To the Graduate Council:

I am submitting herewith a dissertation written by Sudheer Nath Nayani entitled " A Locally Implicit Scheme For Navier-Stokes Equations." I have examined the final copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Mechanical Engineering.

K. C. Reddy, Major Professor

We have read this thesis
and recommend its acceptance:

Accepted for the Council:

Vice Provost
and Dean of the Graduate School

A LOCALLY IMPLICIT SCHEME FOR NAVIER-STOKES EQUATIONS

A Dissertation

Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

Sudheer Nath Nayani

June 1988

ACKNOWLEDGEMENTS

The author is greatly indebted to Dr. K.C.Reddy who defined this topic of study, supervised its progress and guided at every stage, and without which this effort would not have been possible. The author aiso wishes to express his gratitude to other committee members: Dr. R.L.Young, Dr. J.M.Wu, Dr. Jack Benek and Dr. Trevor Moulden.

The author would like to express his gratitude to Mrs. Melisa Weatherford and Mrs. Linda Williams for their help in typing the thesis. Finally, the author is indebted to his wife Usha and son Vikram for their constant encouragement and patience.

ABSTRACT

A locally implicit relaxation scheme, is developed for the steady-state solution of the Navier-Stokes Equations. The method uses finite volume spatial discretization, implicit time integration, Jameson-type artificial dissipation terms and a modified Gauss-Seidel iteration. The scheme does not require any matrix solvers. The convergence rate is significantly enhanced by incorporating the multigrid technique. Applications to the transonic flows show that the method is efficient and robust. The basic scheme can be implemented in two modes, namely, single point scheme and multi-point scheme. Both these methods have been analyzed in detail along with their local linear stability analysis for model elliptic Equation in single as well as in three-dimensions, and the scheme is shown to be unconditionally stable. Subsequently, the single point scheme has been applied to thin-layer Navier-Stokes Equations for a variety of test cases, namely, subsonic and transonic flows over NACA0012 and RAE2822 airfoils and the numerical results have been compared with the available experimental results. An algebraic eddy viscosity turbulence model has been used for turbulent flows. Both elliptic and hyperbolic grid generators have been used to generate C-grids for the above mentioned applications.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

t	time
C_L	lift coefficient
C_D	drag coefficient
C_p	pressure coefficient
u	velocity in x direction
v	velocity in y direction
U	contravariant velocity see Equation(21)
V	contravariant velocity see Equation(22)
J	metric Jacobian
γ	ratio of specific heats
μ	molecular viscosity
μ_t	turbulent viscosity, friction velocity
Re	Reynolds number
Pr	Prandtl number
P_{rt}	turbulent Prandtl number
ρ	density
p	pressure
Q	conserved variables vector
d	artificial dissipation in a particular direction
$\varepsilon^{(2)}$	factor used in artificial dissipation
$\varepsilon^{(4)}$	factor used in artificial dissipation
ν	factor used in artificial dissipation, kinematic viscosity
A, B	inviscid Jacobian matrix
ω	relaxation factor, vorticity
e	error in the solution, energy

r	see Equation(3)
Res	residue
g	amplification factor
du	see Equation(4)
$\Delta u, \Delta Q$	change in the solution per time step
FF	forcing function used in multigrid
L	any operator
I	injection or interpolation operator
τ	shear stress, time
a, c	speed of sound
K	Clauser constant, see Equation(41)
E, F	inviscid flux vectors
S	viscous flux vector
R	simplified viscous jacobian matrix
CFL	Courant number
SJ	see Equation(69)
SK	see Equation(68)
$U1, V1$	velocities, see Equation(70)
D	dissipation vector, see Equation(84)
k	thermal conductivity
y^+	law of the wall coordinate

Subscripts

i, j	x-direction
k	y-direction
ξ, η	directions in the computational plane
∞	free stream condition

in inner

out outer

max maximum

min minimum

Superscripts

m iteration sweep count

n time level

$\hat{}$ see Equation(31)

CHAPTER 1

INTRODUCTION

The main uses of computational fluid dynamics in aeronautical science fall into two broad categories. First is providing reliable aerodynamic predictions, which enable engineerss to design better aiplanes. Significant progress has been made in this area and now there are many CFD codes in the industry which are being used in the design and analysis phases of aerospace vehicles. Second is the use of using computational fluid dynamics for purely scientific investigations[1]. It seems possible that numerical simulation of complex flows not readily accessible to experimental measurements can provide new insights into the underlying physical processes. In particular, computational methods offer a new tool for the study of structures in turbulent flow, and the mechanisms of transition from laminar to turbulent flow.

In the last two decades reliable methods have been developed for predicting transonic flows with shock waves, which require the use of a nonlinear mathematical model. The first breakthrough was the scheme of Murman and Cole[2] and Murman[3] treating the transonic small disturbance Equation. This was a catalyst for widespread development of methods for calculating transonic potential flows in two and three dimensions using either the small disturbance Equation or the full potential Equation.

In parallel there have been efforts to devise efficient algorithms for solving the Euler and Navier-Stokes Equations. Following the pioneering efforts of Magnus and Yoshihara[4], McCormack has introduced his famous explicit difference scheme in 1969[5]. Efforts to improve efficiency has led to the implicit scheme of Beam and Warming [6], which has been adapted to general curvilinear coordinates

by Steger[7]. The need to find a better shock capturing method has also been apparent, and has stimulated the introduction of flux splitting [8]. By 1979,however, Euler methods have remained very expensive, and have not attained the levels of accuracy which justified their routine use for engineering design. Nevertheless, it has already been evident that advances in the available computing power would soon make it entirely feasible to solve the three dimensional Euler Equations, and the eighties have seen widespread efforts to realize this objective. The alternating direction method has been systematically developed into an effective tool, and the current state of art is represented by the computer codes ARC2D and ARC3D[9]. Implicit schemes using LU decomposition[10] and relaxation have also proved successful. Another path of development has led to multistage explicit time stepping schemes[11]. Jameson's FLO52 and FLO57 programs using this concept have been widely used. Stemming from the mathematical theory of shock waves, procedures have also been developed for the design of effective shock capturing schemes. There have been intensive efforts to find more rapidly convergent methods to find steady state solutions. In particular, the use of multigrids, first introduced by Federenko, and subsequently developed by Brandt[12], has been extended to the treatment of hyperbolic systems[13] and has proved to be very effective.

In the following work a locally implicit scheme is presented to solve time-dependent Navier-Stokes Equations for fluid flow problems[14], which can lead to efficient codes for solving large problems. In the proposed method, the flow variables at the center of a computational cell are treated implicitly in a local sense, at every time step, by a modified Gauss-Seidel iteration process[15]. This procedure is unconditionally stable in a local linearized sense, though in practice, Courant numbers of about 10 seem appropriate for the Navier-Stokes Equations. Furthermore, there is only limited coupling at the local level of the Equations and

solution corrections are computed efficiently by explicit expressions. The scheme can be used in two modes: single point and multi-point update. Single point scheme is computationally more efficient of the two and is developedin the main text. Multi-point update scheme is described in Appendix 1[46].

The standard Gauss-Seidel scheme does not work when central differences are used for convection dominated flows. Reddy and Jacocks[15] have done extensive studies on the inviscid compressible flows and found that it is necessary to modify the Gauss-Seidel scheme and incorporate symmetric sweeping in order to make the scheme stable. Chakravarthy[16] and MacCormack[17] have developed relaxation methods successfully for unfactored implicit upwind schemes using TVD concepts or flux vector splitting , but centrally differenced schemes are computationally more efficient than upwind schemes.

In order to understand the locally implicit scheme in the presence of diffusion, the single point method has been applied first to a model Equation, in this case, a time-dependent diffusion Equation and the methodology is described in chapter 2. Initially the Equation is discretized for a one-dimensional diffusion Equation, the modified Gauss-Seidel method is defined and the Equations are derived for a symmetric iteration process. Von-Neumann stability analysis is then carried out for the difference Equations and the scheme is shown to be unconditionally stable. In order to ascertain the merits of this scheme the same analysis is made for a standard Gauss-Seidel scheme and the results are compared. It is shown that there is a range of a parameter(which is defined later)for which the modified scheme works better than the standard scheme. The scheme is then extended to the three-dimensional diffusion Equation and the results of the modified and standard Gauss-Seidel schemes are compared for both in single and multigrid modes. All

the results obtained for the model Equation are for a finite difference formulation and the modified Gauss-Seidel scheme is shown to be superior in convergence, especially in multigrid mode.

Having successfully tested the locally implicit scheme on model Equations in one and three-dimensions, the method has been extended to thin-layer Navier-Stokes Equations in two-dimensions. The application of the method is detailed in chapter 3, starting from Navier-Stokes Equations in the general curvilinear coordinates. In this case the method uses finite volume spatial discretization, implicit time integration and Jameson-type artificial dissipation terms[18] to suppress the nonlinear instabilities. The multigrid technique used here is different from that used for the model Equation. Jameson-type multigrid technique[19] is used here which significantly enhances the convergence rate. The locally implicit scheme does not require any linear Equation solvers. The same logic applied to the model Equation is extended to the Navier-Stokes Equations and the corresponding Equations are derived. The residual terms contain all the required terms of the Equations and no simplifications are made. However, in the implicit part, the nondiagonal terms of the viscous Jacobian are neglected, which saves considerable amount of computational time. An algebraic turbulence model is incorporated in the code. Several test cases have been run ranging from low subsonic to transonic flows. For subsonic laminar flow cases NACA0012 airfoil has been used and for the rest of the test cases RAE2822 airfoil is used. Body fitting and stretched C-grids have been used, using elliptic grid generator[20] for NACA0012 airfoil and hyperbolic grid generator[21] for RAE2822 airfoil. For all the cases the grid size used is 128x32. The test cases chosen are such that either published numerical or experimental data is available for comparision. The results of flows over NACA0012 airfoil have been found to be similar to numerical results of Jame-

son[22]. Cook,McDonald and Firmin have carried out extensive experimentation on RAE2822 airfoil and published the data [23]. Computer runs have been made for some of the available cases and numerical results have been compared with the experimental results. As can be seen from the results fairly good agreement has been obtained. For the turbulent flows an algebraic turbulence model has been used. It is an eddy viscosity model developed by Lomax and Baldwin[24]. The turbulence model is described in detail in Appendix 2.

CHAPTER 2

ONE POINT LOCALLY IMPLICIT SCHEME FOR ELLIPTIC PARTIAL DIFFERENTIAL EQUATIONS

2.1 One Dimensional Diffusion Equation

The basic locally implicit scheme which incorporates a modified Gauss–Seidel scheme as an inner iteration is introduced in this chapter with specific application to one and three-dimensional diffusion Equations. The method is applied first to a one-dimensional diffusion Equation, along with its local linear stability analysis. The scheme is then compared in detail with the standard Gauss–Seidel scheme and the results are presented highlighting the merits and demerits of the new scheme. Subsequently, the same methodology is extended to a three-dimensional diffusion Equation and the corresponding results are discussed. In addition a multigrid convergence acceleration technique is introduced with the locally implicit scheme. Convergence results for the standard and modified Gauss–Seidel iteration are compared.

