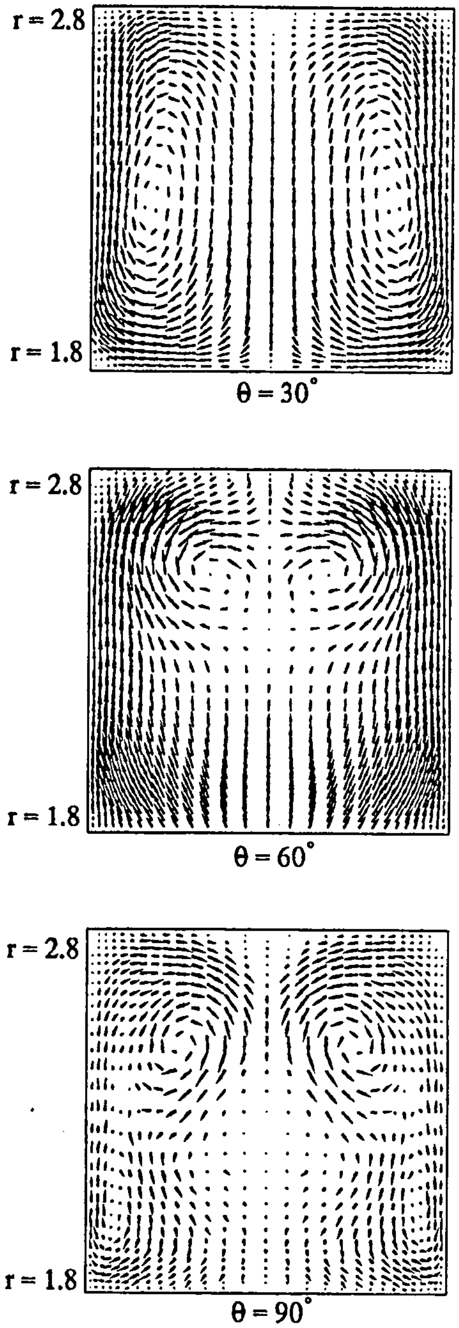


Fig. 16. Cross-sectional velocities for the curved square duct at the streamwise stations given by $\theta = 30°, \theta = 60°$, and $\theta = 90°$ for 51 x 31 x 31 grid.

60.° These show the presence of a recirculation region in the corners of the duct

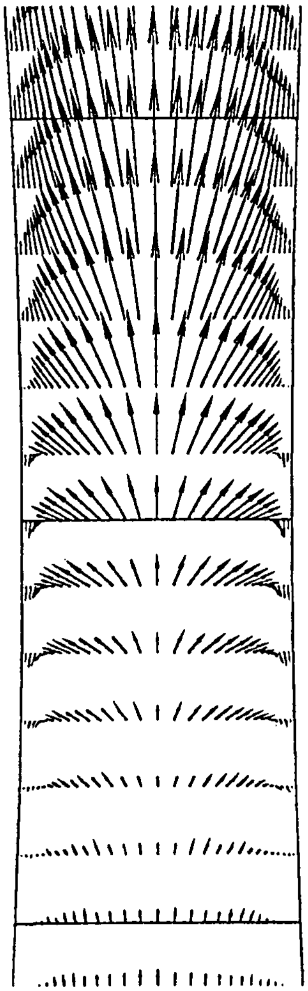


Fig. 17. Velocity vectors next to outer-radius wall showing reverse flow in curved square duct for 51 x 31 x 31 grid.

appearing between $\theta = 0$ and $\theta = 35.°$ In their experiment, Humphrey *et al.* [51] reported recirculating flow in the corners between approximately $\theta = 0$ and $\theta = 25.°$

6.9 Artificial Heart Flow

The present flow solver has been used to compute the flow inside an artificial heart. The artificial heart was designed by Penn State University and is being studied experimentally by Tarbell *et al.* [53]. The purpose of the current calculations is to demonstrate and analyze the present capability to compute a time-accurate incompressible flow through a complex internal device with moving-boundaries. The initial calculations in this effort are presented here. Since a number of primarily geometrical differences exist between these initial calculations and the actual artificial heart experiment, which are detailed below, only the simplest of qualitative comparison between computation and experiment can be made. Further work will attempt to remove these differences so that the actual artificial heart geometry can be accurately modeled.

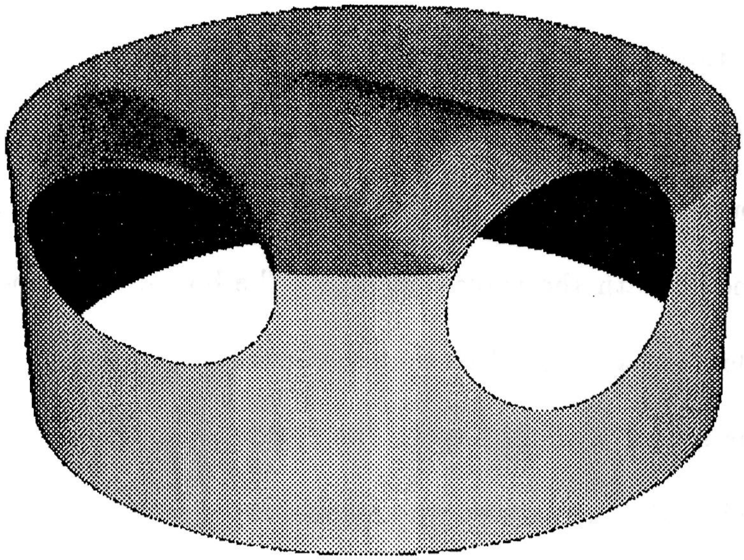


Fig. 18. Artificial-heart geometry showing valve openings.

The geometry used for the current model is depicted in Fig. 18. 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 diameter of the piston is 7.4 cm, with a stroke length of 2.54 cm. The problem was nondimensionalized with a unit length of 2.54 cm and a unit velocity of 40 cm/sec. The actual artificial heart has cylindrical tubes extending out of each of the side valve openings. These contain tilting flat disks which open and close to act as the valves. In the computational model these valves are not modeled, instead the boundary conditions at the side openings are specified to instantaneously open and close at the right moment. This simplification allows a single zone to be used to model the flow inside the chamber. With the addition of more zones, however, it will be possible to model the tilting disk valves as well. Additional simplifications made in the present computational model include the movement of the piston. In the actual device the piston moves through the entire chamber volume, including across the majority of the valve openings. Because of grid-generation problems, moving the piston past the valve openings would become quite difficult, and so the piston is restricted to move only in the volume beneath the valve openings. The bottom most position of the piston is extended lower than normal so that the entire piston displacement volume is close to that in the actual artificial heart. The flow is assumed laminar, and the Reynolds number based on the the unit length and velocity is set to 100. In the actual heart the Reynolds number is about 600, and regions of the flow are turbulent. Finally, the fluid is assumed to be Newtonian. This corresponds to the experiment of Tarbell *et al.* [53] who used a water and glycerin fluid whose viscosity is nearly the same as blood, about 3.5 centipoise, but unlike blood can be simulated by a Newtonian fluid assumption.

Inside the heart an H-H grid topology is used with dimensions of 39 x 39 x 51 Figure 19 shows the grid on the unwrapped surface of the side of the heart chamber. Grid lines were placed along the lines of the valve openings to make boundary condition implementation straight forward. This surface grid was generated by dividing the surface into several zones and using a biharmonic grid generator [54] in each zone.

