An Implicit Numerical Scheme for the Simulation of Internal Viscous Flows on Unstructured Grids

Philip C.E. Jorgenson
Lewis Research Center
Cleveland, Ohio

and

Richard H. Pletcher
Institute for Computational Mechanics in Propulsion
Lewis Research Center
Cleveland, Ohio

and Iowa State University
Ames, Iowa

NASA

ICOMP
Lewis Research Center
Ohio Aerospace Institute

An Implicit Numerical Scheme for the Simulation of Internal Viscous Flows on Unstructured Grids

Philip C.E. Jorgenson*
Lewis Research Center
Cleveland, Ohio 44135
and
Richard H. Pletcher†
Institute for Computational Mechanics in Propulsion
Lewis Research Center
Cleveland, Ohio 44135

and Mechanical Engineering Department
Iowa State University
Ames, Iowa 50010

Abstract

The Navier-Stokes equations are solved numerically for two-dimensional steady viscous laminar flows. The grids are generated based on the method of Delaunay triangulation.

A finite-volume approach is used to discretize the conservation law form of the compressible flow equations written in terms of primitive variables. A preconditioning matrix is added to the equations so that low Mach number flows can be solved economically. The equations are time marched using either an implicit Gauss-Seidel iterative procedure or a solver based on a conjugate gradient like method. A four color scheme is employed to vectorize the block Gauss-Seidel relaxation procedure. This increases the memory requirements minimally and decreases the computer time spent solving the resulting system of equations substantially. A factor of 7.6 speedup in the matrix solver is typical for the viscous equations.

Numerical results are obtained for inviscid flow over a bump in a channel at subsonic and transonic conditions for validation with structured solvers. Viscous results are computed for developing flow in a channel, a symmetric sudden expansion, periodic tandem cylinders in a cross-flow, and a four-port valve. Comparisons are made with available results obtained by other investigators.

1 Introduction

The development of a general computer code that can predict the flow about complex geometries which include complex flow structures is desirable in computational fluid dynamics. Many numerical schemes proposed to date which use the finite difference or finite volume formulation of the flow equations were written to take advantage of some inherent grid structure which then permits flow solutions to be obtained efficiently [1] - [3]. A structured mesh can be defined as a domain that is discretized such that the neighborhood of a cell or node can be related to its own index number. This structure, which makes the solver so efficient, often makes it difficult to obtain reasonable grids about complex flow geometries. Many of these solution algorithms can be found discussed in review papers [4], [5].

An unstructured grid flow solver can alleviate many of the problems associated with structured grids. However, unlike the structured grid, the cell neighborhood of an unstructured grid must be defined explicitly. The triangle is the simplest and most convenient geometric figure that can be used to cover a two-dimensional domain. An advantage of using a simple triangular shaped cell is the ability to generate

grids about arbitrary geometries. Another advantage is the ability to add cells in high gradient regions of the flow field as well as those regions of the flow that are of interest without concern for the surrounding cells. The main disadvantages of using an unstructured mesh lies in the added complexity and memory requirements of the flow solver.

Several researchers have recently reported successful results in solving the Euler and Navier-Stokes equations on unstructured grids [6] - [11]. The research reported here considered the use of unstructured grids in predicting low Mach number flows through internal geometries. To date, there has not been much work done toward applying unstructured grids to viscous internal flows at low Mach numbers.

In the research to be described here, the Euler and Navier-Stokes equations were discretized on unstructured grids composed of triangles in finite volume form using primitive variables; however, the conservation law form was retained. Preconditioning of the time derivative term was used to allow efficient calculations at vanishingly small Mach numbers. The equations were solved iteratively using either the implicit Gauss-Seidel method or an iterative conjugate gradient based solver. Several iterative conjugate gradient based solvers and matrix preconditioners were considered.

Details of the discretization, the preconditioning, the grid generation strategy, and the methods for solving the algebraic equations are given in sections that follow. Results are given for several test cases including the inviscid flow over a bump in a channel at subsonic and transonic conditions, viscous developing flow in a channel at several Reynolds numbers, a symmetric rearward-facing step flow, flow over a cascade of cylinders arranged in tandem, and flow through a four-port valve.

2 Governing Equations

The Navier-Stokes equations were used to model viscous fluid flow problems in this study. In conservation law form and physical coordinates these equations can be written in vector form as

$$\frac{\partial U}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} = 0 \qquad (1)$$

where the vectors

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ e \end{bmatrix},$$

$$E = \begin{bmatrix} \rho u \\ \rho u^2 + P - \tau_{xx} \\ \rho u v - \tau_{xy} \\ (P+e)u - u\tau_{xx} - v\tau_{xy} + q_x \end{bmatrix},$$

and

$$F = \begin{bmatrix} \rho v \\ \rho u v - \tau_{xy} \\ \rho v^2 + P - \tau_{yy} \\ (P+e)v - u\tau_{xy} - v\tau_{yy} + q_y \end{bmatrix}.$$

In this work only Newtonian fluids will be considered, so the shear stress tensors are defined as

$$\begin{aligned} \tau_{xx} &= -\tfrac{2}{3}\mu(u_x + v_y) + 2\mu u_x = \tfrac{2}{3}\mu(2u_x - v_y), \\ \tau_{xy} &= \mu(v_x + u_y), \\ \tau_{yy} &= -\tfrac{2}{3}\mu(u_x + v_y) + 2\mu v_x = \tfrac{2}{3}\mu(2v_y - u_x), \end{aligned} \qquad (2)$$

and

$$\begin{aligned} q_x &= -\kappa T_x, \\ q_y &= -\kappa T_y. \end{aligned} \qquad (3)$$

If a further assumption is made that the gas is ideal, where $\rho = P/RT$ with R being the gas constant per unit mass, the Navier-Stokes equations can be written in nondimensional form as

$$\frac{\partial Q(\tilde{w})}{\partial \tilde{t}} + \frac{\partial G(\tilde{w})}{\partial \tilde{x}} + \frac{\partial H(\tilde{w})}{\partial \tilde{y}} = 0 \qquad (4)$$

with

$$\tilde{w} = \begin{bmatrix} \tilde{P} \\ \tilde{u} \\ \tilde{v} \\ \tilde{T} \end{bmatrix},$$

$$Q = \begin{bmatrix} \frac{\tilde{P}}{\tilde{T}} \\ \frac{\tilde{P}\tilde{u}}{\tilde{T}} \\ \frac{\tilde{P}\tilde{v}}{\tilde{T}} \\ \frac{\tilde{P}}{\tilde{T}}[(\tilde{C}_p - \tilde{R})\tilde{T} + \frac{\tilde{u}^2}{2} + \frac{\tilde{v}^2}{2}] \end{bmatrix},$$

$$G = \begin{bmatrix} \frac{\tilde{P}\tilde{u}}{\tilde{T}} \\ \frac{\tilde{P}\tilde{u}^2}{\tilde{T}} + \tilde{R}\tilde{P} - \tilde{\tau}_{xx} \\ \frac{\tilde{P}\tilde{u}\tilde{v}}{\tilde{T}} - \tilde{\tau}_{xy} \\ \frac{\tilde{P}}{\tilde{T}}(\tilde{C}_p\tilde{T} + \frac{\tilde{u}^2}{2} + \frac{\tilde{v}^2}{2})\tilde{u} - \tilde{u}\tilde{\tau}_{xx} - \tilde{v}\tilde{\tau}_{xy} - C\tilde{T}_x \end{bmatrix},$$

and

$$H = \begin{bmatrix} \frac{\tilde{P}\tilde{v}}{\tilde{T}} \\ \frac{\tilde{P}\tilde{u}\tilde{v}}{\tilde{T}} - \tilde{\tau}_{xy} \\ \frac{\tilde{P}\tilde{v}^2}{\tilde{T}} + \tilde{R}\tilde{P} - \tilde{\tau}_{yy} \\ \frac{\tilde{P}}{\tilde{T}}(\tilde{C}_p\tilde{T} + \frac{\tilde{u}^2}{2} + \frac{\tilde{v}^2}{2})\tilde{v} - \tilde{u}\tilde{\tau}_{xy} - \tilde{v}\tilde{\tau}_{yy} - C\tilde{T}_y \end{bmatrix},$$

where

$$\tilde{\tau}_{xx} = \tfrac{2}{3}\tfrac{\tilde{R}\tilde{\mu}}{Re}(2\tilde{u}_x - \tilde{v}_y),$$
$$\tilde{\tau}_{xy} = \tfrac{\tilde{R}\tilde{\mu}}{Re}(\tilde{v}_x + \tilde{u}_y),$$
$$\tilde{\tau}_{yy} = \tfrac{2}{3}\tfrac{\tilde{R}\tilde{\mu}}{Re}(2\tilde{v}_y - \tilde{u}_x), \qquad (5)$$
$$\mathcal{C} = \tfrac{\tilde{C}_p\tilde{\mu}\tilde{R}}{Pr\cdot Re}.$$