Consider the following one dimensional diffusion Equation.

$$\frac{\partial u}{\partial t} = \nu \frac{\partial^2 u}{\partial x^2} \tag{1}$$

Euler implicit scheme for Equation (1) with central difference approximation for spatial derivative is written as

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = \nu \frac{u_{j-1}^{n+1} - 2u_j^{n+1} + u_{j+1}^{n+1}}{\Delta x^2} \tag{2}$$

This scheme, together with boundary conditions, involves the solution of a tri-diagonal system of Equations at each time step. The scheme is unconditionally stable and leads to an asymptotic solution rapidly. In multi-dimensions the matrix

is too big for direct solution methods. There are a number of iteration methods of solution in literature. The following is a new iterative method designed for obtaining asymptotic solution of Equation (1).

2.2 Delta Form of Equation (2)

Define

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

Equation (2) can be rewritten in terms of Δu's as

$$-r\Delta u_{j-1} + (1+2r)\Delta u_j - r\Delta u_{j+1} = Res_j^n \tag{3}$$

where

$$r = \nu \frac{\Delta t}{\Delta x^2}$$

and

$$Res_j^n = r\left(u_{j-1}^n - 2u_j^n + u_{j+1}^n\right)$$

Res_j^n is a known quantity and Equations (3) represents a linear system for Δu's, when it is augmented with appropriate boundary conditions.

If we use a Gauss–Seidel iteration to solve the system of Equations (3) iteratively, Δu_{j-1} is available from the previous iteration and Δu_{j+1} is set to zero to compute Δu_j while sweeping from left to right. Instead, we modify the Gauss–Seidel iteration as follows.

2.3 Inner Iteration

Define

$$du_j = \Delta u_j^{(m+1)} - \Delta u_j^{(m)} \tag{4}$$

where du_j is the correction for Δu_j in one iteration sweep and m is the iteration index count. Initially

$$\Delta u_j^{(0)} = 0$$

Rewriting Equation (3),

$$-r\left(\Delta u_{j-1}^{(m)} + du_{j-1}\right) + (1+2r)\left(\Delta u_j^{(m)} + du_j\right) - r\left(\Delta u_{j+1}^{(m)} + du_{j+1}\right) = Res_j^n \tag{5}$$

Since we are dealing with a one-dimensional problem we can limit it to 2 sweeps for a symmetrical iteration, i.e., left to right sweep followed by a right to left sweep. Asymptotic steady state solution does not depend on the evolution of Δu's as long as they tend to zero asymptotically. For obtaining time accurate solution however sufficient number of symmetric inner iteration sweeps have to be carried out till the du's become small. Let us consider each sweep along with its Fourier modal analysis.

2.4 Left to Right Sweep

At this stage $m = 0$ and $\Delta u_j^{(0)} = 0$. In the traditional Gauss–Seidel iteration sweeping from left to right, du_{j-1} is available for use at node j and du_{j+1} which is not available is set to zero, i.e., $du_{j+1} = 0$. In the current modified iteration, we approximate $du_{j+1} \simeq du_j$ and substituting it in Equation (5) we obtain

$$-r\left(\Delta u_{j-1}^{(1)}\right) + (1+2r)du_j - rdu_j = Res_j^n$$

or

$$(1+r)du_j = Res_j^n + r\Delta u_{j-1}^{(1)} \tag{6}$$

and

$$\Delta u_j^{(1)} = du_j$$

Therefore, Equation (6) can be rewritten as,

$$(1+r)\Delta u_j^{(1)} = Res_j^n + r\Delta u_{j-1}^{(1)} \tag{7}$$

2.5 Local Stability Analysis

To understand the local amplification of the solution from one time step to the next with this iteration scheme and to study the stability of the scheme, we seek the solution of the difference Equations in terms of Fourier modes. We look for the solution of Equation (7) of the form,

$$u_j^n = V^n\, e^{i\alpha j\Delta x} = V^n\, e^{ij\varsigma}, \quad \varsigma = \alpha\Delta x, \quad 0 \le \varsigma \le \pi$$

$$u_j^{\overline{n}} = V^{\overline{n}}\, e^{ij\varsigma}$$

$$\Delta u_j^{(1)} = u_j^{\overline{n}} - u_j^n = (V^{\overline{n}} - V^n)e^{ij\varsigma}$$

Substituting these into Equation (7) and cancelling common factor $e^{ij\varsigma}$ we obtain

$$(1+r)(V^{\overline{n}} - V^n) = r\left[V^n\, e^{-i\varsigma} - 2V^n + V^n\, e^{i\varsigma}\right] + r(V^{\overline{n}} - V^n)e^{-i\varsigma}$$

dividing by V^n,

$$(1+r)\left(\frac{V^{\overline{n}}}{V^n} - 1\right) = 2r(\cos\,\varsigma - 1) + r\left(\frac{V^{\overline{n}}}{V^n} - 1\right)e^{-i\varsigma}$$

The amplification factor $\overline{g}$ for the left to right iteration sweep is defined by

$$\overline{g} = \frac{V^{\overline{n}}}{V^n}$$

We can obtain $\overline{g}$ in the form

$$\overline{g} = \frac{(r\,\cos\,\varsigma - r + 1) + i\,r\,\sin\,\varsigma}{(1 + r - r\,\cos\,\varsigma) + i\,r\,\sin\,\varsigma}$$

2.6 Right to Left Sweep

For this case $m = 1$ and we approximate $du_{j-1} \simeq du_j$.

Substituting the above in Equation (5),

$$-r\left(\Delta u_{j-1}^{(1)} + du_j\right) + (1+2r)(\Delta u_j^{(1)} + du_j) - r\left(\Delta u_{j+1}^{(2)}\right) = Res_j^n$$

$$(1+r)du_j = Res_j^n + r\Delta u_{j-1}^{(1)} - (1+2r)\Delta u_j^{(1)} + r\left(\Delta u_{j+1}^{(2)}\right) \tag{8}$$

and in general

$$\Delta u_j^{(2)} = \Delta u_j^{(1)} + du_j$$

If the inner iteration is stopped with one symmetric iteration,

$$u_j^{n+1} = u_j^n + \Delta u_j^{(2)}$$

Rewrite Equation (8),

$$\begin{aligned}(1+r)\left(\Delta u_j^{(2)} - \Delta u_j^{(1)}\right) &= r\left(u_{j-1}^n - 2u_j^n + u_{j+1}^n\right) + r\Delta u_{j-1}^{(1)} \\ &\quad - (1+2r)\Delta u_j^{(1)} + r\Delta u_{j+1}^{(2)}\end{aligned}$$

or

$$\begin{aligned}(1+r)\left[\left(u_j^{n+1} - u_j^n\right)\right] &= r\left(u_{j-1}^n - 2u_j^n + u_{j+1}^n\right) \\ &\quad + r\left(u_{j-1}^{\overline{n}} - u_{j-1}^n\right) - (1+2r)\left(u_j^{\overline{n}} - u_j^n\right) + r\left(u_{j+1}^{n+1} - u_{j+1}^n\right)\end{aligned}$$

Simplifying,

$$(r+1)u_j^{n+1} - ru_{j+1}^{n+1} = ru_{j-1}^{\overline{n}} - ru_j^{\overline{n}} + u_j^n \tag{9}$$

2.7 Stability Analysis

As before substituting $u_j^n = V^n e^{ij\varsigma}$ in Equation (9), etc.,

$$(1+r)V^{n+1} - r\,V^{n+1}\,e^{i\varsigma} = r\,V^{\overline{n}}\,e^{-i\varsigma} - r\,V^{\overline{n}} + V^n$$

and dividing it by V^n,

$$g(1+r-r\,e^{i\varsigma}) = r\,\overline{g}(e^{-i\varsigma} - 1) + 1$$

where $g = V^{n+1}/V^n$ is the amplification factor for one time step. Eliminating $\overline{g}$ we can write

$$g = \frac{r\,\overline{g}[(\cos\,\varsigma - 1) - i\,\sin\,\varsigma] + 1}{[(1+r-r\,\cos\,\varsigma) - i\,r\,\sin\,\varsigma]}$$

We now consider the standard Gauss-Seidel scheme, along with its local stability and compare its merits and demerits with the modified Gauss-Seidel scheme. As before, considering a one-dimensional diffusion Equation and discretizing it according to standard Gauss-Seidel method, the Δu_{j+1}^{n+1} term in Equation(2) becomes zero in left to right sweep. Carrying out left to right sweep and then right to left sweep and analyzing the stability of the resulting Equation, the amplification factor g can be derived to be,

$$g = \frac{1 + \overline{g}(r \cos \varsigma - ir \sin \varsigma)}{(1 + 2r - r \cos \varsigma) + ir \sin \varsigma}$$

where

$$\overline{g} = \frac{(1 + r \cos \varsigma) + ir \sin \varsigma}{(1 + 2r - r \cos \varsigma) + ir \sin \varsigma}$$

The stability plot in terms of relationship between the magnitude of the amplification factor and ς for various r, for the modified Gauss-Seidel method is shown in Figure 1. As can be seen from the plot, there is an optimum range of r, namely, r ranging from 5 to 25 for which the method is most efficient in terms of its convergence. It is important to mention here that the value of r equals 10 was found to be the best when applied to three-dimensional diffusion Equation. The stability plot of the standard Gauss-Seidel scheme is shown in Figure 2, and as evident from the plot the stability is more or less insensitive to the value of r. In the subsequent sections of the chapter similar analysis are done for a three-dimensional diffusion Equation, the convergence rate of the two methods are compared and the concept of multigrid technique is introduced, which substantially accelerates the convergence rate.