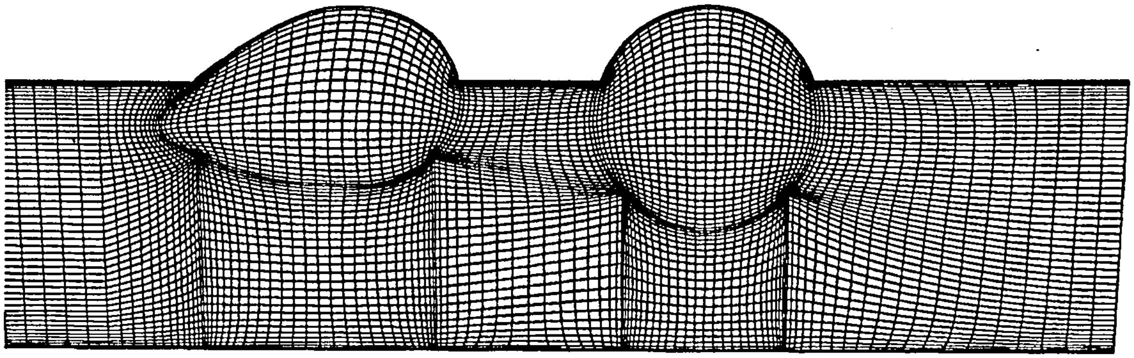
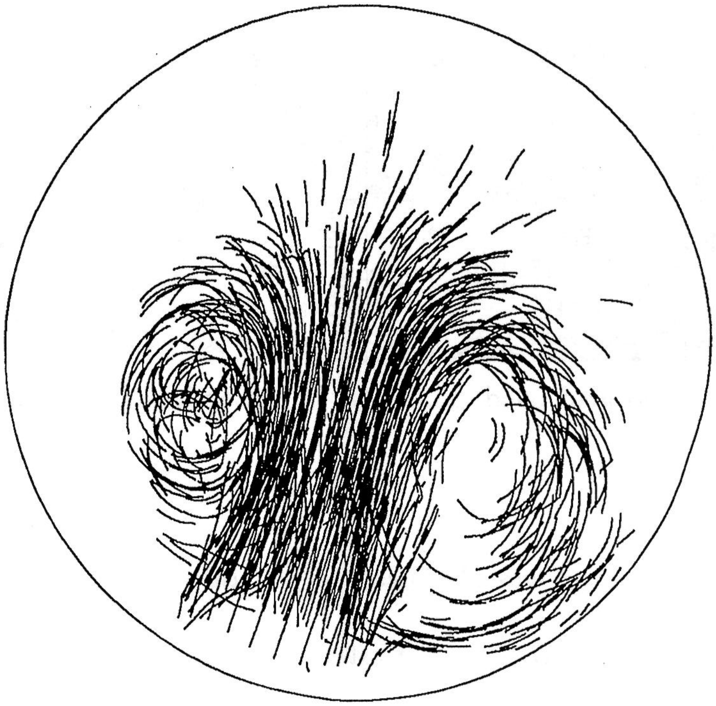


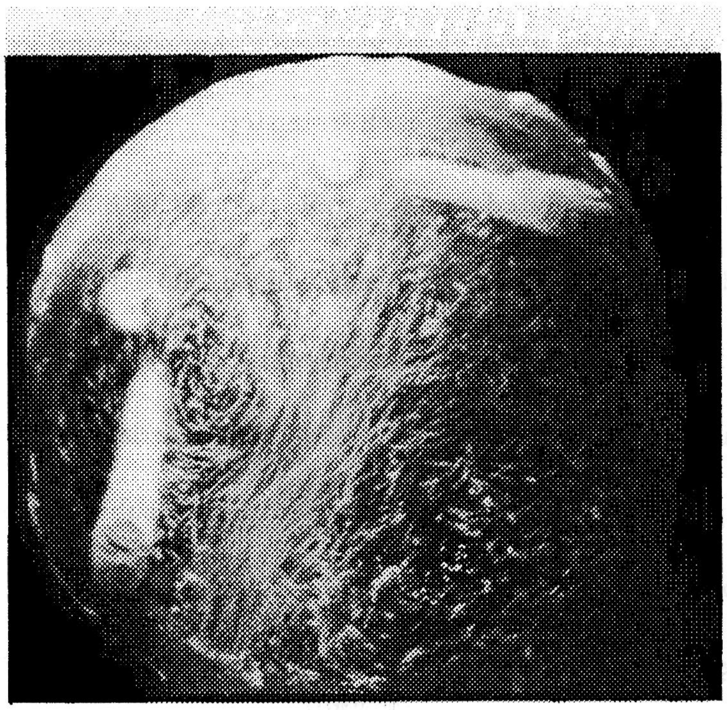
Fig. 19. Unwrapped grid used along side of the artificial heart near the region of the valve openings.

The same grid generator was also used to generate an H grid on the piston surface and on the top surface. To fill in the interior points, an algebraic solver coupled with an elliptic smoother was used. As the piston moved up and down inside the chamber, the grid points below the valve openings were compressed and expanded, respectively. Thus a new grid was generated at each time step.

The flow was computed using a time step Δt of 0.025, and a β of 500. The piston moved with a constant nondimensionalized velocity of ± 0.2 between its top and bottom positions, thus requiring 200 physical time steps for one period of the piston's motion. During each time step, the subiterations were carried out until the maximum residual dropped below 10^{-3} or until a maximum of 20 subiterations were used. During most of the piston's cycle only 12-15 subiterations were required, but when the piston was changing directions, it did not completely converge in 20 subiterations. This did not cause any stability problems, yet it remains to be seen what effect this might have had on the time accuracy. The computing time required for each period of the piston's motion was approximately four hrs. The computations were run for four periods during which particle paths were computed after being released near the inflow valve. Figure



20a. Computation



20b. Experiment

Fig. 20. Incoming particle traces from computations and picture of experimental results as the piston nears the bottom position.

20a shows some of these 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 Fig. 20b, an experimental photograph (Tarbell, J. M.: private communication, 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 21 and 22 show velocity vectors during the inflow phase in planes passing through the center of the inflow valve. The first of these shows a top view of vectors in a plane parallel to the piston, while Fig. 22 shows a side view of vectors in a plane perpendicular to the piston. These figures portray how complicated the vortical structure of this flow is. Figure 21 again shows the presence of two vortices formed as the incoming flow forms a jet. Figure 22 shows also how the flow separates underneath the valve opening, but what it does not show is that the flow there is strongly 3-D with the velocity vectors next to the left wall underneath the valve pointing into the paper. Also in the figure is seen the presence of additional vortices against the back wall opposite the valve opening.