The Reynolds and Prandtl numbers are defined as

$$Re = \frac{\rho_{ref}u_{ref}L_{ref}}{\mu_{ref}}, \qquad Pr = \frac{C_{p_{ref}}\mu_{ref}}{\kappa_{ref}}$$

respectively. The $\tilde{}$ refers to a term that is nondimensional. The variables subscripted ref are reference quantities specific to a particular flow calculation. The test cases presented in this work involve the laminar flow of air where the viscosity is assumed to follow the Sutherland formula. It is important to note that although these equations are written in terms of primitive variables(P, u, v, T), they are still in conservation law form.

All subsequent equations are nondimensional so the $\tilde{}$ is dropped for convenience.

3 Preconditioning

Solving the compressible flow equations for low Mach number cases is difficult because the resulting system is stiff due to the large ratio of acoustic to convective velocities. A temporal preconditioning is used in this work to remove this stiffness.

The approach will be demonstrated by using the Navier-Stokes equations in one dimension. The nondimensional form of these equations is written as

$$\frac{\partial Q(w)}{\partial t} + \frac{\partial G(w)}{\partial x} - \frac{\partial G_v(w)}{\partial x} = 0, \qquad (6)$$

where

$$w = \begin{bmatrix} P \\ u \\ T \end{bmatrix},$$

$$Q = \begin{bmatrix} \frac{P}{RT} \\ \frac{Pu}{RT} \\ \frac{P}{\gamma R} + \frac{M^2\Gamma}{2}\frac{Pu^2}{RT} \end{bmatrix},$$

$$G = \begin{bmatrix} \frac{Pu}{RT} \\ \frac{Pu^2}{RT} + P \\ \frac{Pu}{R} + \frac{M^2\Gamma}{2}\frac{Pu^3}{RT} \end{bmatrix},$$

$$G_v = \begin{bmatrix} 0 \\ \frac{4\mu}{3Re}u_x \\ \frac{4M^2\Gamma\mu}{3Re}uu_x - \frac{\mu}{PrRe}T_x \end{bmatrix}.$$

The quantity M is the reference Mach number, defined as $u_{ref}/\sqrt{\gamma R T_{ref}}$, which appears when the equations are cast in nondimensional form. Here and in the matrices that follow Γ is used for convenience to represents the quantity $\gamma - 1$. This equation can be rewritten as

$$\mathbf{A}_t\frac{\partial w}{\partial t} + \mathbf{A}_x\frac{\partial w}{\partial x} = \frac{\partial G_v(w)}{\partial x}, \qquad (7)$$

where the Jacobian matrices, $\mathbf{A}_t$ and $\mathbf{A}_x$ can be written as

$$\mathbf{A}_t = \begin{bmatrix} \frac{\gamma M^2}{T} & 0 & -\frac{P}{RT^2} \\[2mm] \frac{\gamma M^2 u}{T} & \frac{P}{RT} & -\frac{Pu}{RT^2} \\[2mm] M^2 + \frac{\gamma M^4\Gamma u^2}{2T} & \frac{\gamma M^4\Gamma Pu}{T} & -\frac{\gamma M^4\Gamma Pu^2}{2T^2} \end{bmatrix},$$

and

$$\mathbf{A}_x =$$

$$\begin{bmatrix} \frac{\gamma M^2 u}{T} & \frac{P}{RT} & -\frac{Pu}{RT^2} \\[2mm] \frac{\gamma M^2 u^2}{T} + 1 & \frac{2Pu}{RT} & -\frac{Pu^2}{RT^2} \\[2mm] \gamma M^2 u + \frac{\gamma M^4\Gamma u^3}{2T} & \frac{P}{R} + \frac{3M^4\Gamma Pu^2}{2T} & -\frac{\gamma M^4\Gamma Pu^3}{2T^2} \end{bmatrix}.$$

It is important to note that the quantity P/RT is well behaved as Mach number goes to zero since it is equal to the nondimensional density, ρ/ρ_{ref}. Also note that $R = (\gamma M^2)^{-1}$.

The first column of the Jacobian matrices contains the coefficients associated with pressure. In $\mathbf{A}_t$, these coefficients go to zero as the Mach number goes to zero. This results in the acoustic time scale restriction associated with pressure. As the Mach number approaches zero, a vanishingly small time step is needed to keep the coefficients multiplying pressure finite. More importantly, the system of equations become singular as the Mach number goes to zero, as will be discussed below. For a finite time step, the time derivative of pressure will vanish from the equations. In the limit as the Mach number approaches zero, the equations reduce to their incompressible form. Since the time derivative of pressure does not appear in this form of the incompressible equations, the equations contain no pressure history. There is a more mathematical way of looking at this problem. The system can be rewritten in the form

$$\frac{\partial w}{\partial t} + \mathbf{A}_t^{-1}\mathbf{A}_x\frac{\partial w}{\partial x} = \mathbf{A}_t^{-1}\frac{\partial G_v(w)}{\partial x}. \qquad (8)$$

3

As Mach number, M, becomes small, $\mathbf{A}_t$ becomes ill-conditioned, i.e., the determinant of $\mathbf{A}_t$ becomes small and errors due to round off error will become large when computing $\mathbf{A}_t^{-1}$. In the limit as M goes to zero, $\mathbf{A}_t^{-1}$ is unbounded and the system is singular. The result is often slow convergence due to this stiffness when using a compressible code to compute a flow at very low Mach number. In addition, the eigenvalues($U+C$, $U-C$, and U)of the Jacobian matrix become farther apart as the Mach number goes to zero. A remedy for this is write the equations in the form

$$\mathbf{A}_p \frac{\partial w}{\partial \tau} + \mathbf{A}_t \frac{\partial w}{\partial t} + \mathbf{A}_x \frac{\partial w}{\partial x} = \frac{\partial G(w)}{\partial x}. \qquad (9)$$

The preconditioning Jacobian matrix, $\mathbf{A}_p$, is defined as

$$\mathbf{A}_p = \begin{bmatrix} \frac{1}{T} & 0 & -\frac{P}{RT^2} \\ \frac{u}{T} & \frac{P}{RT} & -\frac{Pu}{RT^2} \\ \frac{1}{\gamma} + \frac{M^2\Gamma u^2}{2T} & \frac{\gamma M^4 \Gamma Pu}{T} & -\frac{\gamma M^4 \Gamma Pu^2}{2T^2} \end{bmatrix}.$$

This Jacobian matrix used in the preconditioning is of the same form as the matrix $\mathbf{A}_t$, but the dependence of Mach number is removed from the terms that are causing the ill-conditioning. This essentially attempts to cluster the eigenvalues around the convective speed. It may be possible to simplify $\mathbf{A}_p$ by setting some of the nondiagonal terms in columns two and three equal to zero.