Solving Equation(1) by modified Gauss-Seidel as well as standard Gauss-Seidel scheme is chosen to be a test problem. Random numbers have been chosen to be the initial solution and dirichlet boundary conditions have been applied on

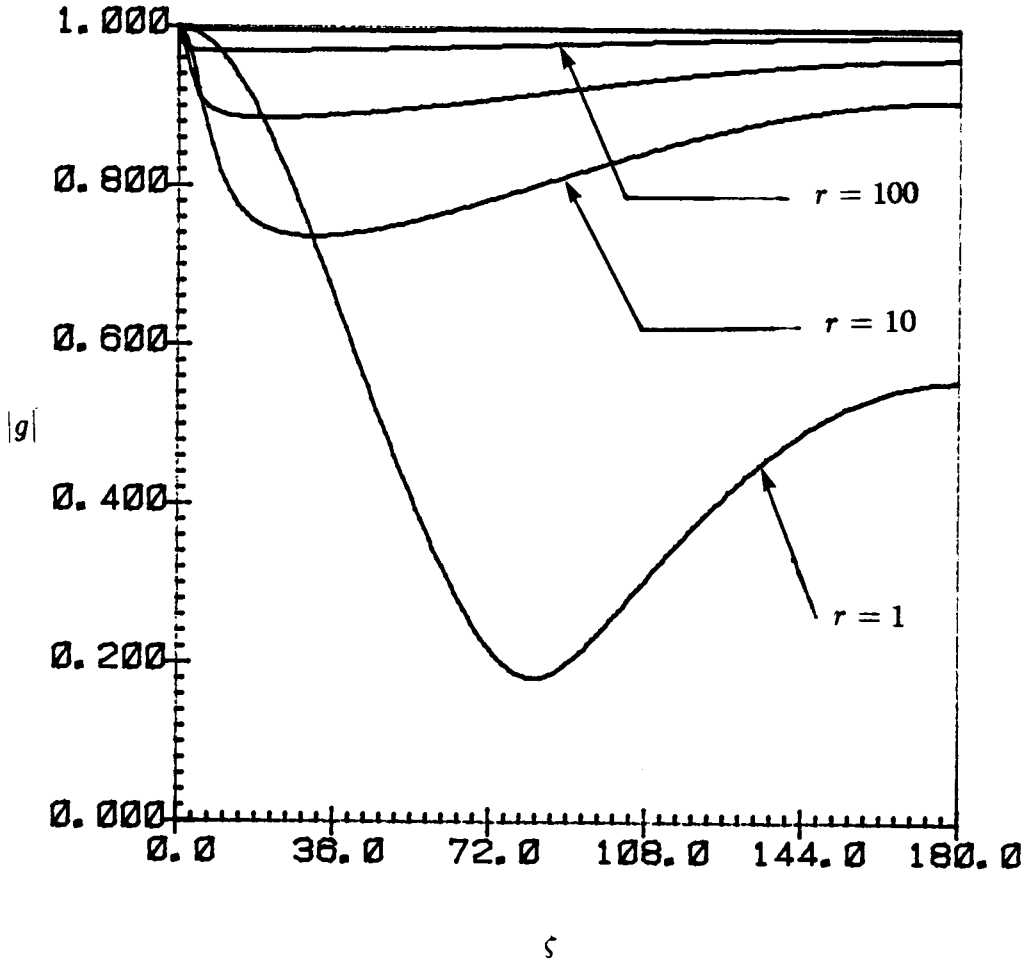


Figure 1: Stability Plot of Modified Gauss–Seidel Scheme
(1–D Diffusion Equation)

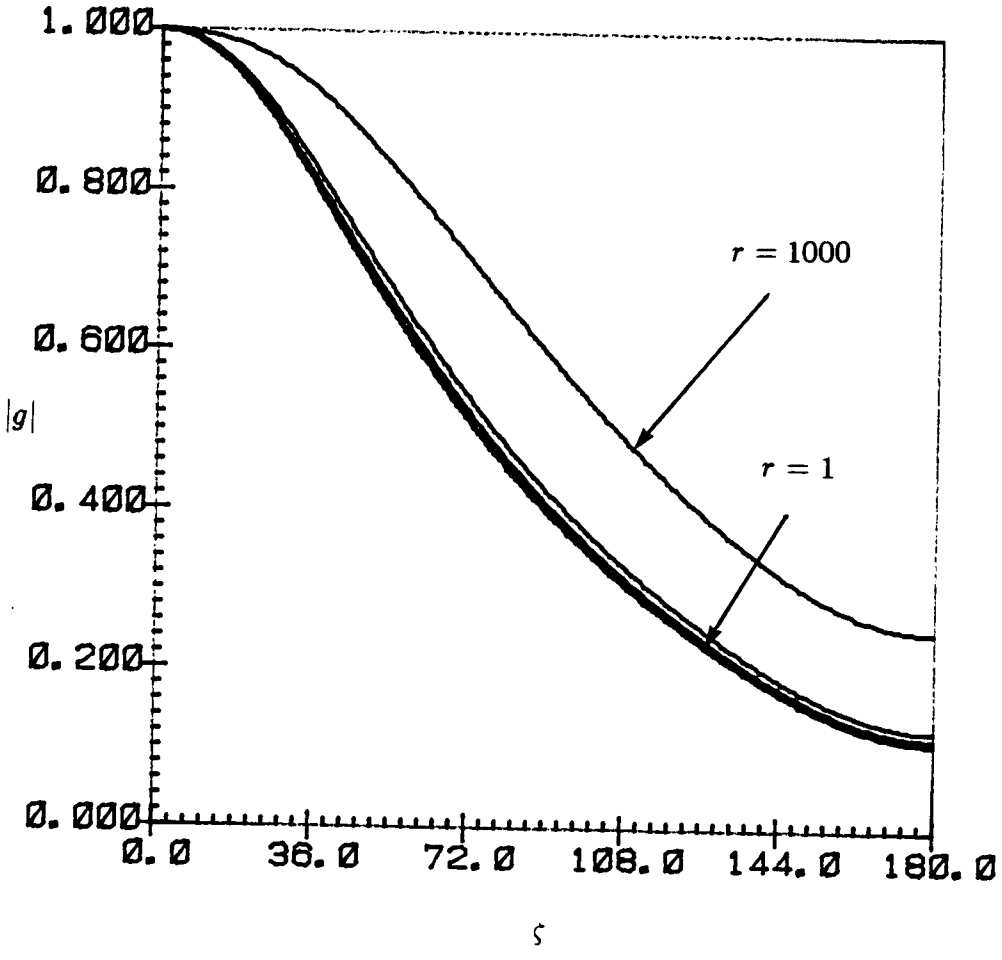


Figure 2: Stability Plot of Standard Gauss–Seidel Scheme
(1–D Diffusion Equation)

the boundary. The convergence history of the locally implicit scheme for one-dimensional diffusion Equation is shown in Figures 3, 4 and 5 where $\log(\Delta u)$ is plotted against number of work units. These three plots are for different grid sizes, namely, 10, 20 and 40 cells respectively. Throughout the text, one iteration in the finest grid level is defined as a work unit. This definition is useful when multigrid technique is introduced later in the chapter.

2.8 Three-Dimensional Diffusion Equation

We extend the previous technique to a three-dimensional diffusion Equation.

$$\frac{\partial u}{\partial t} = \nu \nabla^2 u \tag{10}$$

Central difference approximation for spatial derivatives and Euler implicit scheme for time integration give

$$\frac{u^{n+1}_{i,j,k} - u^{n}_{i,j,k}}{\Delta t} = \nu \left\{ \frac{u^{n+1}_{i-1,j,k} - 2u^{n+1}_{i,j,k} + u^{n+1}_{i+1,j,k}}{\Delta x^2} \right.$$

$$\left. + \frac{u^{n+1}_{i,j-1,k} - 2u^{n+1}_{i,j,k} + u^{n+1}_{i,j+1,k}}{\Delta y^2} + \frac{u^{n+1}_{i,j,k-1} - 2u^{n+1}_{i,j,k} + u^{n+1}_{i,j,k+1}}{\Delta z^2} \right\} \tag{11}$$

For simplicity we assume here,

$$\Delta x = \Delta y = \Delta z$$

Equation (11) is rewritten in delta form as follows

$$\Delta u^{n}_{i,j,k} = \frac{1}{1+6r}\, Res^{n}_{i,j,k} + \left(\frac{r}{1+6r}\right) \left(\Delta u^{n}_{i-1,j,k} + \Delta u^{n}_{i+1,j,k} + \Delta u^{n}_{i,j-1,k} \right.$$

$$\left. + \Delta u^{n}_{i,j+1,k} + \Delta u^{n}_{i,j,k-1} + \Delta u^{n}_{i,j,k+1} \right) \tag{13}$$

where

$$r = \nu \frac{\Delta t}{\Delta x^2}$$

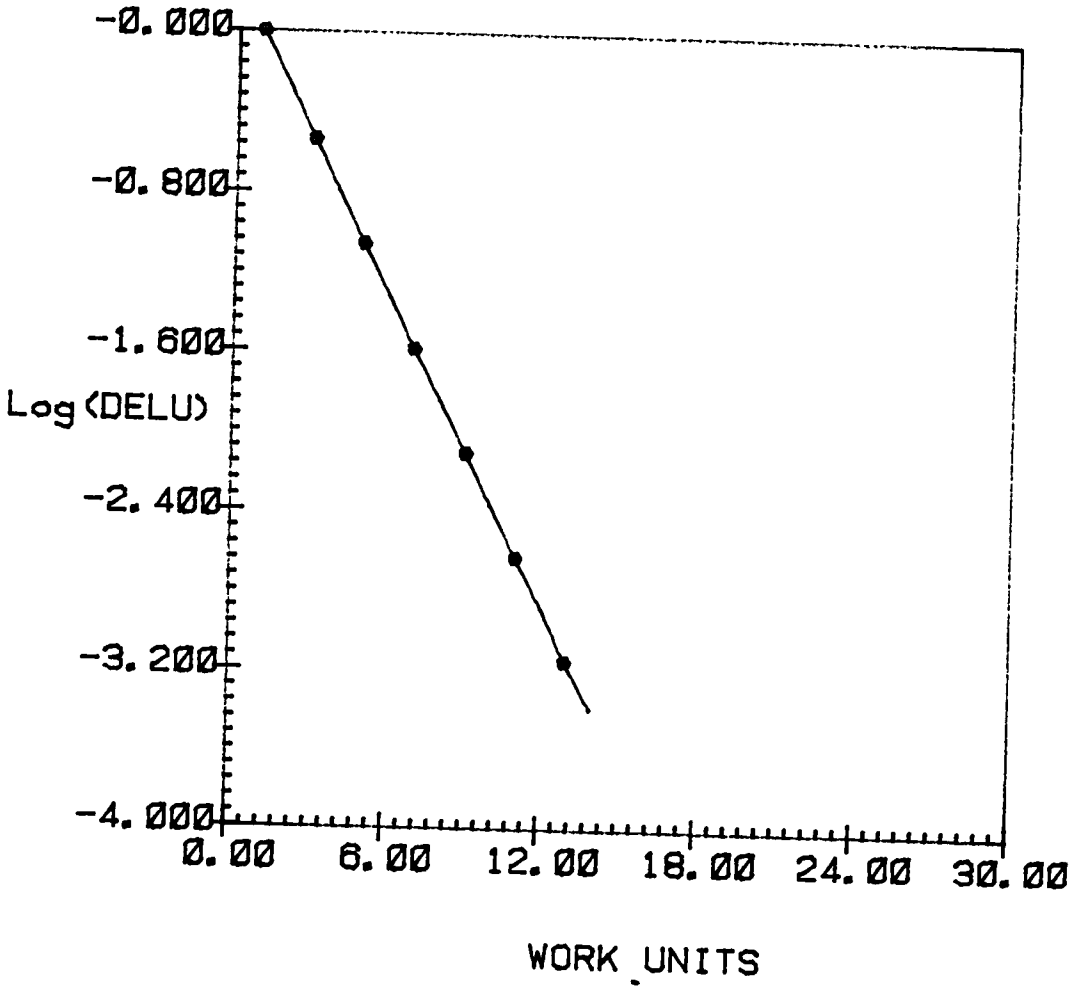


Figure 3: Convergence History of Locally Implicit Scheme
(1-D Diffusion Equation, 10 cells)