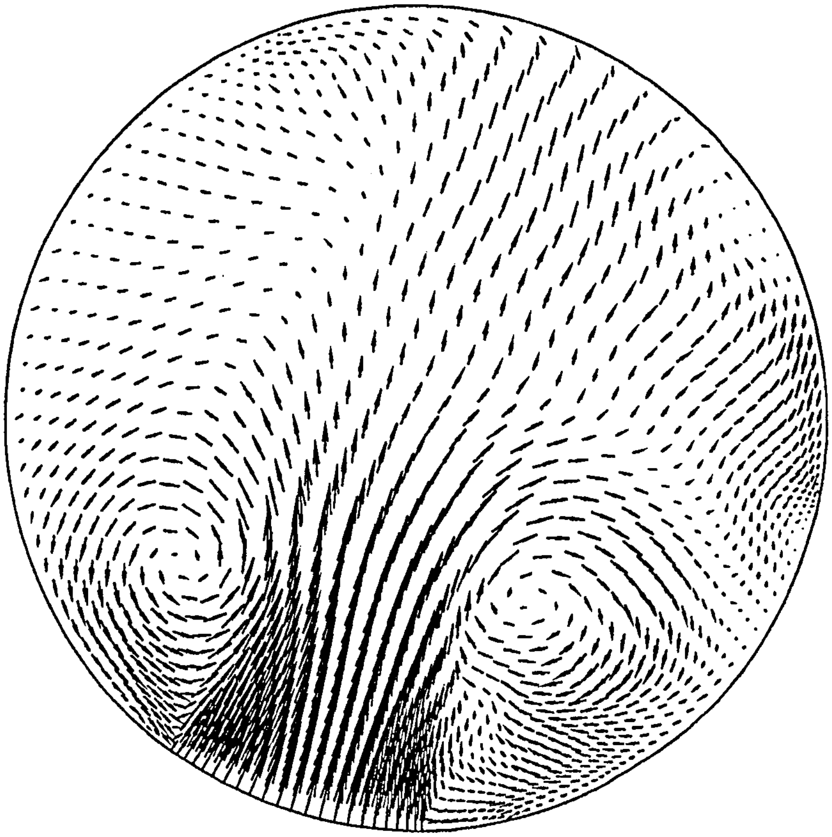


Fig. 21. Top view of velocity vectors in plane through center of inflow valve showing incoming fluid.

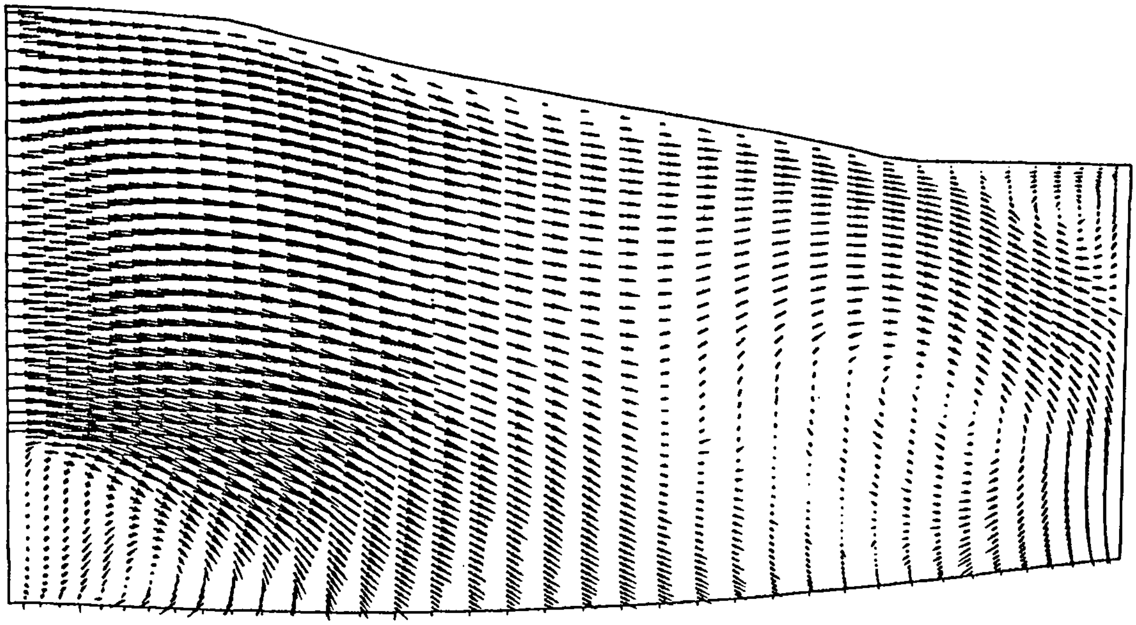


Fig. 22. Side view of velocity vectors in plane through center of inflow valve showing incoming fluid.

CHAPTER SEVEN

CONCLUSIONS

7.1 Summary

An algorithm for computing both steady-state and time-accurate solutions to the incompressible Navier-Stokes equations has been presented. This approach has been shown to be capable of computing flow about 3-D realistic geometries with moving boundaries. The method of artificial compressibility allows the equations to be solved as a hyperbolic system in pseudo-time. This requires the solution of a steady-state problem at each physical time step for the time-accurate formulation. The use of upwind differencing makes the system of numerical equations more diagonally dominant than a central-differencing scheme, leading to significantly faster convergence and far less computing time requirements. This is possible partially because of the use of a nonfactored implicit Gauss-Seidal line-relaxation scheme, making it possible to run the upwind-differencing scheme at very large time steps. Additional advantages of the current code include the smaller number of numerical parameters which must be specified. For this code, only the artificial compressibility β, and the number of line-relaxation sweeps need to be specified. These parameters are generally independent, that is, changing the value of one will not dramatically change the optimum value of the other. In contrast, an artificial compressibility code which uses central differencing and approximate factorization requires the specification of three dependent parameters: the artificial compressibility constant β, the artificial dissipation parameters, and the time step size.

The results showed good comparison with experimental results and with analytical solutions for the eight laminar flow test cases presented. Computing the flow through an artificial heart shows the capability of the code to simulate complicated internal flows with moving boundaries within a reasonable amount of computing time.

The choice of the artificial compressibility parameter β has been found to be very important. Its value can affect the convergence rate dramatically. The optimum values of β were found in the present work using numerical convergence tests. Two general dependencies were observed in this process: that the optimum β is inversely proportional to the Reynolds number of the flow problem; and that β is proportional to the grid density.

7.2 Future Work

Further advances in the convergence speed of the algorithm will still be very useful in increasing the usefulness of this code as a design tool. Several approaches could lead to some improvement in this area. The first and perhaps easiest to implement would be the use of a multigrid acceleration scheme. Some less obvious improvements could be sought after by studying the possibility of diagonalizing the left-hand side tridiagonal-system of equations which are solved for each separate line in the line-relaxation process. This could reduce the work of solving 4x4 blocks (in 3-D) to solving 4 independent scalar equations.

Another concern is the large memory resources required by the current implementation of the line-relaxation scheme in which all of the left-hand side terms are formed at once and stored. This implementation is made easier because the incompressible

Navier-Stokes have four dependent variables in 3-D, in contrast to the compressible equations which have five. This large memory usage makes this code useful for 3-D calculations on only a few of the supercomputers in use today. Some work will have to be done to test the amount of additional time required to compute the implicit-side terms as they are required instead of all at once. Testing of the artificial compressibility method using upwind differencing with several other types of implicit solvers would be very useful in further determining this algorithm's usefulness on supercomputers where memory is not as abundant. Some of these schemes could include approximate factorization and an LU factored scheme, such as the one by Yoon and Jameson [55].