For a time dependent calculation at low Mach number, the preconditioned equations are advanced in the pseudo time frame as well as the real time frame. At each physical time step, the equations are iterated to convergence in pseudo time, τ. At this point the pseudo time term vanishes and the time dependent Navier-Stokes equations are satisfied. The pseudo time iterations also remove the linearization error from the solution at each physical time level. If the low Mach number flow computation is steady, it is only necessary to integrate the equations in the pseudo time frame. This is done by setting the physical time step to a very large number to remove the effect of the physical time derivative from the preconditioned equations.

At higher Mach numbers the pseudo time term is not needed, although convergence does not appear to deteriorate with its continued use.

Pseudo time terms for the full two-dimensional equations are added to the diagonal blocks of the sparse matrix, $\mathbf{A}$, in Eq. (18). These terms are formed by a direct extension of the one-dimensional example above.

4 Grid Generation

The triangular shaped cell is the simplest geometric shape that can be used to cover a two-dimensional computational domain. With an unstructured grid, individual cells can no longer refer to their neighbors simply by incrementing an index as in a structured grid. Instead, the neighborhood of a cell is determined through a connectivity matrix. This connectivity matrix usually contains cell based information as well as edge based information. Details of the connectivity matrix required by the computer flow code developed in this work will be discussed later in this section.

The use of a triangular unstructured grid formulation has some distinct advantages over a structured grid. One advantage to using triangular cells is that with them it is easy to generate grids about complex geometries. This reduces the amount of time required to generate a suitable grid. Also grid adaptation can be done locally without adding unnecessary cells to other regions of the domain.

The method used in this work follows a path similar to that of Holmes and Snyder [12] to triangulate a region. First, the boundaries that describe the computational domain are defined. Next, the boundaries are discretized in a counter-clockwise direction. These discrete points are then triangulated using the Delaunay criterion that no other point in the computational domain lies within the circumcircle of a given triangular cell. Points must now be added to the domain to obtain a reasonable grid. A new point can be added based on any criteria one chooses. The grid point insertion in the current work is done according to one of the following three geometric criteria: improve the triangle with the smallest aspect ratio, reduce the maximum area triangle, reduce the size of the triangle with the largest circumcircle radius. These geometric constraints can be used in any combination. Some other criteria that could be used for local retriangulation are increase minimum angle, decrease maximum angle, and maintain equal length sides, to name a few. An example of a coarse grid about a square hole generated in the above manner is shown in Fig. 1.

The flow code requirements dictate the type of output that the grid generation scheme must provide. A connectivity array must be generated for an unstructured grid so that a cell neighborhood is completely defined for the flow code. The code in the current work was based on a cell centered scheme. Here the triangle itself was the control volume used in the finite volume formulation.

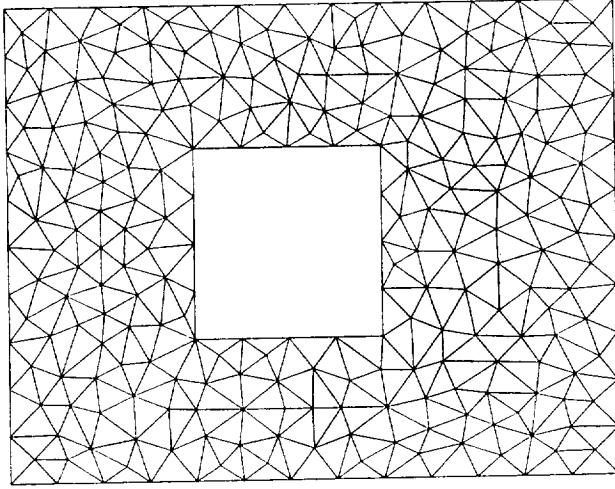Connectivity is determined by cell nodes, cell faces,

Figure 1: Final coarse triangulation of domain

NCELL(1:3,49)=37,118,16

NCELL(4:6,49)=1,53,62

NFACE(1:2,118)=49,513

Figure 2: Connectivity requirements for a single cell

and face cells. Cell nodes are the nodes at the vertices of a triangle. The cell faces are the edges of a triangle. The face cells are the cells that lie on either side of a given edge. Typically the x, y coordinates of the cell nodes are the only grid floating point numbers required as input by a flow code. The grid connectivity is defined through integer arrays. Two connectivity arrays were needed in this work to define the discretized geometry of a computational domain, Fig. 2. The two-dimensional array $NCELL$ contains the edge and node connectivity for a given cell. The first dimension of $NCELL$ contains 6 elements. The second dimension has a length n, where n is the total number of triangular cells in the computational domain. Consider the cell number i. The first three elements of the first dimension of array $NCELL$ are the edge numbers of cell i. The last three elements of the first dimension of array $NCELL$ are the vertex node numbers of cell i. This allows the access to the edge and node numbers that define a given cell, in this case cell number i. The array $NFACE$ is also two-dimensional. It contains cell connectivity for a specific edge. The first dimension is of length 2. The second dimension is of length m, where m is the total number of edges that make up the computational domain. The second dimension identifies the edge(in this example, j). The first dimension of array $NFACE$ contains the cell numbers that are adjacent to one another sharing the common edge j. These two integer arrays along with the floating point arrays x and y define the geometry for the present scheme.

Boundary information must be defined explicitly. Solid wall, exit, and inlet boundaries are implied through the edge connectivity array, $NFACE$. For a solid wall boundary, one of the elements of the first dimension of the array $NFACE$ will contain the value 0. This tells any cell that refers to that edge that it borders a solid wall boundary. Similarly, an exit boundary is adjacent to a cell number of -1, and an inlet boundary borders a cell number of -2. Periodic and symmetric boundaries are handled through special connectivity. This again is done by including the appropriate cell information in array $NFACE$. A symmetry boundary cell will have an edge that borders itself. So the first dimension of array $NFACE$ for the symmetry face will have both elements referring to the same cell number. For a periodic boundary the elements will refer to cell numbers that are separated by one periodic pitch. Boundary conditions will be addressed later.

5   Discretization Technique

The finite volume formulation of the governing equations is well suited for application to an unstructured discretization of the flow domain. The nondimensional Navier-Stokes equation written in differential form Eq. (4) is first recast in integral form for an arbitrary volume, V as

$$\int_\Omega \left[\frac{\partial Q}{\partial t} + \left(\frac{\partial G}{\partial x} + \frac{\partial H}{\partial y} \right) \right] dV = 0. \qquad (10)$$

Using Gauss's theorem, the area integral of the flux derivatives can be rewritten as the surface integral of the flux quantities around the perimeter of the volume, V. This allows the Eq. (10) to be written as

$$\frac{\partial}{\partial t} \int_\Omega Q dV + \oint_\Gamma (G dy - H dx) = 0. \qquad (11)$$

For each control volume consisting of a triangular element, Eq. (11) is evaluated as

$$\frac{d}{dt}(A_i Q_i) + \sum_{j=1}^{3}(G_j \,\triangle\, y_j - H_j \,\triangle\, x_j) = 0. \quad (12)$$

where Q_i is the vector of conserved quantities in cell i, G_j and H_j are the flux vector quantities across edge j, and $\triangle x_j$ and $\triangle y_j$ are the differences in Cartesian nodal coordinates that define edge j. The summation on j proceeds in a counterclockwise manner around the edges of cell i. Also it is understood that $\triangle x_j$ stands for $x(end) - x(beginning)$ as the evaluation proceeds in a counterclockwise manner around the sides of a control volume. The quantity A_i is the area of cell i defined as

$$A_i = \sqrt{s(s-a)(s-b)(s-c)}. \quad (13)$$

The variable s is the semiperimeter of cell i,

$$s = \frac{1}{2}(a+b+c) \quad (14)$$

and the quantities a, b, c refer to the lengths of the sides of the cell i. Cell face flow quantities required by Eq. (12) for the computation of the inviscid flux terms were approximated by using the average of the cell centered values on both sides of a given cell face. The numerical integration of these quantities around the edges of the cell results in a central difference scheme that is second order accurate in space. The viscous terms required the computation of the derivatives on the faces of the triangle control volume. To compute these terms, the level 2 cells shown in Fig. 3 were used and a different path integral was evaluated. Again this yields a second order accurate scheme in space. A total of 10 cell centered quantities was used in the computation of the viscous quantities of the summation term of Eq. (12). Specifically the viscous terms in the x-momentum equation are written as