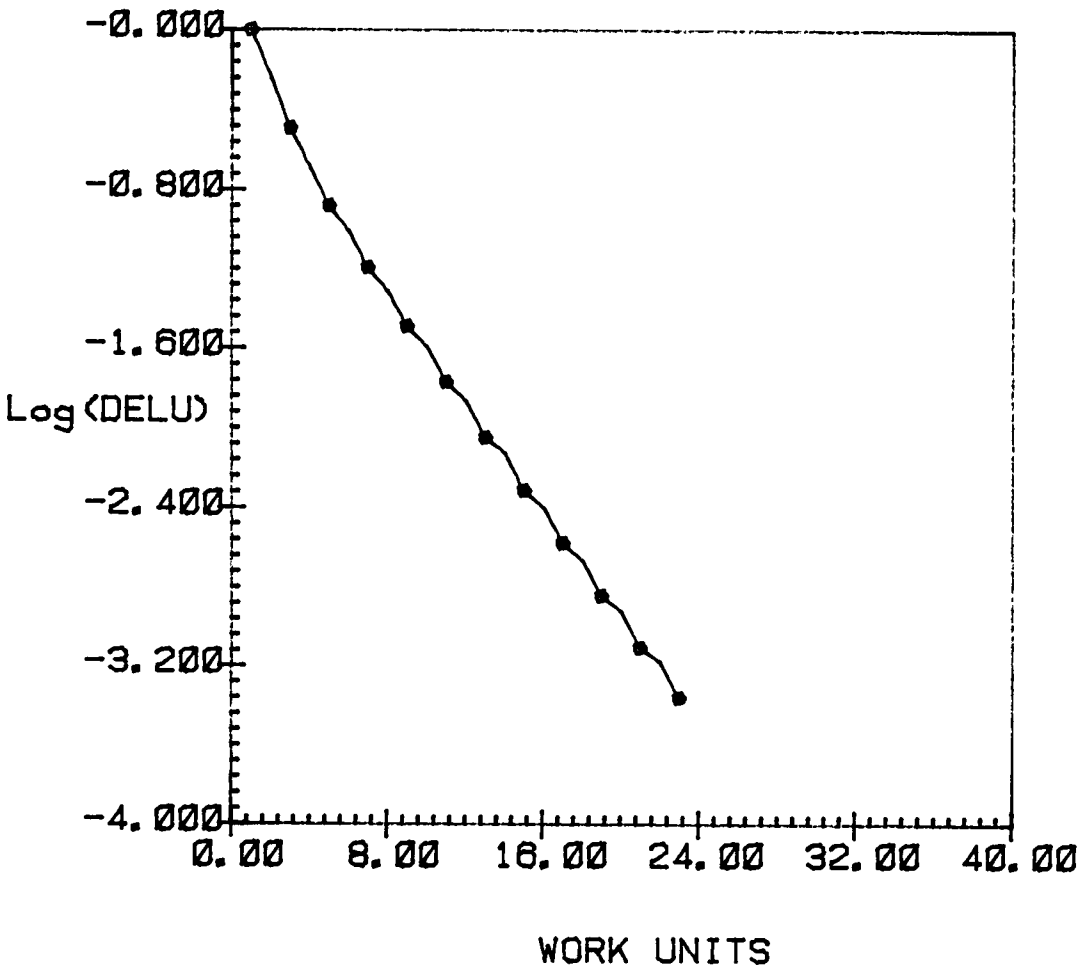


Figure 4: Convergence History of Locally Implicit Scheme
(1–D Diffusion Equation, 20 cells)

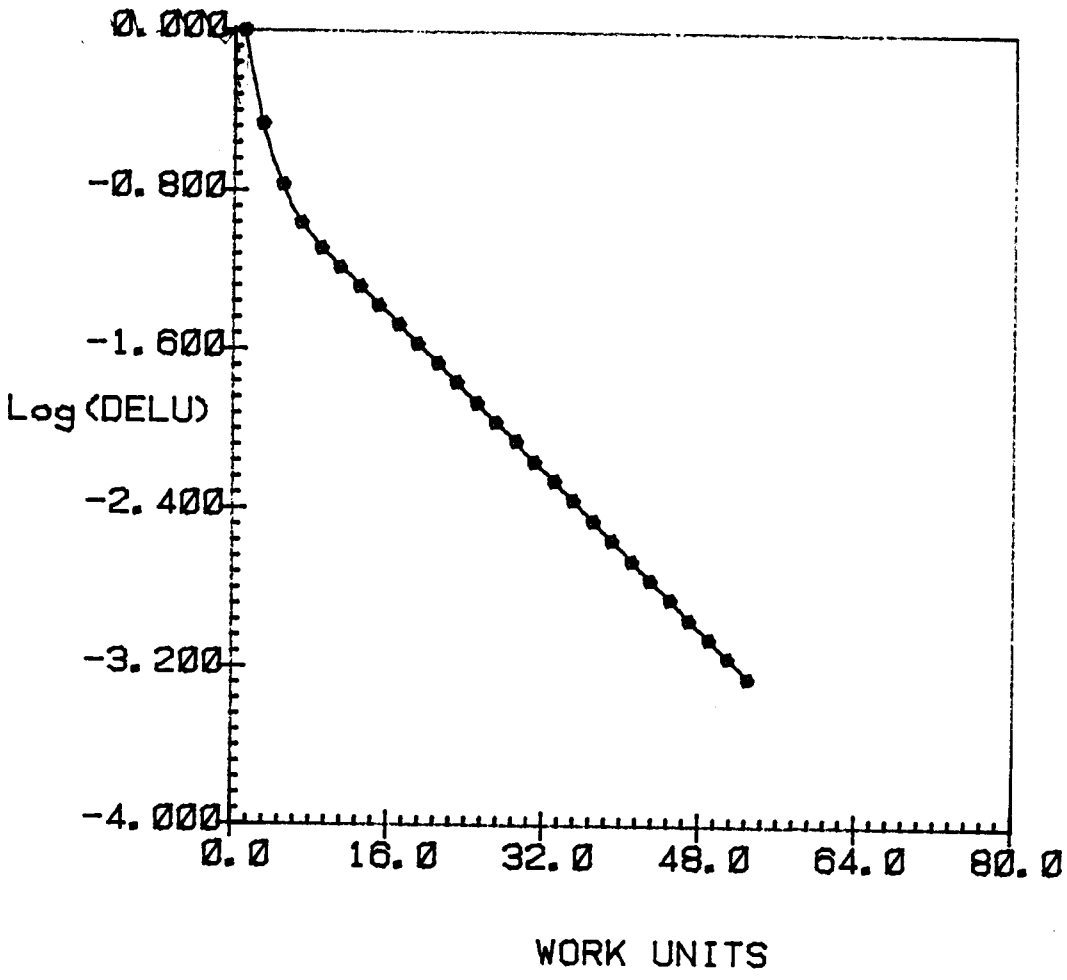


Figure 5: Convergence History of Locally Implicit Scheme
(1-D Diffusion Equation, 40 cells)

and

$$Res^n_{i,j,k} = r\left\{\left(u^n_{i-1,j,k} - 2u^n_{i,j,k} + u^n_{i+1,j,k}\right) + \left(u^n_{i,j-1,k} - 2u^n_{i,j,k} + u^n_{i,j+1,k}\right) + \left(u^n_{i,j,k-1} - 2u^n_{i,j,k} + u^n_{i,j,k+1}\right)\right\}$$

2.9 Inner Iteration

$$\Delta u^{(m+1)}_{i,j,k} = \Delta u^{(m)}_{i,j,k} + du_{i,j,k}$$

$$\Delta u^{(0)}_{i,j,k} = 0$$

Equation (13) becomes,

$$\left(\Delta u^{(m)}_{i,j,k} + du_{i,j,k}\right) = \frac{1}{(1+6r)}\, Res^n_{i,j,k} + \left(\frac{r}{1+6r}\right)\left\{\left(\Delta u^{(m)}_{i-1,j,k} + du_{i-1,j,k}\right) + \left(\Delta u^{(m)}_{i,j-1,k} + du_{i,j-1,k}\right) + \left(\Delta u^{(m)}_{i,j,k-1} + du_{i,j,k-1}\right) + \left(\Delta u^{(m)}_{i+1,j,k} + du_{i+1,j,k}\right) + \left(\Delta u^{(m)}_{i,j+1,k} + du_{i,j+1,k}\right) + \left(\Delta u^{(m)}_{i,j,k+1} + du_{i,j,k+1}\right)\right\} \quad (14)$$

Being a three-dimensional problem the symmetric sweeping would involve up to eight sweeps.

Sweep (1)

If the first iteration sweep starts at the corner (1, 1, 1) of the computational domain, modified Gauss–Seidel iteration introduces the approximation,

$$du_{i+1,j,k} = du_{i,j+1,k} = du_{i,j,k+1} = du_{i,j,k}$$

Substituting the above in Equation (14),

$$\left(\Delta u^m_{i,j,k} + du_{i,j,k}\right) = \frac{1}{(1+6r)}\, Res^n_{i,j,k} + \left(\frac{r}{1+6r}\right)\left\{\Delta u^{(m+1)}_{i-1,j,k} + \Delta u^{(m+1)}_{i,j-1,k} + \Delta u^{(m+1)}_{i,j,k-1}\right\} + \left(\frac{3r}{1+6r}\right) du_{i,j,k} + \left(\frac{r}{1+6r}\right)\left(\Delta u^{(m)}_{i+1,j,k} + \Delta u^{(m)}_{i,j+1,k} + \Delta u^{(m)}_{i,j,k+1}\right)$$

In the first sweep, $m = 0$, $\Delta u^{(0)}_{i,j,k} = 0$ and we obtain $\Delta u^{(1)}_{i,j,k} = du_{i,j,k}$ as

$$du_{i,j,k} = \frac{1}{(1+3r)}\, Res^{n}_{i,j,k} + \left(\frac{r}{1+3r}\right)\left\{\Delta u^{(m+1)}_{i-1,j,k} + \Delta u^{(m+1)}_{i,j-1,k} + \Delta u^{(m+1)}_{i,j,k-1} + \Delta u^{(m)}_{i+1,j,k} + \Delta u^{(m)}_{i,j+1,k} + \Delta u^{(m)}_{i,j,k+1}\right\} - \frac{(1+6r)}{(1+3r)}\Delta u^{(m)}_{i,j,k} \qquad (15)$$

Similar Equations can be written for the other seven sweeps, starting at the other corners of the computational domain.