APPENDIX A

VISCOUS FLUXES

This appendix presents the differential formulas for the viscous fluxes in generalized coordinates in both 2-D and 3-D for the following sets of conditions: nonconstant viscosity on a nonorthogonal mesh; nonconstant viscosity on an orthogonal mesh; constant viscosity on a nonorthogonal mesh; and constant viscosity on an orthogonal mesh. In the following equations, the velocity gradients in the viscous fluxes were written as

$$\frac{\partial u}{\partial \xi} = u_{\xi}, \text{ etc.}$$

and the metrics of the transformation have been represented with

$$\frac{\partial \xi}{\partial x} = \xi_{x}, \text{ etc.}$$

Three-Dimensional Formulation

Nonconstant Viscosity, Nonorthogonal Mesh

$$\hat{e}_v = \frac{\nu}{J}\begin{bmatrix}\begin{array}{c}
(2\xi_x^2 + \xi_y^2 + \xi_z^2)u_\xi + (2\xi_x\eta_x + \xi_y\eta_y + \xi_z\eta_z)u_\eta + (2\xi_x\zeta_x + \xi_y\zeta_y + \xi_z\zeta_z)u_\zeta \\
+\xi_y\xi_x v_\xi + \xi_y\eta_x v_\eta + \xi_y\zeta_x v_\zeta \\
+\xi_z\xi_x w_\xi + \xi_z\eta_x w_\eta + \xi_z\zeta_x w_\zeta \\
\\
(\xi_x^2 + 2\xi_y^2 + \xi_z^2)v_\xi + (\xi_x\eta_x + 2\xi_y\eta_y + \xi_z\eta_z)v_\eta + (\xi_x\zeta_x + 2\xi_y\zeta_y + \xi_z\zeta_z)v_\zeta \\
+\xi_x\xi_y u_\xi + \xi_x\eta_y u_\eta + \xi_x\zeta_y u_\zeta \\
+\xi_z\xi_y w_\xi + \xi_z\eta_y w_\eta + \xi_z\zeta_y w_\zeta \\
\\
(\xi_x^2 + \xi_y^2 + 2\xi_z^2)w_\xi + (\xi_x\eta_x + \xi_y\eta_y + 2\xi_z\eta_z)w_\eta + (\xi_x\zeta_x + \xi_y\zeta_y + 2\xi_z\zeta_z)w_\zeta \\
+\xi_x\xi_z u_\xi + \xi_x\eta_z u_\eta + \xi_x\zeta_z u_\zeta \\
+\xi_y\xi_z v_\xi + \xi_y\eta_z v_\eta + \xi_y\zeta_z v_\zeta
\end{array}\end{bmatrix}$$

$$\hat{f}_v = \frac{\nu}{J}\begin{bmatrix}\begin{array}{c}
(2\eta_x\xi_x + \eta_y\xi_y + \eta_z\xi_z)u_\xi + (2\eta_x^2 + \eta_y^2 + \eta_z^2)u_\eta + (2\eta_x\zeta_x + \eta_y\zeta_y + \eta_z\zeta_z)u_\zeta \\
+\eta_y\xi_x v_\xi + \eta_y\eta_x v_\eta + \eta_y\zeta_x v_\zeta \\
+\eta_z\xi_x w_\xi + \eta_z\eta_x w_\eta + \eta_z\zeta_x w_\zeta \\
\\
(\eta_x\xi_x + 2\eta_y\xi_y + \eta_z\xi_z)v_\xi + (\eta_x^2 + 2\eta_y^2 + \eta_z^2)v_\eta + (\eta_x\zeta_x + 2\eta_y\zeta_y + \eta_z\zeta_z)v_\zeta \\
+\eta_x\xi_y u_\xi + \eta_x\eta_y u_\eta + \eta_x\zeta_y u_\zeta \\
+\eta_z\xi_y w_\xi + \eta_z\eta_y w_\eta + \eta_z\zeta_y w_\zeta \\
\\
(\eta_x\xi_x + \eta_y\xi_y + 2\eta_z\xi_z)w_\xi + (\eta_x^2 + \eta_y^2 + 2\eta_z^2)w_\eta + (\eta_x\zeta_x + \eta_y\zeta_y + 2\eta_z\zeta_z)w_\zeta \\
+\eta_x\xi_z u_\xi + \eta_x\eta_z u_\eta + \eta_x\zeta_z u_\zeta \\
+\eta_y\xi_z v_\xi + \eta_y\eta_z v_\eta + \eta_y\zeta_z v_\zeta
\end{array}\end{bmatrix}$$

$$\hat{g}_v = \frac{\nu}{J}\begin{bmatrix}\begin{array}{c}
(2\zeta_x\xi_x + \zeta_y\xi_y + \zeta_z\xi_z)u_\xi + (2\zeta_x\eta_x + \zeta_y\eta_y + \zeta_z\eta_z)u_\eta + (2\zeta_x^2 + \zeta_y^2 + \zeta_z^2)u_\zeta \\
+\zeta_y\xi_x v_\xi + \zeta_y\eta_x v_\eta + \zeta_y\zeta_x v_\zeta \\
+\zeta_z\xi_x w_\xi + \zeta_z\eta_x w_\eta + \zeta_z\zeta_x w_\zeta \\
\\
(\zeta_x\xi_x + 2\zeta_y\xi_y + \zeta_z\xi_z)v_\xi + (\zeta_x\eta_x + 2\zeta_y\eta_y + \zeta_z\eta_z)v_\eta + (\zeta_x^2 + 2\zeta_y^2 + \zeta_z^2)v_\zeta \\
+\zeta_x\xi_y u_\xi + \zeta_x\eta_y u_\eta + \zeta_x\zeta_y u_\zeta \\
+\zeta_z\xi_y w_\xi + \zeta_z\eta_y w_\eta + \zeta_z\zeta_y w_\zeta \\
\\
(\zeta_x\xi_x + \zeta_y\xi_y + 2\zeta_z\xi_z)w_\xi + (\zeta_x\eta_x + \zeta_y\eta_y + 2\zeta_z\eta_z)w_\eta + (\zeta_x^2 + \zeta_y^2 + 2\zeta_z^2)w_\zeta \\
+\zeta_x\xi_z u_\xi + \zeta_x\eta_z u_\eta + \zeta_x\zeta_z u_\zeta \\
+\zeta_y\xi_z v_\xi + \zeta_y\eta_z v_\eta + \zeta_y\zeta_z v_\zeta
\end{array}\end{bmatrix} \tag{A.1}$$

Nonconstant Viscosity, Orthogonal Mesh

$$\hat{e}_v = \frac{\nu}{J}\begin{bmatrix} \xi_x^2 u_\xi + \xi_x \eta_x u_\eta + \xi_x \zeta_x u_\zeta \\ +\xi_y \xi_x v_\xi + \xi_y \eta_x v_\eta + \xi_y \zeta_x v_\zeta \\ +\xi_z \xi_x w_\xi + \xi_z \eta_x w_\eta + \xi_z \zeta_x w_\zeta \\ \\ \xi_x \xi_y u_\xi + \xi_x \eta_y u_\eta + \xi_x \zeta_y u_\zeta \\ +\xi_y^2 v_\xi + \xi_y \eta_y v_\eta + \xi_y \zeta_y v_\zeta \\ +\xi_z \xi_y w_\xi + \xi_z \eta_y w_\eta + \xi_z \zeta_y w_\zeta \\ \\ \xi_x \xi_z u_\xi + \xi_x \eta_z u_\eta + \xi_x \zeta_z u_\zeta \\ +\xi_y \xi_z v_\xi + \xi_y \eta_z v_\eta + \xi_y \zeta_z v_\zeta \\ +\xi_z^2 w_\xi + \xi_z \eta_z w_\eta + \xi_z \zeta_z w_\zeta \end{bmatrix}$$