$$\sum_{j=1}^{3}(-\tau_{xx} \,\triangle\, y + \tau_{xy} \,\triangle\, x)_j$$

$$= \sum_{j=1}^{3} \frac{\mu R}{Re} [-\frac{2}{3} \left(2\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} \right) \triangle y \quad (15)$$

$$+ \left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right) \triangle x]_j,$$

where the summation is over the three sides of the triangular control volume. A typical derivative can be recast in integral form as
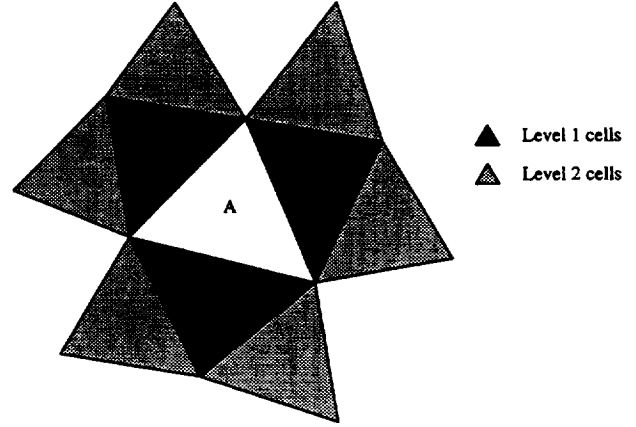
Figure 3: Cell level dependence

$$\frac{\partial u}{\partial x} = \frac{1}{S'} \oint_{\Gamma'} u\,dy \quad (16)$$

where S' is the sum the areas of the two cells across a given edge, and the integral is along the path that traverses the outer boundary of the two cells in a counterclockwise direction. These derivatives are interpreted as mean values over the area S'.

The system of equations was integrated in time using an implicit scheme written in delta form. Newton linearization was used on nonlinear terms. For example, the terms

$$\frac{P}{T} \simeq \frac{\hat{P}}{\hat{T}} + \frac{1}{\hat{T}}\triangle P - \frac{\hat{P}}{\hat{T}^2}\triangle T,$$

$$\frac{Pu}{T} \simeq \frac{\hat{P}\hat{u}}{\hat{T}} + \frac{\hat{P}}{\hat{T}}\triangle u + \frac{\hat{u}}{\hat{T}}\triangle P - \frac{\hat{P}\hat{u}}{\hat{T}^2}\triangle T, \quad (17)$$

$$\frac{Pv}{T} \simeq \frac{\hat{P}\hat{v}}{\hat{T}} + \frac{\hat{P}}{\hat{T}}\triangle v + \frac{\hat{v}}{\hat{T}}\triangle P - \frac{\hat{P}\hat{v}}{\hat{T}^2}\triangle T$$

are substituted into the continuity and momentum equations. The ^ terms take on provisional values of the primitive variables; and the delta quantities represent the differences between values at the new time level and the provisional values, e.g., $\triangle u = u - \hat{u}$. These equations can be iterated at a specific point in time until the linearization errors are reduced to a satisfactory level. For steady state problems, this iteration process at each time level is not necessary. Similar terms are used to linearize the energy equation. However, the nonlinear dissipation terms in the energy equation were linearized by evaluating them explicitly in terms of the provisional values, rather

than using Newton linearization. The result of the above substitutions is the matrix equation

$$A\vec{x} = \vec{b}. \tag{18}$$

The matrix A contains the linearized terms which multiply the vector of unknown delta quantities, $\vec{x}$. The vector $\vec{b}$ represents the residual of the equations which should go to zero as the solution approaches convergence. There are four equations written for each cell in the computational domain. For the central difference formulation given here, each cell is dependent on the level one cells through the convective terms as well as the level two cells through its diffusive terms, as illustrated in Fig. 3. This along with the unstructured grid yields a sparse matrix whose elements are block 4×4 matrices. The general row of the matrix A has ten non-zero blocks of coefficients. The number of blocks vary near a boundary.

6   Artificial Dissipation

Artificial dissipation was needed in the current implementation of the flow equations to prevent the odd-even decoupling seen in central difference computer flow codes. Two schemes were used in this work. The first scheme was based on the research of Jameson and Mavriplis [6] and was used specifically for inviscid subsonic and transonic test cases. The second version was developed to be used with the low Mach number test cases where preconditioning was employed. Both dissipation schemes were added explicitly to the system of flow equations.

The second type of dissipation was developed to be used with the temporally preconditioned scheme. This form of the dissipation was used for the viscous subsonic flows computed in the present work. Here a biharmonic operator was used on the primitive flow variables. The second difference of the primitives were computed as

$$ap_{i2} = \sum_{j=1}^{3} [w_j] - 3w_i,$$

where w was the vector of primitive variables. Again the summation was done over the index j. The result represents the second difference of the variables in cell i. The fourth difference is then computed by summing the third difference over the edges of the cell,

$$ap_{i4} = \sum_{j=1}^{3} [ap_{j2}] - 3ap_{i2}.$$

The resulting fourth difference was then premultiplied by the preconditioning matrix A_p. Then it was multiplied by the appropriate coefficient to make it consistent with the other terms and included explicitly on the right-hand-side of the system of equations.

7   Boundary Conditions

For the subsonic viscous flow cases in this study, the inlet boundary conditions were imposed by specifying the velocity components, u and v, as well as the static temperature, T. The static pressure was extrapolated from the interior to the inlet. At the exit, u, v, and T were extrapolated downstream. The static pressure, P, was specified. At the solid wall the static temperature was specified. The u and v velocity components were set to zero to enforce the no-slip condition. Static pressure was specified as symmetric in the ghost cell to give a zero normal derivative at the solid surface. Symmetry and periodic boundary conditions were imposed by simply specifying the appropriate cell connectivity. At a symmetry boundary cell, values were reflected across the boundary. At a periodic boundary cell, values were transposed by the periodic pitch of the computational domain.

For supersonic viscous flow, boundary conditions at a solid wall remained the same as in subsonic flow. At the inlet, all flow quantities, P, u, v, and T were specified. At the exit all the flow quantities were extrapolated.

It is important to note that one of the test cases computed to verify the code was inviscid. For this, the viscous boundary conditions described above were modified appropriately. Since the inflow and outflow were subsonic, it was only necessary to modify the no-slip solid wall boundary condition to a tangency boundary condition.

In the current work, temporal preconditioning was used to compute low Mach number flows as described in the previous section. The preconditioning did not affect how the boundary conditions were specified, but it is interesting to note that the eigenvalues of the system were modified. The eigenvalues must now be obtained from the matrix that results from the product $A_p{}^{-1}A_x$. In one dimension

$$A_p{}^{-1} = \begin{bmatrix} \frac{\gamma M^2 \Gamma u^2}{2} & -\gamma M^2 \Gamma u & \gamma \\ -\frac{uRT}{P} & \frac{RT}{P} & 0 \\ -\frac{(\Gamma u^2 - 2RT)T}{2P} & -\frac{\Gamma Tu}{P} & \frac{\gamma RT}{P} \end{bmatrix},$$

where $\Gamma = \gamma - 1$. One eigenvalue remains unchanged that is $\lambda_1 = U$. The other two eigenvalues $\lambda_{2,3}$ for the given one-dimensional system take on a similar but yet a more complex form than that described

7

by Withington et al. [15]. However, in the limit as Mach number approaches zero, the eigenvalues of the system are

$$\lambda_1 = U$$

and

$$\lambda_{2,3} = \frac{U \pm \sqrt{U^2 + 4\gamma T}}{2}.$$

By substituting in the appropriate nondimensional quantities the resulting ratio of largest to smallest eigenvalue takes on the value of $(1 + \sqrt{5})/(1 - \sqrt{5})$. This is the same quantity that was obtained by Withington et al. [15] with the ratio of specific heats set equal to one in the present work. The preconditioning essentially allows all of the equations of the system to be integrated at the same pseudo-time rate. This can be compared with a scheme without preconditioning where the ratio of largest to smallest eigenvalues is infinity.