2.10 Stability Analysis of 3–D Diffusion Equation

The local stability analysis of the 3–D diffusion Equation can be carried out very similar to that of the 1–D diffusion Equation. We look for the solution of the Equation (15) of the form,

$$u^{n}_{j,k,l} = V^{n} e^{i(j\xi + k\eta + l\varsigma)} \qquad (16)$$

where,

$$i = \sqrt{-1}$$

$$\xi = \alpha \Delta x$$

$$\eta = \beta \Delta y$$

$$\varsigma = \gamma \Delta z$$

It is clear from Equation (16) that for the inner iteration,

$$\Delta u^{(m)} = \Delta V^{(m)} e^{i(j\xi + k\eta + l\varsigma)}$$

also

$$du_{j,k,l} = \Delta u^{(m+1)}_{j,k,l} - \Delta u^{(m)}_{j,k,l}, \quad m = 0, 1, - - - - 7$$

Finally the solution is updated by,

$$u^{n+1}_{j,k,l} = u^{n}_{j,k,l} + \Delta u^{(8)}_{j,k,l} = V^{n+1} e^{i(j\xi + k\eta + i\varsigma)}$$

Sweep 1: Node (j,k,l)

Rewriting Equation (15),

$$(1+3r)du_{j,k,l} = h(\xi,\eta,\varsigma) + r\left(\Delta u^{(m+1)}_{j-1,k,l} + \Delta u^{(m+1)}_{j,k-1,l} + \Delta u^{(m+1)}_{j,k,l-1} + \Delta u^{(m)}_{j+1,k,l} + \Delta u^{(m)}_{j,k+1,l} + \Delta u^{(m)}_{j,k,l+1}\right) - (1+6r)\Delta u^{(m)}_{j,k,l} \qquad (17)$$

where,

$$h(\xi,\eta,\varsigma) = Res^n_{j,k,l} = r\left\{(u^n_{j-1,k,l} - 2u^n_{j,k,l} + u^n_{j+1,k,l}) + (u^n_{j,k-1,l} - 2u^n_{j,k,l} + u^n_{j,k+1,l}) + (u^n_{j,k,l-1} - 2u^n_{j,k,l} + u_{j,k,l+1})\right\}$$
$$= 2r(\cos\xi + \cos\eta + \cos\varsigma - 3) \qquad (18)$$

For this sweep $m = 0$ and $\Delta V^{(0)} = 0$. Equation (17) can be rewritten as,

$$(1+3r)(\Delta V^{(m+1)} - \Delta V^{(m)}) = V^n h(\xi,\eta,\varsigma) + r\left\{\Delta V^{(m+1)}(e^{-i\xi} + e^{-i\eta} + e^{-i\varsigma}) + \Delta V^{(m)}(e^{i\xi} + e^{i\eta} + e^{i\varsigma})\right\} - (1+6r)\Delta V^{(m)}$$

or

$$(1+3r)\Delta V^{(1)} = V^n h(\xi,\eta,\varsigma) + r\Delta V^{(1)}(e^{-i\xi} + e^{-i\eta} + e^{-i\varsigma})$$
$$\Delta V^{(1)} = \frac{h(\xi,\eta,\varsigma)}{f_1(\xi,\eta,\varsigma)} V^n = g^{(1)} V^n$$

where

$$f_1 = \left\{(1+3r) - r\left(e^{-i\xi} + e^{-i\eta} + e^{-i\varsigma}\right)\right\}$$

$g^{(1)} =$ amplification factor for sweep 1

Sweep 2: $m = 1$, node $(j+1,\ k+1,\ l+1)$

$$(1+3r)\left(\Delta V^{(m+1)} - \Delta V^{(m)}\right) = V^n h(\xi,\eta,\varsigma) + r\left\{\Delta V^{(m+1)}\left(e^{i\xi} + e^{i\eta} + e^{i\varsigma}\right) + \Delta V^{(m)}\left(e^{-i\xi} + e^{-i\eta} + e^{-i\varsigma}\right)\right\} - (1+6r)\Delta V^{(m)}$$

or

$$(1+3r)\left(\Delta V^{(2)} - \Delta V^{(1)}\right) = V^n h(\xi,\eta,\varsigma) + r\left\{\Delta V^{(2)}\left(e^{i\xi} + e^{i\eta} + e^{i\varsigma}\right) + \Delta V^{(1)}\left(e^{-i\xi} + e^{-i\eta} + e^{-i\varsigma}\right)\right\} - (1+6r)\Delta V^{(1)}$$

$$\Delta V^{(2)} = \frac{V^n h(\xi,\eta,\varsigma) + e_2(\xi,\eta,\varsigma)\Delta V^{(1)}}{f_2(\xi,\eta,\varsigma)}$$

$$= \frac{V^n h + e_2 g^{(1)} v^n}{f_2} = g^{(2)} v^n$$

where

$$g^{(2)} = \frac{h + e_2 g^{(1)}}{f_2}$$

$$e_2 = r\left(e^{-i\xi} + e^{-i\eta} + e^{-i\varsigma}\right) - 3r$$

$$f_2 = (1 + 3r) - r\left(e^{i\xi} + e^{i\eta} + e^{i\varsigma}\right)$$

Noticing the pattern, we can write down the Equations for the other nodes as follows

$$\Delta V^{(p)} = \frac{V^n h(\xi,\eta,\varsigma) + e_p(\xi,\eta,\varsigma) g^{(p-1)} V^n}{f_p(\xi,\eta,\varsigma)}$$

$$= g^{(p)} V^n, \quad p = 3, 4, - - - - -, 8$$

where

$$g^{(p)} = \frac{h(\xi,\eta,\varsigma) + e_p(\xi,\eta,\varsigma) g^{(p-1)}}{f_p(\xi,\eta,\varsigma)}$$

e_p and f_p are similar to e_2 and f_2 and depend on the starting node of the inner iteration sweep. The overall amplification factor g is determined as follows

$$V^{n+1} = V^n + \Delta V^{(8)} = V^n + g^{(8)} V^n$$

$$= V^n (1 + g^{(8)})$$

$$g = \frac{V^{n+1}}{V^n} = 1 + g^{(8)}$$

The stability plot of the 3–D diffusion Equation is shown in Figure 6. Table 1 shows maximum and minimum values of $|g|$ over the entire range of $(\xi,\eta,\varsigma)$for various values of r and proves that the scheme is unconditionally stable. The domain of (ξ,η,ς) is $0 \leq \xi \leq \pi$, $0 \leq \eta \leq \pi$ and $0 \leq \varsigma \leq \pi$.

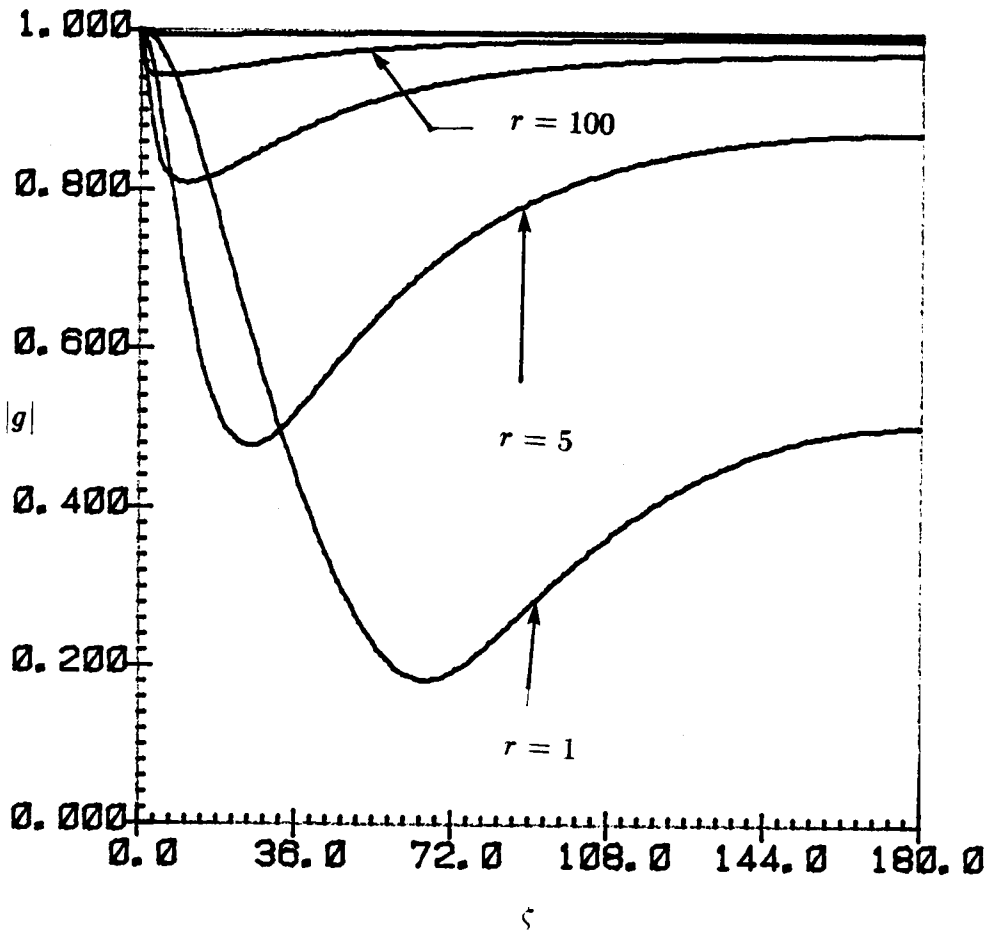


Figure 6: Stability Plot of Modified Gauss–Seidel Scheme
(3–D Diffusion Equation)

Table 1. Stability of 3–D Diffusion Equation

r	Min $\|g\|$	Max $\|g\|$
1	0.1829	1.0
5	0.4787	1.0
10	0.6365	1.0
25	0.8094	1.0
50	0.8938	1.0
100	0.9436	1.0

2.11 Multigrid Acceleration Technique

The multigrid technique adapted here for the elliptic problem is slightly different from the multigrid technique used for Navier–Stokes Equations which will be described later. The general multigrid algorithm represents one of the most powerful methods of accelerating convergence of a numerical solution to its steady state. The basic idea of a multigrid strategy is to perform time steps on coarser meshes to calculate corrections to a solution on a finer mesh. The advantages of time stepping on coarse meshes are two fold: first, the low frequency oscillations are removed much faster on coarser grids, and secondly the work involved is much less because of the smaller number of grid points in coarser grids.