$$\hat{f}_v = \frac{\nu}{J}\begin{bmatrix} \eta_x \xi_x u_\xi + \eta_x^2 u_\eta + \eta_x \zeta_x u_\zeta \\ +\eta_y \xi_x v_\xi + \eta_y \eta_x v_\eta + \eta_y \zeta_x v_\zeta \\ +\eta_z \xi_x w_\xi + \eta_z \eta_x w_\eta + \eta_z \zeta_x w_\zeta \\ \\ \eta_x \xi_y u_\xi + \eta_x \eta_y u_\eta + \eta_x \zeta_y u_\zeta \\ +\eta_y \xi_y v_\xi + \eta_y^2 v_\eta + \eta_y \zeta_y v_\zeta \\ +\eta_z \xi_y w_\xi + \eta_z \eta_y w_\eta + \eta_z \zeta_y w_\zeta \\ \\ \eta_x \xi_z u_\xi + \eta_x \eta_z u_\eta + \eta_x \zeta_z u_\zeta \\ +\eta_y \xi_z v_\xi + \eta_y \eta_z v_\eta + \eta_y \zeta_z v_\zeta \\ +\eta_z \xi_z w_\xi + \eta_z^2 w_\eta + \eta_z \zeta_z w_\zeta \end{bmatrix} \tag{A.2}$$

$$\hat{g}_v = \frac{\nu}{J}\begin{bmatrix} \zeta_x \xi_x u_\xi + \zeta_x \eta_x u_\eta + \zeta_x^2 u_\zeta \\ +\zeta_y \xi_x v_\xi + \zeta_y \eta_x v_\eta + \zeta_y \zeta_x v_\zeta \\ +\zeta_z \xi_x w_\xi + \zeta_z \eta_x w_\eta + \zeta_z \zeta_x w_\zeta \\ \\ \zeta_x \xi_y u_\xi + \zeta_x \eta_y u_\eta + \zeta_x \zeta_y u_\zeta \\ +\zeta_y \xi_y v_\xi + \zeta_y \eta_y v_\eta + \zeta_y^2 v_\zeta \\ +\zeta_z \xi_y w_\xi + \zeta_z \eta_y w_\eta + \zeta_z \zeta_y w_\zeta \\ \\ \zeta_x \xi_z u_\xi + \zeta_x \eta_z u_\eta + \zeta_x \zeta_z u_\zeta \\ +\zeta_y \xi_z v_\xi + \zeta_y \eta_z v_\eta + \zeta_y \zeta_z v_\zeta \\ +\zeta_z \xi_z w_\xi + \zeta_z \eta_z w_\eta + \zeta_z^2 w_\zeta \end{bmatrix}$$

Constant Viscosity, Nonorthogonal Mesh

$$
\begin{aligned}
\hat{e}_v &= \frac{\nu}{J}\begin{bmatrix} (\nabla\xi\cdot\nabla\xi)u_\xi + (\nabla\xi\cdot\nabla\eta)u_\eta + (\nabla\xi\cdot\nabla\zeta)u_\zeta \\ (\nabla\xi\cdot\nabla\xi)v_\xi + (\nabla\xi\cdot\nabla\eta)v_\eta + (\nabla\xi\cdot\nabla\zeta)v_\zeta \\ (\nabla\xi\cdot\nabla\xi)w_\xi + (\nabla\xi\cdot\nabla\eta)w_\eta + (\nabla\xi\cdot\nabla\zeta)w_\zeta \end{bmatrix} \\
\hat{f}_v &= \frac{\nu}{J}\begin{bmatrix} (\nabla\eta\cdot\nabla\xi)u_\xi + (\nabla\eta\cdot\nabla\eta)u_\eta + (\nabla\eta\cdot\nabla\zeta)u_\zeta \\ (\nabla\eta\cdot\nabla\xi)v_\xi + (\nabla\eta\cdot\nabla\eta)v_\eta + (\nabla\eta\cdot\nabla\zeta)v_\zeta \\ (\nabla\eta\cdot\nabla\xi)w_\xi + (\nabla\eta\cdot\nabla\eta)w_\eta + (\nabla\eta\cdot\nabla\zeta)w_\zeta \end{bmatrix} \\
\hat{g}_v &= \frac{\nu}{J}\begin{bmatrix} (\nabla\zeta\cdot\nabla\xi)u_\xi + (\nabla\zeta\cdot\nabla\eta)u_\eta + (\nabla\zeta\cdot\nabla\zeta)u_\zeta \\ (\nabla\zeta\cdot\nabla\xi)v_\xi + (\nabla\zeta\cdot\nabla\eta)v_\eta + (\nabla\zeta\cdot\nabla\zeta)v_\zeta \\ (\nabla\zeta\cdot\nabla\xi)w_\xi + (\nabla\zeta\cdot\nabla\eta)w_\eta + (\nabla\zeta\cdot\nabla\zeta)w_\zeta \end{bmatrix}
\end{aligned}
\tag{A.3}
$$

Constant Viscosity, Orthogonal Mesh

$$
\begin{aligned}
\hat{e}_v &= \frac{\nu}{J}\begin{bmatrix} (\nabla\xi\cdot\nabla\xi)u_\xi \\ (\nabla\xi\cdot\nabla\xi)v_\xi \\ (\nabla\xi\cdot\nabla\xi)w_\xi \end{bmatrix} \\
\hat{f}_v &= \frac{\nu}{J}\begin{bmatrix} (\nabla\eta\cdot\nabla\eta)u_\eta \\ (\nabla\eta\cdot\nabla\eta)v_\eta \\ (\nabla\eta\cdot\nabla\eta)w_\eta \end{bmatrix} \\
\hat{g}_v &= \frac{\nu}{J}\begin{bmatrix} (\nabla\zeta\cdot\nabla\zeta)u_\zeta \\ (\nabla\zeta\cdot\nabla\zeta)v_\zeta \\ (\nabla\zeta\cdot\nabla\zeta)w_\zeta \end{bmatrix}
\end{aligned}
\tag{A.4}
$$

Two-Dimensional Formulation

Nonconstant Viscosity, Nonorthogonal Mesh

$$
\begin{aligned}
\hat{e}_v &= \frac{\nu}{J}\begin{bmatrix} (2\xi_x^2 + \xi_y^2)u_\xi + (2\xi_x\eta_x + \xi_y\eta_y)u_\eta + \xi_x\xi_y v_\xi + \xi_y\eta_x v_\eta \\ (\xi_x^2 + 2\xi_y^2)v_\xi + (\xi_x\eta_x + 2\xi_y\eta_y)v_\eta + \xi_x\xi_y u_\xi + \xi_x\eta_y u_\eta \end{bmatrix} \\
\hat{f}_v &= \frac{\nu}{J}\begin{bmatrix} (2\eta_x^2 + \eta_y^2)u_\eta + (2\xi_x\eta_x + \xi_y\eta_y)u_\xi + \xi_x\eta_y v_\xi + \eta_x\eta_y v_\eta \\ (\eta_x^2 + 2\eta_y^2)v_\eta + (\xi_x\eta_x + 2\xi_y\eta_y)v_\xi + \xi_y\eta_x u_\xi + \eta_x\eta_y u_\eta \end{bmatrix}
\end{aligned}
\tag{A.5}
$$