8 Sparse Matrix Solvers

The system of algebraic equations being solved in the present implicit unstructured grid formulation is represented by Eq. (18). The matrix **A** is sparse. There is usually no particular pattern to the nonzero elements when the matrix arises from an unstructured grid formulation. However, the blocks on the main diagonal of this sparse matrix always have some nonzero entries. Figure 4 shows a representative form of the sparse matrix **A**. The solid squares represent 4×4 blocks with at least the diagonal elements of the diagonal block being nonzero. The remaining blocks contain zeros. In the present method, a two-dimensional viscous flow computation requires a maximum of ten 4×4 blocks in each row of the **A** matrix. In contrast, an implicit structured solver can be written such that the resulting **A** matrix on the left-hand side has some special structure that allows the matrix equation to be solved by some well established methods. The equations can often be cast in a form that results in a block bidiagonal or block tridiagonal matrix. This structure is not generally available to the solution of the flow equations written for an unstructured grid.

Several iterative methods were examined in this work. The first was a point Gauss-Seidel scheme where only the diagonal elements of the diagonal blocks of matrix **A** were retained on the left-hand side as unknowns. This scheme was successful for many of the simpler problems but was prone to divergence

$$[A] =$$
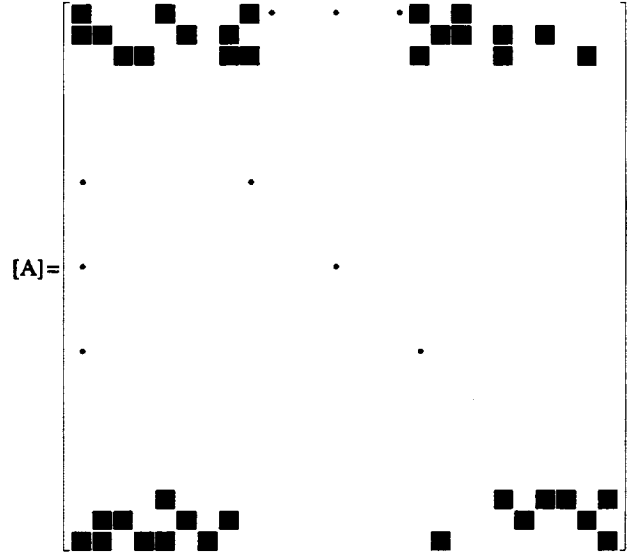
Figure 4: Form of sparse matrix **A**

when starting with poor initial conditions. It seemed to be very sensitive to lack of diagonal dominance.

Another iterative scheme used was the point block Gauss-Seidel method. Here the diagonal 4×4 blocks of matrix **A** were retained on the left hand side. The remaining matrix equation was solved using **LU** decomposition. The **L** and **U** matrices refer to the lower and upper triangular decomposition of the diagonal block of the matrix **A**. This was found to be more robust than the previous scheme.

Even though the full sparse block matrix is $N \times N$, it is only necessary to store the nonzero blocks. This gives a maximum block matrix of $10 \times N$ for a viscous code. However, the bandwidth could still be the maximum, N.

The commercially available sparse iterative solver, SITRSOL [17], which resides on the Cray YMP as a callable subroutine was also evaluated for solving the above matrix equation. SITRSOL takes advantage of the matrix sparseness by only storing the nonzero entries. The package makes available to the user several iterative methods as well as preconditioners for solving non-symmetric positive indefinite sparse linear systems. In the present work the biconjugate gradient method, the generalized minimal residual method, and the generalized conjugate residual method were considered. An incomplete **LU** preconditioner was also used. These three iterative methods are of the preconditioned conjugate gradient type.

The iterative solver SITRSOL was used on one of the test cases to be shown in the results section and

its effectiveness was compared with that of the point block Gauss-Seidel method. The conclusions shown in this work are provisional. More experience needs to be obtained to make a true evaluation of the various solvers and preconditioners.

The solution of Eq. (18) using a point block Gauss-Seidel method suffers from recurrence. The penalty is seen in vectorization. This recurrence can be eliminated with a minimum effect on the solution convergence rate by using a coloring scheme. The idea comes from a problem which arose in graph theory. A theorem states that a map can be colored with only four colors such that no two regions of the same color share a border. The conjecture was proven through exhaustive computation by Appel and Haken [16] in 1976. This theorem was implemented by first coloring the unstructured grid according to the theorem and storing all cell numbers of given color in an integer array. The scheme was most efficient when the number of cells in each color integer array was about equal.

This gave a color map that was then used as input to the flow code. The Gauss-Seidel algorithm was then written to contain four loops corresponding to the four colors of the colored grid. Each single colored loop contained no level 1 cell recurrence, so it was vectorized. On a typical problem in the present study, the solution time for the algebraic system(the Gauss-Seidel subroutine) was reduced by a factor of 7.6 times by using this four color partitioning. Recurrence is still present but only through the level 2 cells, illustrated in Fig. 3, required in the viscous terms. The result is that the quantities in the level 2 cells are lagged from the previous iteration time step. However, this does not seem to effect the convergence rate.

9 Results

The results presented in this section will be used to demonstrate two conclusions. The initial results will show the validity of the code. And later results will indicate the versatility of the unstructured grid over the structured grid formulation. Comparisons will be made with data available from other investigators.

9.1 Bump on Wall

Inviscid flow over a bump in a channel was computed at two values of Mach number. A Mach number of 0.5 was used for the first test case. Figure 5 shows Mach number contours. The flow was subsonic so the inviscid flow was symmetric about the middle of the bump. This could be seen more clearly when the
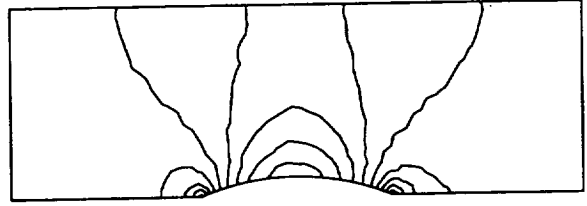
Figure 5: Constant Mach number contours for flow over a symmetrical bump in a channel, $M_{in} = 0.5$
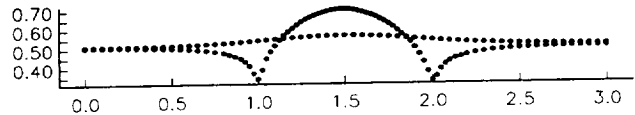
Figure 6: Upper and lower wall Mach number distribution

upper and lower wall Mach number distributions were plotted. The results of this subsonic case compared well with those reported by Ni, [18].

A second test case was computed at a Mach number of 0.675 at the inlet. Here the flow was transonic over the bump. The grid used for this test case can be seen in Fig. 7. A supersonic bubble was formed on the bump, Fig. 8. The location of the shock was shown clearly in the plot of upper and lower wall Mach number distributions, Fig. 9. The location of the shock compared well with the results of Ni, [18] as well as that of Chima et al, [19]. The sonic line that impinged on the aft side of the bump was at a distance of 72 percent of the chord length from the head of the bump in the above cases. The present case locates the sonic line at 73 percent of the chord length.

These inviscid test cases required the addition of artificial dissipation. For the subsonic case, a fourth difference was added to prevent the odd-even decoupling of the solution seen in central difference schemes. The transonic case also required the additional second difference to prevent oscillations from occurring about the discontinuity. In both cases the dissipation model was similar to that of Jameson et al. [3]. Later Jameson and Mavriplis [6] implemented this type of dissipation model for an explicit unstructured grid flow solver.