2.12 Injection Cycle

From a chosen fine grid, a sequence of coarser grids is formed by combining groups of adjacent cells in a systematic manner. At a grid level denoted by a mesh

parameter h, suppose the delta form of the Equations to be solved is denoted by

$$L_h(\Delta u) = Res_h(u) + FF_h \tag{19}$$

where FF_h is a forcing function, which is zero for the finest grid level. One or more time step calculations are carried out at this grid level and the solution is updated. Then the solution u_h is injected to a coarser grid level denoted by the subscript $2h$. The type of injection of the solution and residues is problem oriented. For the diffusion Equation which is discretized by using a finite difference method, the following injection strategy is employed. Since the fine and coarse meshes have common points,

$$u_{2h} = u_h$$

at all the points common to both grids. The residual is injected in such a way that equal weightages are given to all the surrounding nodes. For example in a three-dimensional problem the current node (i, j, k) will get $1/7$ the weightage of its own residual and $1/7$ the weightage of all the surrounding nodes.

$$(Res_{2h})_{i,j,k} = \frac{1}{7}\Big\{(Res_h)_{i,j,k} + (Res_h)_{i-1,j,k} + (Res_h)_{i+1,j,k} + (Res_h)_{i,j-1,k} + (Res_h)_{i,j+1,k} + (Res_h)_{i,j,k-1} + (Res_h)_{i,j,k+1}\Big\}$$

Now a forcing function FF is defined in such a way that,

$$FF_{2h} = Res_{2h} - Res(u_{2h}) \tag{20}$$

where $Res(u_{2h})$ is the residue computed after injection. Then the following Equation is solved on the coarse grid,

$$L_{2h}\Delta u = Res(u_{2h}) + FF_{2h} \tag{21}$$

This is done for one or more time steps by the locally implicit scheme outlined earlier. This process is repeated until the coarsest grid solution is computed.

2.13 Interpolation Cycle

The coarse grid information is passed back to the finer grids as follows. The difference in the updated solution and the injected solution on the coarse grid is interpolated in computational space at the grid points of the next finer mesh and is added to the solution which is previously computed at that level. That is,

$$(u_h)_{\text{new}} = (u_h)_{\text{old}} + I_{2h}^{h}\left(u_{2h} - I_{h}^{2h}(u_h)_{\text{old}}\right) \tag{22}$$

where I_{2h}^{h} represents interpolation operator from coarse to fine grid etc. we now consider two options: one is to interpolate to the next finer grid level and secondly to carry out relaxation at the intermediate grid levels. The later has been chosen for the present case.

In order to understand the convergence process, several computer runs were made with varying number of sweeps for the three-dimensional problem, namely, 2, 4 and 8 sweeps. Figure 7 shows the convergence history for both single and multigrids for the case with 4 sweeps. Figure 8 is a similar plot with 8 sweeps. As can be seen, the locally implicit scheme converges faster than the Gauss–Seidel scheme and works faster as the number of sweeps increase, where as the standard Gauss–Seidel scheme is more or less insensitive to the number of sweeps.

Similar analysis have been made by Ratcliff [45] for a convection dominated flow, Reddy and Jacocks [15] for Euler Equations and the present scheme has been shown to converge for convection dominated flows where as the standard Gauss-Seidel scheme is unstable. Having established the scheme, the method has been applied to thin layer Navier–Stokes Equations where convection and diffusion are equally important.

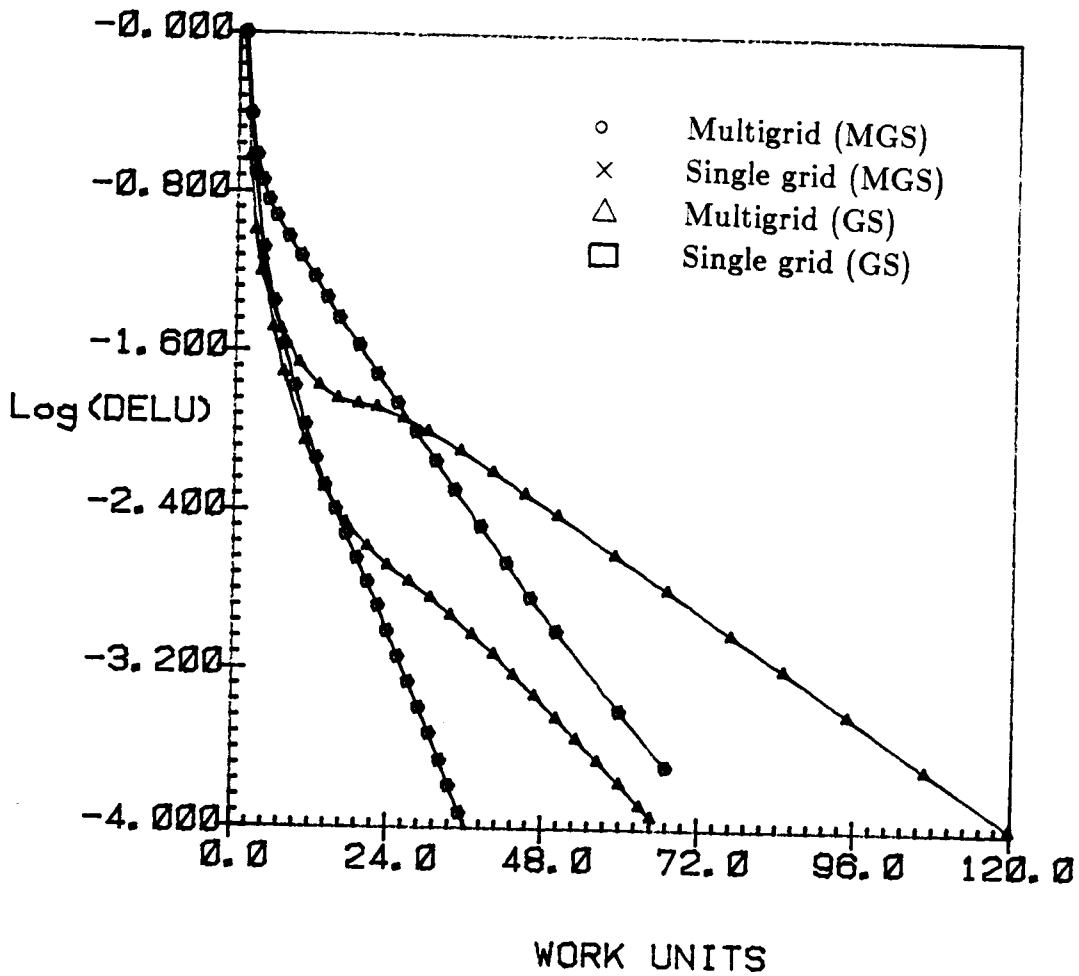


Figure 7: Comparison of Convergence of Modified and Standard Gauss-Seidel Schemes in Both Single and Multigrid (3-D Diffusion Equation, 4 Sweeps)

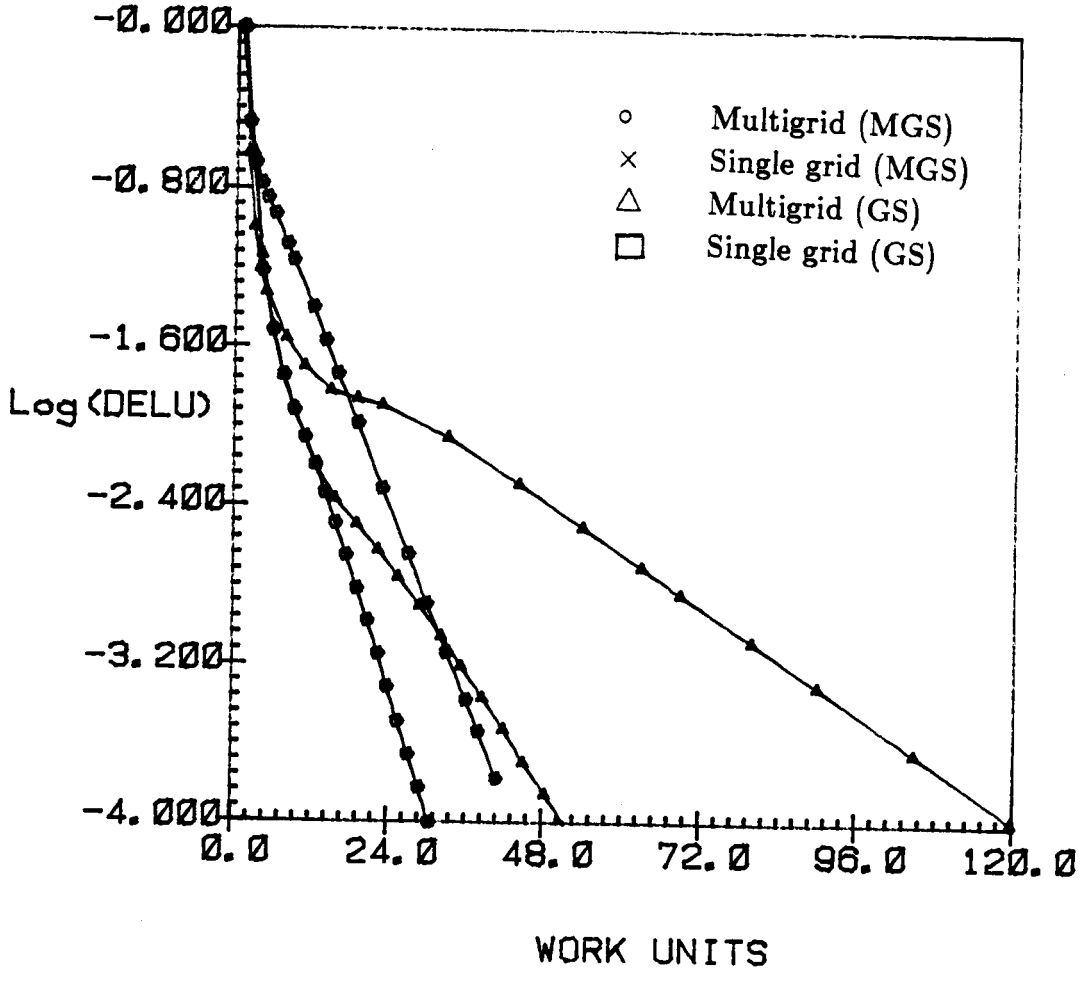


Figure 8: Comparison of Convergence of Modified and Standard Gauss–Seidel Schemes in Both Single and Multigrid (3–D Diffusion Equation, 8 Sweeps)

CHAPTER 3

A LOCALLY IMPLICIT SCHEME FOR NAVIER–STOKES EQUATIONS

A locally implicit method for solving the Navier–Stokes Equations is described in this chapter. The technique demonstrated for a model Equation in Chapter 2 is extended to the Navier–Stokes Equations. The method uses finite volume spatial discretization, local time integration and Jameson–type artificial dissipation terms. The Navier–Stokes Equations are written in a general curvilinear form and the linearized Equations are solved by a modified Gauss–Seidel inner iteration.

The multigrid scheme which significantly enhances the convergence rate for the Navier–Stokes Equations is different from that of the model Equation and it is described in section 3.10. Several test cases have been run and the numerical results are compared with the available experimental results [23]. Flow fields around two different airfoils have been computed. For laminar, low Reynolds number and subsonic flows an NACA0012 airfoil with C-grid has been used. Subsonic and transonic flows have been computed around a RAE2822 airfoil at high Reynolds numbers with turbulence. An elliptic grid generator has been used for the NACA0012 airfoil and a hyperbolic grid generator is used for RAE2822 which requires a highly stretched grid. The grid size used in all these applications is 128 x 32.