Nonconstant Viscosity, Orthogonal Mesh

$$\begin{aligned}
\hat{e}_v &= \frac{\nu}{J}\begin{bmatrix} (2\xi_x^2+\xi_y^2)u_\xi+\xi_x\eta_x u_\eta+\xi_x\xi_y v_\xi+\xi_y\eta_x v_\eta \\ (\xi_x^2+2\xi_y^2)v_\xi+\xi_y\eta_y v_\eta+\xi_x\xi_y u_\xi+\xi_x\eta_y u_\eta \end{bmatrix} \\
\hat{f}_v &= \frac{\nu}{J}\begin{bmatrix} (2\eta_x^2+\eta_y^2)u_\eta+\xi_x\eta_x u_\xi+\xi_x\eta_y v_\xi+\eta_x\eta_y v_\eta \\ (\eta_x^2+2\eta_y^2)v_\eta+\xi_y\eta_y v_\xi+\xi_y\eta_x u_\xi+\eta_x\eta_y u_\eta \end{bmatrix}
\end{aligned} \tag{A.6}$$

Constant Viscosity, Nonorthogonal Mesh

$$\begin{aligned}
\hat{e}_v &= \frac{\nu}{J}\begin{bmatrix} (\xi_x^2+\xi_y^2)u_\xi+(\xi_x\eta_x+\xi_y\eta_y)u_\eta \\ (\xi_x^2+\xi_y^2)v_\xi+(\xi_x\eta_x+\xi_y\eta_y)v_\eta \end{bmatrix} \\
\hat{f}_v &= \frac{\nu}{J}\begin{bmatrix} (\eta_x^2+\eta_y^2)u_\eta+(\xi_x\eta_x+\xi_y\eta_y)u_\xi \\ (\eta_x^2+\eta_y^2)v_\eta+(\xi_x\eta_x+\xi_y\eta_y)v_\xi \end{bmatrix}
\end{aligned} \tag{A.7}$$

Constant Viscosity, Orthogonal Mesh

$$\begin{aligned}
\hat{e}_v &= \frac{\nu}{J}\begin{bmatrix} (\xi_x^2+\xi_y^2)u_\xi \\ (\xi_x^2+\xi_y^2)v_\xi \end{bmatrix} \\
\hat{f}_v &= \frac{\nu}{J}\begin{bmatrix} (\eta_x^2+\eta_y^2)u_\eta \\ (\eta_x^2+\eta_y^2)v_\eta \end{bmatrix}
\end{aligned} \tag{A.8}$$

APPENDIX B

EIGENSYSTEM OF THE CONVECTIVE FLUX JACOBIAN

3-D System of Equations

To form the delta fluxes used in the flux-difference splitting scheme, the eigensystem of the convective flux Jacobian is needed. For the equations in 3-D a generalized flux vector is given by

$$\hat{E}_i = \begin{bmatrix} \beta Q \\ k_t u + k_x p + uQ \\ k_t v + k_y p + vQ \\ k_t w + k_z p + wQ \end{bmatrix} \tag{B.1}$$

where $\hat{E}_i = \hat{E}, \hat{F}, \hat{G}$ for $i = 1, 2,$ and 3 respectively, and the normalized metrics are represented with

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

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

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

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

and the scaled contravariant velocity is

$$Q = k_x u + k_y v + k_z w$$

The Jacobian matrix $A_i = \frac{\partial \hat{E}_i}{\partial D}$ of the flux vector in Eq. (B.1) is given by

$$\hat{A}_i = \begin{bmatrix} 0 & k_x\beta & k_y\beta & k_z\beta \\ k_x & k_x u + Q + k_t & k_y u & k_z u \\ k_y & k_x v & k_y v + Q + k_t & k_z v \\ k_z & k_x w & k_y w & k_z w + Q + k_t \end{bmatrix} \tag{B.2}$$

A similarity transform for the Jacobian matrix is introduced

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

where

$$\Lambda_i = \mathrm{diag}[\lambda_1, \lambda_2, \lambda_3, \lambda_4]$$

$$\lambda_1 = Q + k_t$$

$$\lambda_2 = Q + k_t \qquad (B.3)$$

$$\lambda_3 = Q + k_t/2 + c$$

$$\lambda_4 = Q + k_t/2 - c$$

and where c is the scaled artificial speed of sound given by

$$c = \sqrt{(Q + k_t/2)^2 + \beta(k_x^2 + k_y^2 + k_z^2)} \qquad (B.4)$$

The matrix of the right eigenvectors is given by

$$X_i = \begin{bmatrix} 0 & 0 & \beta(c - k_t/2) & -\beta(c + k_t/2) \\ x_k & x_{kk} & u\lambda_3 + \beta k_x & u\lambda_4 + \beta k_x \\ y_k & y_{kk} & v\lambda_3 + \beta k_y & v\lambda_4 + \beta k_y \\ z_k & z_{kk} & w\lambda_3 + \beta k_z & w\lambda_4 + \beta k_z \end{bmatrix} \qquad (B.5)$$

and its inverse is given by

$$X_i^{-1} = \frac{1}{c^2 - k_t^2/4} \begin{bmatrix} x_{kk}(k_z v - k_y w) + y_{kk}(k_x w - k_z u) + z_{kk}(k_y u - k_x v) & y_{kk}(\lambda_1 w + \beta k_z) - z_{kk}(\lambda_1 v + \beta k_y) & z_{kk}(\lambda_1 u + \beta k_x) - x_{kk}(\lambda_1 w + \beta k_z) & x_{kk}(\lambda_1 v + \beta k_y) - y_{kk}(\lambda_1 u + \beta k_x) \\ x_k(k_y w - k_z v) + y_k(k_z u - k_x w) + z_k(k_x v - k_y u) & -y_k(\lambda_1 w + \beta k_z) + z_k(\lambda v + \beta k_y) & -z_k(\lambda_1 u + \beta k_x) + x_k(\lambda_1 w + \beta k_z) & -x_k(\lambda_1 v + \beta k_y) + y_k(\lambda_1 u + \beta k_x) \\ -\lambda_4(c + k_t/2)/(2\beta c) & k_x(c + k_t/2)/(2c) & k_y(c + k_t/2)/(2c) & k_z(c + k_t/2)/(2c) \\ -\lambda_3(c - k_t/2)/(2\beta c) & k_x(c - k_t/2)/(2c) & k_y(c - k_t/2)/(2c) & k_z(c - k_t/2)/(2c) \end{bmatrix} \qquad (B.6)$$

where

$$x_k = \frac{\partial x}{\partial \xi_{i+1}}$$
$$y_k = \frac{\partial y}{\partial \xi_{i+1}}$$
$$z_k = \frac{\partial z}{\partial \xi_{i+1}}$$
$$x_{kk} = \frac{\partial x}{\partial \xi_{i+2}}$$
$$y_{kk} = \frac{\partial y}{\partial \xi_{i+2}}$$
$$z_{kk} = \frac{\partial z}{\partial \xi_{i+2}}$$

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

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

2-D System of Equations

A generalized flux vector is given by

$$\hat{E}_i = \begin{bmatrix} \beta Q \\ k_x p + uQ + k_t u \\ k_y p + vQ + k_t v \end{bmatrix} \qquad (B.7)$$

where $\hat{E}_i = \hat{E}, \hat{F}$ for $i =$ 1,2 respectively, and the normalized metrics are represented with