9.2 Developing Channel Flow

Developing flow in a channel was used to validate the code for viscous flows. It also served the purpose of testing the preconditioning used for computing low Mach number flows. Comparisons were made between the Gauss-Seidel method and the solver SITR-SOL for solving the sparse matrix equation.

The code was validated on four developing channel flow test cases. A low inlet Mach number flow of 0.05 was used to compute flows at Reynolds numbers of 1, 20, 150, and 1500 based on the inlet uniform velocity, density, and full channel height. Because the inlet Mach number was held constant, the channel height was varied to obtain the appropriate Reynolds number. Unstructured grids of 1114,1969,4800, and 4800 cells were used for the Reynolds number flows of 1, 20, 150, and 1500 respectively. Uniform flow enters the channel with a nondimensional uniform velocity of one and accelerates to a nondimensional centerline velocity of 1.5. In order to compare with published results for incompressible flows, it was necessary to make a correction to the centerline velocity at the low Reynolds numbers due to the larger density variation from the inlet to the exit of the channel. Figure 10 shows the centerline velocity of the channel flow at various Reynolds numbers. These results were compared with other computations by Tenpas and Pletcher [20], Morihara and Cheng [21], and Chilukuri and Pletcher [22]. At a Reynolds number of 20 the centerline velocity of the current study slightly under predicted the centerline velocities obtained by the other investigators near the exit of the channel. At a Reynolds number of 1500, the results of the present study show a more rapid acceleration of the flow than indicated by the solution of the partially parabolized Navier-Stokes equations.

Typical convergence histories for the code are shown in Fig. 11. The convergence criteria was based on the residual of the continuity equation in delta form which should approach machine zero as the solution goes to a steady state. The solution of the matrix equation was done by the block Gauss-Seidel method. In general, the solution converged at nearly the same rate over a wide range of Mach numbers holding the Reynolds number equal to 20 for the four flow test cases. This illustrates the benefits of the preconditioning. Without preconditioning, it was necessary to run the code at a much smaller time step thus decreasing the rate of convergence. At Mach numbers lower than 0.1 the code without preconditioning did not converge.

The sparse matrix solver SITRSOL was used for comparison with the Gauss-Seidel method. The con-
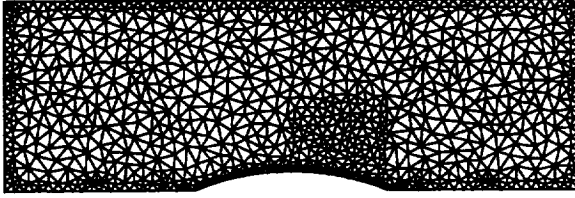
Figure 7: Computational grid for the symmetrical bump in a channel test case, $M_{in} = 0.675$
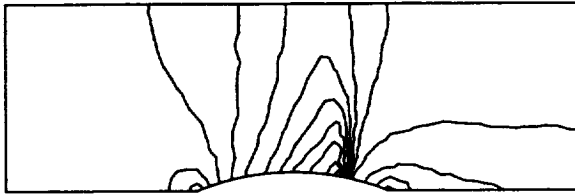
Figure 8: Constant Mach number contours for flow over a symmetrical bump in a channel, $M_{in} = 0.675$
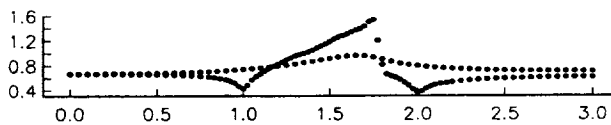
Figure 9: Upper and lower wall Mach number distribution
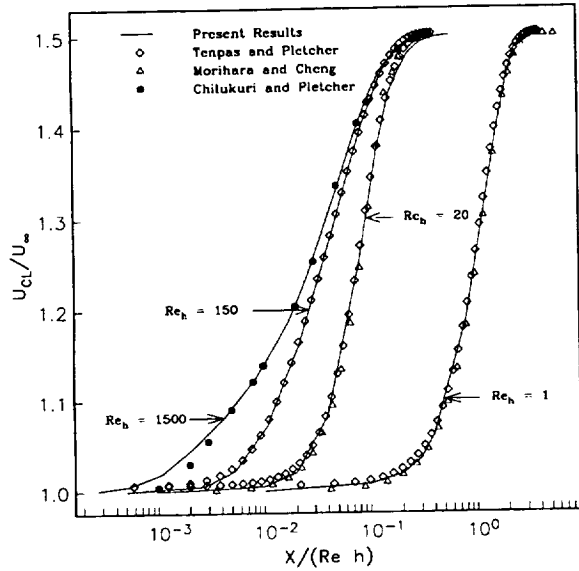
Figure 10: Centerline velocity profiles for developing flow in a channel at $M_{in} = 0.05$ with $Re_h = 1, 20, 150, 1500$

Figure 12: Convergence history of developing flow in a channel, $Re_h = 20$ computed with sparse matrix solver

Figure 11: Convergence history for developing channel flow over a range of Mach numbers at $Re_h = 20$ using the block Gauss-Seidel solver

vergence history for three different conjugate gradient like methods with the **ILU** used as a preconditioner is shown in Fig. 12. The Gauss-Seidel method without the coloring algorithm took 13.5 minutes on the Cray YMP. The same computation with a color map supplied for vectorization of the Gauss-Seidel algorithm gave a speedup factor of 7.6 over the standard Gauss-Seidel matrix solver. The overall computer time was reduced to 11.4 minutes, or a speed up of 15.5 percent. This suggests that more attention should be given to the vectorization of other parts of the flow code. The coloring scheme did not have much effect on the convergence history of this viscous calculation. The same grid was used to make comparisons with SITRSOL. The bi-conjugate gradient method took 9.3 minutes of computer time to reach about the same level of convergence as the Gauss-Seidel method. The generalized conjugate gradient residual method required 10.23 minutes of computer time. About the same level of convergence was obtained by the generalized minimum residual method in 5.5 minutes.

9.3 Sudden Expansion

The previous test cases could have easily been computed using a structured grid approach. The sudden expansion test case demonstrates the capability of the unstructured grid generation and its abil-

11

ity to obtain a grid in a domain that would otherwise need a patched or masked grid to work for a structured flow code. The results from this computation were compared with the experimental results obtained by Durst et al. [23]. They noted that though at lower Reynolds numbers the flow was symmetric about the centerline of the expansion, there were three-dimensional effects near the separated regions. A plane symmetric sudden expansion with a downstream channel height to step height ratio of 3:1 was computed. The Reynolds number for this flow was 56 based on the upstream channel height and the centerline upstream velocity. A fully developed parabolic profile was prescribed at the inlet which was located one step height upstream of the expansion. The Reynolds number was computed at a streamwise location 0.25 step heights upstream of the expansion. It was interesting to note that the profile at this location was already anticipating the expansion corner. The flow near the wall begins to accelerate; and to conserve mass, the centerline velocity decreases. The flow separates at the step, reattaches downstream, and returns to a fully developed profile about ten step heights from the expansion. The streamwise velocity component at six specific channel locations are shown in Fig. 13. Comparisons were made with the laser anemometer experimental data presented by Durst et al. [23]. The centerline velocity distribution was compared with the laser anemometer data and with the viscous-inviscid interaction computational method of Kwon et al. [24] and is shown in Fig. 14. The predicted centerline velocity appears larger than the experimental values downstream, but the correct value of one-third the upstream fully developed centerline velocity was obtained in the present calculation.

9.4 Periodic Tandem Circular Cylinders in Cross Flow

The flow was computed over a cascade of tandem circular cylinders. This computation should be of practical interest in that geometries of this sort are encountered when modeling flow through heat exchangers. These tube heat exchangers can be found in automobile radiators, room heaters and gas and air heaters. With the unstructured grid formulation, it was easier to generate a computational grid about in-line as well as staggered cascades of tubes. Some of the geometric quantities that affect the flow characteristics of the heat exchanger are the size and shape of the tubes as well as their vertical and horizontal spacing. This type of parametric study is ideal for the unstructured grid formulation.