The Navier–Stokes Equations written in generalized curvilinear coordinates in non-dimensional form are

$$\partial_\tau \hat{Q} + \partial_\xi \hat{E} + \partial_\eta \hat{F} = Re^{-1}[\partial_\xi \hat{E}_v + \partial_\eta \hat{F}_v] \tag{23}$$

where

$$\widehat{Q} = J^{-1}\begin{bmatrix}\rho\\ \rho u\\ \rho v\\ e\end{bmatrix}\ ,\ \widehat{E} = J^{-1}\begin{bmatrix}\rho U\\ \rho uU + \xi_x p\\ \rho vU + \xi_y p\\ U(e+p) - \xi_t p\end{bmatrix}\ ,\ \widehat{F} = J^{-1}\begin{bmatrix}\rho V\\ \rho uV + \eta_x p\\ \rho vV + \eta_y p\\ V(e+p) - \eta_t p\end{bmatrix}$$

with

$$U = \xi_t + \xi_x u + \xi_y v \tag{24}$$

$$V = \eta_t + \eta_x u + \eta_y v \tag{25}$$

the contravariant velocities. The viscous flux terms are

$$\widehat{E}_v = J^{-1}(\xi_x E_v + \xi_y F_v) \tag{26}$$

and

$$\widehat{F}_v = J^{-1}(\eta_x E_v + \eta_y F_v) \tag{27}$$

where,

$$E_v = \begin{bmatrix}0\\ \tau_{xx}\\ \tau_{xy}\\ f_4\end{bmatrix}\ ,\ F_v = \begin{bmatrix}0\\ \tau_{xy}\\ \tau_{yy}\\ g_4\end{bmatrix}$$

$$J^{-1} = (x_\xi y_\eta - x_\eta y_\xi)$$

and

$$\xi_x = J y_\eta, \quad \xi_y = -J x_\eta$$

$$\eta_x = -J y_\eta, \quad \eta_y = J x_\xi$$

The stress terms, such as τ_{xx} are also transformed in terms of the ξ and η derivatives where

$$\tau_{xx} = \mu(4(\xi_x u_\xi + \eta_x u_\eta) - 2(\xi_y v_\xi + \eta_y v_\eta))/3 \tag{28}$$

$$\tau_{xy} = \mu(\xi_y u_\xi + \eta_y u_\eta + \xi_x v_\xi + \eta_x v_\eta) \tag{29}$$

$$\tau_{yy} = \mu(-2(\xi_x u_\xi + \eta_x u_\eta) + 4(\xi_y v_\xi + \eta_y v_\eta))/3 \tag{30}$$

$$f_4 = u\tau_{xx} + v\tau_{xy} + \mu pr^{-1}(\gamma - 1)^{-1}(\xi_x \partial_\xi a^2 + \eta_x \partial_\eta a^2) \tag{31}$$

$$g_4 = u\tau_{xy} + v\tau_{yy} + \mu pi^{-1}(\gamma - 1)^{-1}(\xi_y \partial_\xi a^2 + \eta_y \partial_\eta a^2) \tag{32}$$

with terms such as u_x expanded by chain rule.

Re is Reynolds number and *Pr* is the Prandtl number. Velocities are non-dimensionalized by the free stream speed of sound a_∞, density by free stream density ρ_∞, and e and p by $\rho_\infty a_\infty^2$ and viscosity by μ_∞.

3.1 Thin-Layer Approximation

In high Reynolds number viscous flows the effects of viscosity are concentrated near rigid boundaries and in the wake regions. Typically in computations we have only enough grid points available (due to computer storage limits) to concentrate grid lines near the rigid surfaces. The resulting grid systems usually have fine grid spacing in directions nearly normal to the surfaces and coarse grid spacing along the surface.

Even if we numerically solve the full Navier–Stokes Equations, the viscous terms associated with derivatives along the body may not be resolved. For attached and mildly separated flows these terms are negligible while the terms in the near normal direction are the substantial terms.

The thin layer approximation requires that:

1. All body surfaces be mapped onto coordinate surfaces. Specifically, η = constant coordinate surfaces, see Figure 9 .

2. Grid spacing is clustered to the body surfaces such that sufficient resolution

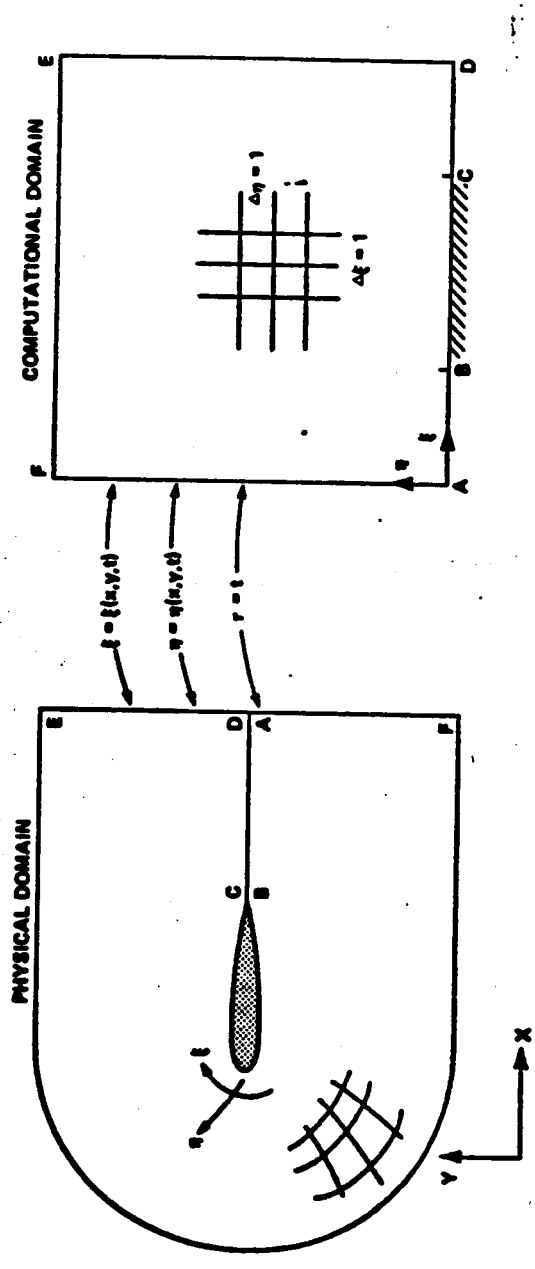


Figure 9: Schematic of Coordinate Transformation

for a particular Reynolds number is obtained. (At least one or two points in the sublayer.)

3. All the viscous derivatives in the ξ direction are neglected, while the terms in the η direction are retained.

The thin layer approximation is similar in philosophy but not the same as the boundary layer approximation. The normal momentum Equation is solved and the computed pressure can vary through the boundary layer.

The thin layer approximation can break down for low Reynolds numbers and in regions of massive flow separation. It is not a necessary step in the development of the Equations and numerical algorithm. The full Navier–Stokes Equations can be solved if sufficient resolution is provided and is warranted by the physical situation.

3.2 Thin–Layer Equations

$$\partial_\tau \hat{Q} + \partial_\xi \hat{E} + \partial_\eta \hat{F} = Re^{-1} \partial_\eta \hat{S} \tag{33}$$

where

$$\hat{Q} = J^{-1} Q \tag{34}$$

$$\hat{S} = J^{-1} \begin{bmatrix} 0 \\ (\eta_x m_1 + \eta_y m_2) \\ (\eta_x m_2 + \eta_y m_3) \\ \eta_x(um_1 + vm_2 + m_4) + \eta_y(um_2 + vm_3 + m_5) \end{bmatrix}$$

and

$$m_1 = \frac{\mu}{3}(4\eta_x u_\eta - 2\eta_y v_\eta) \tag{35}$$

$$m_2 = \mu(\eta_y u_\eta + \eta_x v_\eta) \tag{36}$$

$$m_3 = \frac{\mu}{3}(-2\eta_x u_\eta + 4\eta_y v_\eta) \tag{37}$$

$$m_4 = \frac{\mu}{pr(\gamma-1)}\eta_x \partial_\eta(a^2) \tag{38}$$

$$m_5 = \frac{\mu}{pr(\gamma-1)}\eta_y \partial_\eta(a^2) \tag{39}$$

It is convenient to rewrite $\widehat{E}$ and $\widehat{F}$ in terms of E and F as defined below.

$$\widehat{E} = J^{-1}(\xi_x E + \xi_y F) \tag{40}$$

$$\widehat{F} = J^{-1}(\eta_x E + \eta_y F) \tag{41}$$

where

$$E = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(e+p) \end{bmatrix}, \quad F = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(e+p) \end{bmatrix}$$

The thin-layer Equations are rewritten as,

$$\frac{\partial}{\partial \tau}(J^{-1}Q) + \frac{\partial}{\partial \xi}(J^{-1}\xi_x E + J^{-1}\xi_y F) + \frac{\partial}{\partial \eta}(J^{-1}\eta_x E + J^{-1}\eta_y F) - Re^{-1}\frac{\partial}{\partial \eta}\widehat{S} = 0 \tag{42}$$

3.3 Finite Volume Approach

Integrating the thin-layer Navier-Stokes Equations over a unit square domain Ω in the computational plane (ξ, η) bounded by $\partial\Omega$ we get,

$$\int\int_{\Omega} \frac{\partial}{\partial \tau}(J^{-1}Q) + \int\int_{\Omega} \frac{\partial}{\partial \xi}(J^{-1}\xi_x E + J^{-1}\xi_y F) + \frac{\partial}{\partial \eta}(J^{-1}\eta_x E + J^{-1}\eta_y F)$$

$$-Re^{-1}\int\int_{\Omega} \frac{\partial}{\partial \eta}\widehat{S} = 0$$

Rewriting the above Equation,

$$\int\int_{\Omega} \frac{\partial}{\partial \tau}(J^{-1}Q) + \int\int_{\Omega} \frac{\partial}{\partial \xi}(y_\eta E - x_\eta F) + \frac{\partial}{\partial \eta}(-y_\xi E + x_\xi F)$$

$$-Re^{-1}\int\int_{\Omega} \frac{\partial}{\partial \eta}\widehat{S} = 0 \tag{43}$$

Denote

$$K = y_\eta E - x_\eta F \tag{44}$$

$$L = -y_\xi E + x_\xi F \tag{45}$$

substituting the above in Equation (43) and applying Green's theorem and introducing artifical dissipation, we have

$$J^{-1}\int\int_{\Omega} \frac{\partial Q}{\partial \tau} d\xi d\eta + \int_{\partial\Omega} (K, L).d\vec{s}$$

$$-Re^{-1}\int_{\partial\Omega} (o, \widehat{S}).d\vec{s} = D \tag{46}$$

where D is the artificial dissipation added to suppress the nonlinear instabilities. Normally, D is less than the order of the truncation error. Discretizing Equation (46) and assuming $\Delta\xi = \Delta\eta = 1$ in the computational domain,

$$\frac{J^{-1}_{j,k}}{\Delta\tau}\Delta Q_{j,k} + (y_\eta E - x_\eta F)\Big|^{j+\frac{1}{2},k}_{j-\frac{1}{2},k} + (-y_\xi E + x_\xi F)\Big|^{j,k+\frac{1}{2}}_{j,k-\frac{1}{2}}$$

$$-Re^{-1}(\widehat{S})\Big|^{j,k+\frac{1}{2}}_{j,k-\frac{1}{2}} - D = 0 \tag{47}$$