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

and the scaled contravariant velocity is

$$Q = k_x u + k_y v$$

The Jacobian matrices for this system are given by

$$\hat{A}_i = \frac{\partial \hat{E}_i}{\partial D} = \begin{bmatrix} 0 & \beta k_x & \beta k_y \\ k_x & k_x u + Q + k_t & k_y u \\ k_y & k_x v & k_y v + Q + k_t \end{bmatrix} \qquad (B.8)$$

A similarity transform for the Jacobian matrix is introduced

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

where

$$\begin{aligned}
\Lambda_i &= \mathrm{diag}[\lambda_1, \lambda_2, \lambda_3] \\
\lambda_1 &= Q + k_t \\
\lambda_2 &= Q + c + \frac{1}{2}k_t \\
\lambda_3 &= Q - c + \frac{1}{2}k_t
\end{aligned} \tag{B.9}$$

and where c is the scaled artificial speed of sound given by

$$c = \sqrt{(Q + \frac{1}{2}k_t)^2 + \beta(k_x^2 + k_y^2)} \tag{B.10}$$

The matrix of the right eigenvectors is given by

$$X_i = \frac{1}{2\beta c(c^2 - \frac{1}{4}k_t^2)} \begin{bmatrix} 0 & \beta(c^2 - \frac{1}{4}k_t^2) & -\beta(c^2 - \frac{1}{4}k_t^2) \\ -2\beta c k_y & (u\lambda_2 + \beta k_x)(c + \frac{1}{2}k_t) & (u\lambda_3 + \beta k_x)(c - \frac{1}{2}k_t) \\ 2\beta c k_x & (v\lambda_2 + \beta k_y)(c + \frac{1}{2}k_t) & (v\lambda_3 + \beta k_y)(c - \frac{1}{2}k_t) \end{bmatrix} \tag{B.11}$$

and its inverse is given by

$$X_i^{-1} = \begin{bmatrix} k_y u - k_x v & -v\lambda_1 - \beta k_y & u\lambda_1 + \beta k_x \\ -\lambda_3 & \beta k_x & \beta k_y \\ -\lambda_2 & \beta k_x & \beta k_y \end{bmatrix} \tag{B.12}$$

REFERENCES

1. Fasel, M.: Investigation of the Stability of Boundary Layers by a Finite-Difference Model of the Navier-Stokes Equations. Journal of Fluid Mechanics, vol. 78, pt. 2, 1976, pp. 355–383.
2. Aziz K.; and Hellums, J. D.: Numerical Solution of the Three-Dimensional Equations of Motion of Laminar Natural Convection. Physics of Fluids, vol. 10, no. 2, 1967, pp. 314–324.
3. Hafez, M.; Dacles, J.; and Soliman, M.: A Velocity/Vorticity Method For Viscous Incompressible Flow Calculations. To appear in "Proceedings of the 11th ICNMFD," Springer-Verlag, 1988.
4. Harlow, F. H.; and Welch, J. E.: Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface. Physics of Fluids, vol. 8, 1965, pp. 2182–2189.
5. Chorin, A. J.: Numerical Solution of the Navier-Stokes Equations. Mathematics of Computation, vol. 22, no. 104, 1968, pp. 745–762.
6. Kim, J.; and Moin, P.: Application of a Fractional-Step Method to Incompressible Navier-Stokes Equations. Journal of Computational Physics, vol. 59, 1985, pp. 308–323.
7. Rosenfeld, M.; Kwak, D.; and Vinokur, M.: A Solution Method for the Unsteady Incompressible Navier-Stokes Equations in Generalized Coordinate Systems. AIAA Paper 88-0718, 1988.

8. Chorin, A. J.: A Numerical Method for Solving Incompressible Viscous Flow Problems. Journal of Computational Physics, vol. 2, 1967, pp. 12–26.
9. Kwak, D.; Chang, J. L. C.; Shanks, S. P.; and Chakravarthy, S. R.: A Three-Dimensional Incompressible Navier-Stokes Flow Solver Using Primitive Variables. AIAA Journal, vol. 24, no. 3, 1986, pp. 390–396.
10. Merkle, C. L.; and Athavale, M.: Time-Accurate Unsteady Incompressible Flow Algorithms Based on Artificial Compressibility, AIAA Paper 87-1137, 1987.
11. Rogers, S. E.; Kwak, D.; and Kaul, U.: On the Accuracy of the Pseudocompressibility Method in solving the Incompressible Navier-Stokes Equations. Applied Mathematical Modelling, vol. 11, 1987, pp. 35–44.
12. Roe, P. L.: Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes. Journal of Computational Physics, vol. 43, pp. 357–372, 1981.
13. Chakravarthy, S. R.; and Osher, S.: A New Class of High Accuracy TVD Schemes For Hyperbolic Conservation Laws. AIAA Paper 85-0363, 1985.
14. Steger, J. L.; and Warming, R. F.: Flux Vector Splitting of the Inviscid Gasdynamic Equations With Application to Finite Difference Methods. NASA TM 78605, 1979.
15. 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.
16. Hartwich, P. M.; and Hsu, C. H.: High Resolution Upwind Schemes for the Three-Dimensional Incompressible Navier-Stokes Equations. AIAA Paper 87-0547, 1987.
17. Hsu, C. H.; and Hartwich, P. M.: Incompressible Navier-Stokes Solutions for a Sharp-Edged Double-Delta Wing. AIAA Paper 87-0206, 1987.

18. Beam, R. M.; and Warming, R. F.: An Implicit Finite-Difference Algorithm for Hyperbolic Systems in Conservation-Law Form. Journal of Computational Physics, vol. 22, 1976, pp. 87–110.

19. Briley, W. R.; and McDonald, H.: Solution of the Three-Dimensional Compressible Navier-Stokes Equations by an Implicit Technique. Proceedings of the Fourth International Conference on Numerical Methods in Fluid Dynamics, Lecture Notes in Physics, vol. 35, Springer-Verlag, New York, 1975, pp. 105–110.

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

21. Pulliam, T. H.; and Steger, J. L.: On Implicit Finite-Difference Simulations of Three-Dimensional Flow. AIAA Paper 78-10, 1978.

22. Thomas, P. D.; and Lombard, C. K.: Geometric Conservation Law and Its Application to Flow Computations on Moving Grids. AIAA Journal, vol. 17, no. 10, 1979, pp. 1030-1037.

23. Buning, P. G.; and Steger, J. L.: Solution of the Two-Dimensional Euler Equations with Generalized Coordinate Transformation Using Flux Vector Splitting. AIAA Paper 82-0971, 1982.

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

25. Pulliam, T. H.: Artificial Dissipation Models for the Euler Equations. AIAA Journal, vol. 24, no. 12, 1986, pp. 1931–1940.

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

27. Yee, H. C.: Linearized Form of Implicit TVD Schemes For The Multidimensional Euler and Navier-Stokes Equations. Computers and Mathematics with Applications, vol. 12A, nos. 4/5, 1986, pp. 413–432.

28. Chakravarthy, S. R.: Relaxation Methods for Unfactored Implicit Upwind Schemes. AIAA Paper 84-0165, 1984.

29. Merkle, C. L.; and Tsai, P. Y. L.: Application of Runge-Kutta Schemes to Incompressible Flows. AIAA Paper 86-0553, 1986.