The model problems presented here were compared
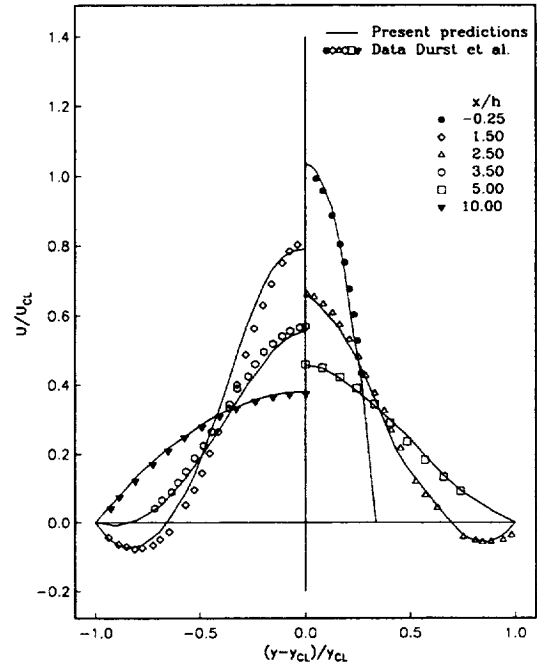
Figure 13: Velocity profiles for a laminar flow in a channel with a 3:1 symmetric sudden expansion, $Re_h = 56$
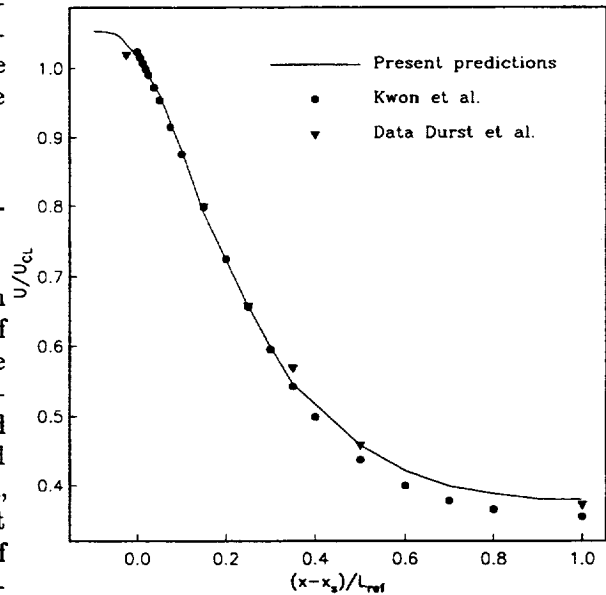
Figure 14: Centerline velocity distribution for a laminar flow in a channel with a 3:1 symmetric sudden expansion, $Re_h = 56$

12
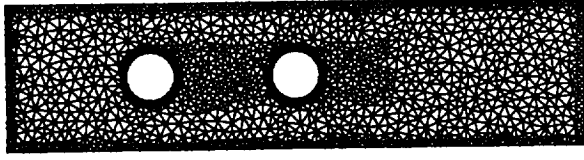
Figure 15: Grid about periodic tandem circular cylinders

Streamlines, Gordon
Present Results
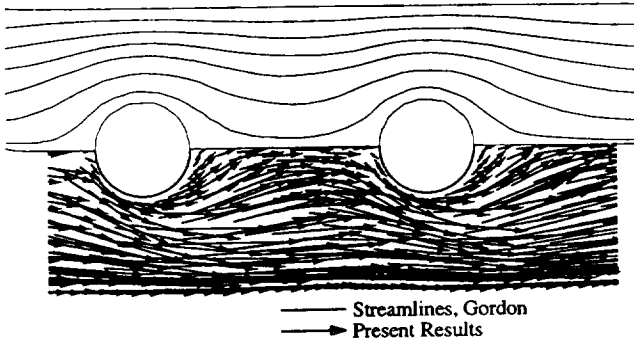
Figure 16: Periodic tandem circular cylinders in cross flow, $Re_r = 1$

Streamlines, Gordon
Present Results

Figure 17: Periodic tandem circular cylinders in cross flow, $Re_r = 20$

with the incompressible numerical results of Gordon [25]. Uniform flow conditions were prescribed perpendicular to the cascade upstream of the first bank of tubes. Periodic boundary conditions were imposed at the upper and lower geometric boundaries to simulate an infinite number of parallel rows. The tubes in this case were in-line. An upstream Mach of 0.05 was used for both computed test cases. Both cases were computed on the same geometry. The geometry used in both test cases and the grid used for the second test case is shown in Fig. 15.

The first test case was computed at a Reynolds number of 1.0 based on the upstream conditions and cylinder radius. The velocity vectors are shown in Fig. 16 and compare well with the streamlines computed by Gordon [25]. At this Reynolds number the flow was nearly symmetric about both cylinders indicating that there was minimal influence of one bank of cylinders on the other.

The second test case was computed at a Reynolds number of 20.0 based on the same conditions as the test case above. Here the flow separates behind both cylinder banks. An enlargement of the velocity vectors near the cylinders is shown in Fig. 17 and the length of the separated regions behind both of the cylinders compares well with those computed by Gordon [25]. As noted by Gordon [25], the length of the separation behind the second bank of tubes is slightly smaller than that behind the first bank. The first set of cylinders accelerates the flow in the freestream so

the slower wake flow impacts the second set of cylinders. The flow was symmetric about an imaginary horizontal line that passed through the centers of the cylinders. There does not seem to be any difference in the angular location of the actual separation point on either cylinder.

9.5 Four Port Valve

The final results are presented to show the versatility of applying boundary conditions when using an unstructured grid flow solver. The geometry represents a two-dimensional valve with four ports where inlet or outlet flow boundary conditions can be specified. A reference was made to such a flow geometry in an article by Ackert [26]. The actual flow conditions were not given. Here the flow was computed at two Reynolds numbers. Fluid enters through a channel on the left, enters the circular cavity, and exits through a channel at the bottom. For both test cases fully developed conditions were prescribed at the inlet. The flow redevelops along the open channel. Both test cases were computed with an inlet Mach number of 0.05. An interesting aspect of the geometry was that the closed valve ports acted as driven cavities. The unstructured grid formulation allows the application of exit boundary conditions at any or all of the three remaining ports. This type of valve geometry can be found in an application like fluid networks.

Fluid flow was first simulated in the valve geometry at a Reynolds number of 10 based on inlet conditions and channel height. The velocity vectors of this steady state flow are shown in Fig. 18. Here the fluid near the wall of the inlet channel was accelerated as the corner of the cavity was anticipated. A clockwise rotation of the fluid was followed through the circular volume. The fluid in the closed cavity ports was driven in a counterclockwise rotation. The band of fluid then enters the open lower exit channel and again becomes fully developed.

Figure 19: Four port valve, $Re_h = 50$

Figure 18: Four port valve, $Re_h = 10$

Next the same valve geometry was used to simulate the fluid flow at a Reynolds number of 50 based on inlet conditions and channel height. The velocity vectors of this computation are shown in Fig. 19. Again a fully developed velocity profile was specified at the inlet to the channel. The velocity of the fluid near the wall accelerates as it approaches the entrance to the circular chamber. Contrary to the previous case, the banded fluid actually drove a large volume of fluid in a counterclockwise direction. This had the effect of driving the closed valve ports in an opposite rotation direction from that of the lower Reynolds number test case. Also, the band of fluid did not diffuse as much across the circular volume. In the open exit channel the fluid redevelops into a fully developed parabolic profile.

10 Conclusions

A two-dimensional unstructured grid implicit flow solver was described. Although only internal flow problems were considered in this study, the method is believed to be applicable to external flows as well. The compressible flow equations were discretized in

14

finite volume form.

Several conclusions can be drawn from the present study.

1. A triangular unstructured grid can be generated about very complex geometries where the use of a single structured grid cannot be considered in most cases. This gives the advantage that a single computer code can be used in a wide variety of flow geometry applications. However, this advantage is somewhat dampened by the complexity of coding required for solving a system of differential equations on this unstructured grid. Details such as boundary conditions are more difficult to implement on an unstructured grid.

2. It was found that the diagonal block Gauss-Seidel solver was more robust than the point diagonal Gauss-Seidel version of solver. The diagonal point solver seemed sensitive to initial conditions and diagonal dominance.

3. A coloring scheme was used to take advantage of the vectorization of the implicit Gauss-Seidel solver. A minimum of extra storage was necessary for a significant reduction in computer time. The time spent in the solver was decreased by a factor of 7.6. It was found that the recurrence in the viscous fluxes had little affect on the convergence of the solution to a steady state.

4. The use of the sparse matrix iterative solver allowed a much larger time step to be used than that of the Gauss-Seidel solver. However, every time step using the sparse matrix solver was significantly more expensive. Even so, the sparse solver ran at 2 to 2.5 times faster than the block Gauss-Seidel solver. Several different conjugate gradient like solvers were tested with the matrix preconditioners available with SITRSOL. Some of the preconditioners did not allow a solution to the equations. The incomplete LU preconditioner was found to be the best. The solvers all exhibited the same basic convergence rate. The difference in the solvers was in the time that was required to obtain the same convergence level. The generalized minimum residual method was found to be the fastest for the particular test case that was being computed.

5. A temporal preconditioning was added to the flow equations to allow solutions at very low Mach numbers. The preconditioner was implemented such that both steady state and time accurate flows could be computed. Steady state solutions were considered in this study. Mach number flows as low as 0.0005 were computed without degradation to the convergence rate of the solution procedure. Without the preconditioning, convergence was either very slow due to the necessity of running at a much smaller time step, or the equations could not be converged to a solution. Preconditioning was relatively easy to add to the numerical code.

Acknowledgment

References

[1] Steger, J., and Warming, R. F., "Flux Vector Splitting of the Inviscid Gasdynamics Equations with Application to Finite-Difference Methods," NASA TM-78605, 1978.

[2] MacCormack, R. W., "A Numerical Method for Solving the Equations of Compressible Viscous Flow," AIAA 81-0110, 1981.

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

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

[5] Jameson, A., "Successes and Challenges in Computational Aerodynamics," AIAA 87-1184, 1987.

[6] Jameson, A., Mavriplis, D., "Finite Volume Solution of the Two Dimensional Euler Equations on a Regular Triangular Mesh," AIAA 85-0435, 1985.

[7] Mavriplis, D., Jameson, A., Martinelli, L., "Multigrid Solution of the Navier-Stokes Equations on Triangular Meshes," AIAA 89-0120, 1989.

[8] Venkatakrishnan, V., Barth, T. J., "Application of Direct Solvers to Unstructured Meshes for the Euler and Navier-Stokes Equations Using Upwind Schemes," AIAA 89-0364, 1989.

[9] Barth, T. J., Jespersen, D. C., "The Design and Application of Upwind Schemes on Unstructured Meshes," AIAA 89-0366, 1989.

[10] Holmes, D.G., and Connel, S. D., "Solution of the 2D Navier-Stokes Equations on Unstructured Adaptive Grids," AIAA 89-1932, 1989.

[11] Batina, J. T., "Development of Unstructured Grid Methods for Steady and Unsteady Aerodynamic Analysis," Presented at the 17th Congress of the International Council of the Aeronautical Sciences, Stockholm, Sweden, 1990.

[12] Holmes, D.G., and Snyder, D. D., "The Generation of Unstructured Triangular Meshes Using Delaunay Triangulation," *Numerical Grid Generation in Computational Fluid Mechanics '88*, Pineridge Press, Miami, 1988. pp. 643-652.

[13] Choi, D., Merkel, C. L., "Application of Time-Iterative Schemes to Incompressible Flows," AIAA 84-1638, 1984.

[14] Turkel, E, "Preconditioned Methods for Solving the Incompressible and Low Speed Compressible Equations," *Journal of Computational Physics*, 72, pp. 277-298, 1987.

[15] Withington, J. P., Shuen, J. S., Yang, V., "A Time Accurate, Implicit Method for Chemically Reacting Flows at All Mach Numbers," AIAA 91-0581, 1991.

[16] Appel, K. and Haken, W., "Every planer map is 4-colorable," *Bull. Am. Math. Soc.* 82, pp. 711-712, 1976.

[17] Cray Research, Inc., "Volume 3: UNICOS Math and Scientific Library Reference Manual," SR-2081 6.0, pp. 227-243, 1991.

[18] Ni, R., "A Multi-Grid Scheme for Solving the Euler Equations," AIAA 81-1025, 1981.

[19] Chima, R. V., Turkel, E., Schaffer, S., "Comparison of Three Explicit Multigrid Methods for the Euler and Navier-Stokes Equations," AIAA 87-0602, 1987.

[20] Tenpas, P. W. and Pletcher, R. H., "Solution of the Navier-Stokes equations for subsonic flows using a coupled space-marching method," AIAA 87-1173-cp, 1987.

[21] Morihara, H. and Cheng, R. T., "Numerical solution of the viscous flow in the entrance region of parallel plates," *Journal of Computational Physics*, 11, No. 4, pp. 550-572, 1973.

[22] Chilukuri, R. and Pletcher, R. H., "Numerical solution to the partially parabolized Navier-Stokes equations for developing flow in a channel," *Numerical Heat Transfer*, 3, No. 2, pp. 169-187, 1980.

[23] Durst, F., Melling, A., Whitelaw, J. H., "Low Reynolds number flow over a plane symmetric sudden expansion," *Journal of Fluid Mechanics*, 64, pp. 441-428, 1974.

[24] Kwon, O. K., Pletcher, R. H., and Lewis, J. P., "Prediction of sudden expansion flows using the boundary-layer equations," *Trans. ASME, J. Fluids Engineering*, 106, pp. 285-291, 1984.

[25] Gordon, D., "Numerical Calculations on Viscous Flow Fields through Cylinder Arrays," *Computers and Fluids*, 6, pp. 1-13, 1978.

[26] Ackeret, J., "Aspects of Internal Flow," Proceedings of the Symposium on the Fluid Mechanics of Internal Flows, General Motors Research Laboratories, Warren, MI, 1965.

REPORT DOCUMENTATION PAGE

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	January 1994	Technical Memorandum

4. TITLE AND SUBTITLE

An Implicit Numerical Scheme for the Simulation of Internal Viscous Flows on Unstructured Grids

5. FUNDING NUMBERS

WU–505–62–52

6. AUTHOR(S)

Phillip C.E. Jorgenson and Richard H. Pletcher

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135–3191

8. PERFORMING ORGANIZATION REPORT NUMBER

E–8279

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, D.C. 20546–0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA TM–106437
AIAA–94–0306

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified - Unlimited
Subject Category 64

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The Navier-Stokes equations are solved numerically for two-dimensional steady viscous laminar flows. The grids are generated based on the method of Delaunay triangulation. A finite-volume approach is used to discretize the conservation law form of the compressible flow equations written in terms of primitive variables. A preconditioning matrix is added to the equations so that low Mach number flows can be solved economically. The equations are time marched using either an implicit Gauss-Seidel iterative procedure or a solver based on a conjugate gradient like method. A four color scheme is employed to vectorize the block Gauss-Seidel relaxation procedure. This increases the memory requirements minimally and decreases the computer time spent solving the resulting system of equations substantially. A factor of 7.6 speedup in the matrix solver is typical for the viscous equations. Numerical results are obtained for inviscid flow over a bump in a channel at subsonic and transonic conditions for validation with structured solvers. Viscous results are computed for developing flow in a channel, a symmetric sudden expansion, periodic tandem cylinders in a cross-flow, and a four-port valve. Comparisons are made with available results obtained by other investigators.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Navier-Stokes equations; Central difference; Unstructured grid; Implicit; Preconditioning	20
	16. PRICE CODE A03

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102