30. Ghia, U.; Ghia, K. N.; and Shin, C. T.: High-Re Solutions for Incompressible Flow Using the Navier-Stokes Equations and a Multigrid Method. Journal of Computational Physics, vol. 48, 1982, pp. 387–411.

31. Schreiber, R.; and Keller, H. B.: Driven Cavity Flows by Efficient Numerical Techniques. Journal of Computational Physics, vol. 49, 1983, pp. 310–333.

32. Vanka, S. P.: Block-Implicit Multigrid Solution of Navier-Stokes Equations in Primitive Variables. Journal of Computational Physics, vol. 65, 1986, pp. 138–158.

33. Benjamin, A. S.; and Denny, V. E.: On the Convergence of Numerical Solutions for 2-D Flows in a Cavity at Large Re. Journal of Computational Physics, vol. 33, 1979, pp. 340–358.

34. Armaly, B. F.; Durst, F.; Periera, J. C. F.; and Schönung, B.: Experimental and Theoretical Investigation of Backward-Facing Step Flow. Journal of Fluid Mechanics, vol. 127, 1983, pp. 473–496.

35. Takami, H.; and Keller, H. B.: Steady Two-Dimensional Viscous Flow of an Incompressible Fluid Past a Circular Cylinder, Physics of Fluids Supplement II, vol. 12, 1969, pp. II-51–II-56.

36. Dennis, S. C. R.; and Chang, G. Z.: Numerical Solutions for Steady Flow Past a Circular Cylinder at Reynolds Numbers up to 100. Journal of Fluid Mechanics, vol. 42, 1970, pp. 471–489.

37. Tuann, S. Y.; and Olson, M. D.: Numerical Studies of the Flow Around a Circular Cylinder by a Finite Element Method. Computers and Fluids, vol. 6, 1978, pp. 219–240.

38. 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. Journal of Fluid Mechanics, vol. 165, 1986, pp. 79–130.

39. 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 1. Steady Flow. Journal of Fluid Mechanics, vol. 79, no. 2, 1977, pp. 231–256.

40. Tritton, D. J.: Experiments on the Flow Past a Circular Cylinder at Low Reynolds Numbers. Journal of Fluid Mechanics, vol. 6, 1959, pp. 547–567.

41. White, F. M.: Viscous Fluid Flow, McGraw-Hill, New York, 1974, pp. 148–149.

42. 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.

43. Lecointe, Y.; and Piquet, J.: On the Use of Several Compact Methods for the Study of Unsteady Incompressible Viscous Flow Round a Circular Cylinder. Computers and Fluids, vol. 12, no. 4, 1984, pp. 255–280.

44. Martinez, G.: Caractéristiques Dynamiques et Thermiques de l'Écoulements Autour d'un Cylindre Circulaire à Nombres de Reynolds Modérés. Thèse D. I. – I. N. P. Toulouse, 1979.

45. Lin, C. L.; Pepper, D. W.; and Lee, S. C.: Numerical Methods for Separated Flow Solutions Around a Circular Cylinder. AIAA Journal, vol. 14, 1976, pp. 900–907.

46. Thoman, D.; and Szewczyk, A.: Time Dependent Viscous Flow Over a Circular Cylinder. Physics of Fluids, Supplement II, 1969, pp. 79–86.

47. Wille, R.: Karman Vortex Streets. Advances in Applied Mechanics, vol. 6, 1960, pp. 273.

48. Kovasznay, L. S. G.: Hot-wire Investigation of the Wake Behind Cylinders at Low Reynolds Numbers. Proceedings of the Royal Society A, vol. 198, 1949, pp. 174–190.

49. Roshko, A.: On the Development of Turbulent Wakes From Vortex Streets. NACA Rep. 1191, 1954.

50. Van Dyke, M.: An Album of Fluid Motion, The Parabolic Press, Stanford, CA, 1982, p. 57.

51. Humphrey, J. A. C.; Taylor, A. M. K.; and Whitelaw, J. H.: Laminar Flow in a Square Duct of Strong Curvature. Journal of Fluid Mechanics, vol. 83, pt. 3, 1977, pp. 509–527.

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

53. 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. Journal of Biomechanical Engineering, Trans. ASME, vol. 108, 1986, pp. 232–238.

54. Bjorstad, P. E.: Numerical Solution of the Biharmonic Equation. Ph.D. Dissertation, Stanford University, 1980.

55. Yoon, S.; and Jameson, A.: An LU-SSOR Scheme for the Euler and Navier-Stokes Equations. AIAA Paper 87-0600.

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.
NASA TM-102199

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
Numerical Solution of the Incompressible Navier-Stokes Equations

5. Report Date
November 1990

6. Performing Organization Code

7. Author(s)
Stuart E. Rogers

8. Performing Organization Report No.
A-89159

10. Work Unit No.
505-60

9. Performing Organization Name and Address
Ames Research Center
Moffett Field, CA 94035-1000

11. Contract or Grant No.

13. Type of Report and Period Covered
Technical Memorandum

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

14. Sponsoring Agency Code

15. Supplementary Notes
Point of Contact: Stuart E. Rogers, Ames Research Center, MS 258-1
Moffett Field, CA 94035-1000
(415) 604-4481 or FTS 464-4481
Published as doctoral dissertation, Stanford University, Palo Alto, CA, March 1989.

16. Abstract

The current work is initiated in an effort to obtain an efficient, accurate, and robust algorithm for the numerical solution of the incompressible Navier-Stokes equations in two- and three-dimensional generalized curvilinear coordinates for both steady-state and time-dependent flow problems. This is accomplished with the use of the method of artificial compressibility and a high-order flux-difference splitting technique for the differencing of the convective terms. Time accuracy is obtained in the numerical solutions by subiterating the equations in pseudo-time for each physical time step. The system of equations is solved with a line-relaxation scheme which allows the use of very large pseudo-time steps leading to fast convergence for steady-state problems as well as for the subiterations of time-dependent problems. Numerous laminar test flow problems are computed and presented with a comparison against analytically known solutions or experimental results. These include the flow in a driven cavity, the flow over a backward-facing step, the steady and unsteady flow over a circular cylinder, flow over an oscillating plate, flow through a one-dimensional inviscid channel with oscillating back pressure, the steady-state flow through a square duct with a 90° bend, and the flow through an artificial heart configuration with moving boundaries. An adequate comparison with the analytical or experimental results is obtained in all cases. Numerical comparisons of the upwind differencing with central differencing plus artificial dissipation indicates that the upwind differencing provides a much more robust algorithm, which requires significantly less computing time. The time-dependent problems require on the order of 10 to 20 subiterations, indicating that the elliptical nature of the problem does require a substantial amount of computing effort.

17. Key Words (Suggested by Author(s))
Incompressible Navier-Stokes
Artificial compressibility
Upwind differencing
Artificial heart

18. Distribution Statement
Unclassified-Unlimited

Subject Category – 64

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	104	A06

NASA FORM 1626 OCT 86

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

DO NOT REMOVE SLIP FROM MATERIAL

Delete your name from this slip when returning material to the library.

NAME	DATE	MS
~~Meelan Choudhari~~	~~6/97~~	~~922~~

NASA Langley (Rev. Dec. 1991) RIAD N-75