3.4 Euler Implicit Scheme

Writing Euler implicit scheme for time discretization,

$$\frac{J_{j,k}^{-1}}{\Delta\tau}\Delta Q_{j,k}^{n} + (y_\eta E^{n+1} - x_\eta F^{n+1})_{j+\frac{1}{2},k} - (y_\eta E^{n+1} - x_\eta F^{n+1})_{j-\frac{1}{2},k}$$

$$+(-y_\xi E^{n+1} + x_\xi F^{n+1})_{j,k+\frac{1}{2}} - (-y_\xi E^{n+1} + x_\xi F^{n+1})_{j,k-\frac{1}{2}}$$

$$-Re^{-1}(\widehat{S}^{n+1})_{j,k+\frac{1}{2}} + Re^{-1}(\widehat{S}^{n+1})_{j,k-\frac{1}{2}} - D_{j,k}(Q^{n+1}) = 0 \qquad (48)$$

3.5 Approximation To The $\widehat{S}^{n+1}$ term

$$\widehat{S}^{n+1} = \left[\widehat{S}(\widehat{Q},\widehat{Q}\eta)\right]^{n+1} = \widehat{S}(\widehat{Q}^{n+1},\widehat{Q}_\eta^{n+1})$$

$$= \widehat{S}(\widehat{Q}^{n} + \Delta\widehat{Q}^{n},\widehat{Q}_\eta^{n} + \Delta\widehat{Q}_\eta^{n})$$

$$S^{n+1} = \widehat{S}(\widehat{Q}^{n},\widehat{Q}_\eta^{n}) + \widehat{M}^{n}\Delta\widehat{Q}^{n} + \overline{M}^{n}\Delta Q_\eta^{n} \qquad (49)$$

where

$$\widehat{M} = \partial\widehat{S}/\partial\widehat{Q} \text{ and } \overline{M} = \partial\widehat{S}/\partial\widehat{Q}_\eta$$

But,

$$\overline{M}\Delta Q_\eta^{n} = (\overline{M}\Delta\widehat{Q})_\eta^{n} - \overline{M}_\eta^{n}\Delta\widehat{Q}^{n} \qquad (50)$$

Therefore,

$$\widehat{S}^{n+1} = \widehat{S}^{n} + \widehat{M}^{n}\Delta\widehat{Q}^{n} - \overline{M}_\eta^{n}\Delta\widehat{Q}^{n} + (\overline{M}\Delta\widehat{Q})_\eta^{n}$$

$$= \widehat{S}^{n} + (\widehat{M}^{n} - \overline{M}_\eta^{n})\Delta\widehat{Q}^{n} + \frac{\partial}{\partial\eta}(\overline{M}\Delta\widehat{Q})^{n} \qquad (51)$$

$$\overline{M} = \frac{\partial\widehat{S}}{\partial\widehat{Q}_\eta} = \begin{bmatrix} 0 & & \\ & \frac{J^2\mu}{\rho}(\frac{4}{3}y_\xi^2 + x_\xi^2) & \\ & & \frac{J^2\mu}{\rho}(y_\xi^2 + \frac{4}{3}x_\xi^2) \\ & & \frac{J^2\mu}{\rho}\frac{\gamma}{Pr}(x_\xi^2 + y_\xi^2) \end{bmatrix}$$

3.6 Jameson Model For $D_{j,k}(Q^{n+1})$ Term

$$D_{j,k}(Q^{n+1}) = D_{j,k}(Q^{n}) + \Delta D_{j,k}(Q^{n}) \qquad (52)$$

where

$$D_{j,k}(Q) = d\Big|_{j-\frac{1}{2},k}^{j+\frac{1}{2},k} + d\Big|_{j,k-\frac{1}{2}}^{j,k+\frac{1}{2}} \qquad (53)$$

$$\Delta D_{j,k}(Q) = \Delta d\Big|_{j-\frac{1}{2},k}^{j+\frac{1}{2},k} + \Delta d\Big|_{j,k-\frac{1}{2}}^{j,k+\frac{1}{2}} \tag{54}$$

$$d_{j+\frac{1}{2},k} = \frac{J^{-1}_{j+\frac{1}{2},k}}{\Delta\tau}\Big[\varepsilon^{(2)}_{j+\frac{1}{2},k}(Q_{j+1,k} - Q_{j,k}) - \varepsilon^{(4)}_{j+\frac{1}{2},k}(Q_{j+2,k} - 3Q_{j+1,k} + 3Q_{j,k} - Q_{j-1,k})\Big] \tag{55}$$

$$\Delta d_{j+\frac{1}{2},k} = \frac{J^{-1}_{j+\frac{1}{2},k}}{\Delta\tau}\Big[\varepsilon^{(2)}_{j+\frac{1}{2},k}(\Delta Q_{j+1,k} - \Delta Q_{j,k}) - \varepsilon^{(4)}_{j+\frac{1}{2},k}(\Delta Q_{j+2,k} - 3\Delta Q_{j+1,k} + 3\Delta Q_{j,k} - \Delta Q_{j-1,k})\Big] \quad \text{etc.} \tag{56}$$

$$\varepsilon^{(2)}_{j+\frac{1}{2},k} = K^{(2)}\ \ \mathrm{Max}(\nu_{j+1,k}\ ,\ \nu_{j,k}) \tag{57}$$

$$\varepsilon^{(4)}_{j+\frac{1}{2},k} = \mathrm{Max}[0, (K^{(4)} - \varepsilon^{(2)}_{j+\frac{1}{2},k})] \quad \text{etc.} \tag{58}$$

and

$$\nu_{j,k} = \frac{|p_{j+1,k} - 2p_{j,k} + p_{j-1,k}|}{|p_{j+1,k} + 2p_{j,k} + p_{j-1,k}|}$$

Rewriting Equation (48),

$$\frac{J^{-1}_{j,k}}{\Delta\tau}\Delta Q^n_{j,k} + [(y_\eta A^n - x_\eta B^n)\Delta Q^n]\Big|_{j-\frac{1}{2},k}^{j+\frac{1}{2},k} + [(-y_\xi A^n + x_\xi B^n)\Delta Q^n]\Big|_{j,k-\frac{1}{2}}^{j,k+\frac{1}{2}} - Re^{-1}(\Delta\widehat{S}^n)\Big|_{j,k-\frac{1}{2}}^{j,k+\frac{1}{2}} - \Delta d\Big|_{j-\frac{1}{2},k}^{j+\frac{1}{2},k} - \Delta d\Big|_{j,k-\frac{1}{2}}^{j,k+\frac{1}{2}} = Res^n_{j,k} \tag{59}$$

where

$$A = \frac{\partial E}{\partial Q}$$

$$= -\begin{bmatrix} 0 & -1 & 0 & 0 \\ \frac{(3-\gamma)u^2}{2} + \frac{(1-\gamma)v^2}{2} & (\gamma-3)u & (\gamma-1)v & (1-\gamma) \\ uv & -v & -u & 0 \\ \frac{\gamma eu}{\rho} + (1-\gamma)u(u^2+v^2) & \frac{-\gamma e}{\rho} + \frac{(\gamma-1)}{2}(3u^2+v^2) & (\gamma-1)uv & -\gamma u \end{bmatrix}$$

$$B = \frac{\partial F}{\partial Q}$$

$$= - \begin{bmatrix} 0 & 0 & -1 & 0 \\ uv & -v & -u & 0 \\ \frac{(3-\gamma)}{2}v^2 + \frac{(1-\gamma)}{2}u^2 & (\gamma-1)u & (\gamma-3)v & (1-\gamma) \\ \frac{\gamma ev}{\rho} + (1-\gamma)v(u^2+v^2) & (\gamma-1)uv & \frac{-\gamma e}{\rho} + \frac{(\gamma-1)}{2}(3v^2+u^2) & -\gamma v \end{bmatrix}$$

$$Res^n_{j,k} = -[y_\eta E^n - x_\eta F^n]\Big|^{j+\frac{1}{2},k}_{j-\frac{1}{2},k} - [y_\xi E^n + x_\xi F^n]\Big|^{j,k+\frac{1}{2}}_{j,k-\frac{1}{2}} + Re^{-1}(\widehat{S}^n)\Big|^{j,k+\frac{1}{2}}_{j,k-\frac{1}{2}} + d\Big|^{j+\frac{1}{2},k}_{j-\frac{1}{2},k} + d\Big|^{j,k+\frac{1}{2}}_{j,k-\frac{1}{2}} \quad (60)$$

Rewriting Equation (59),

$$\frac{J^{-1}_{j,k}}{\Delta\tau}\Delta Q^n_{j,k} + [(y_\eta A^n - x_\eta B^n)\Delta Q^n]\Big|^{j+\frac{1}{2},k}_{j-\frac{1}{2},k} + [(-y_\xi A^n + x_\xi B^n)\Delta Q^n]\Big|^{j,k+\frac{1}{2}}_{j,k-\frac{1}{2}} - Re^{-1}$$

$$[(\widehat{M}^n - \overline{M}^n_\eta)\Delta\widehat{Q}^n + \frac{\partial}{\partial\eta}(\overline{M}\Delta\widehat{Q})^n]\Big|^{j,k+\frac{1}{2}}_{j,k-\frac{1}{2}} - (\Delta d)\Big|^{j+\frac{1}{2},k}_{j-\frac{1}{2},k} - (\Delta d)\Big|^{j,k+\frac{1}{2}}_{j,k-\frac{1}{2}} = Res^n_{j,k} \quad (61)$$

For steady-state solution we need to drive the right hand side, Res^n to zero asymptotically as $n \to \infty$. We can make simplification to left hand side operator to enhance computational efficiency. Once such simplication is to drop $(\widehat{M}^n - \overline{M}^n_\eta)$ term and retain only the diagonal terms of $\overline{M}$. The justification for the above is that, even though we are simplifying the implicit part, all the terms are retained in the residue, hence the steady state solution is not affected.

$$\text{Diag } \overline{M} = \text{Diag}\left[0, \frac{J^2\mu}{\rho}(\tfrac{4}{3}y^2_\xi + x^2_\xi), \frac{J^2\mu}{\rho}(y^2_\xi + \tfrac{4}{3}x^2_\xi), \frac{J^2\mu}{\rho}\frac{\gamma}{Pr}(x^2_\xi + y^2_\xi)\right] \quad (62)$$

$$\frac{J^{-1}_{j,k}}{\Delta\tau}\Delta Q^n_{j,k} + [(y_\eta A^n - x_\eta B^n)\Delta Q^n]\Big|^{j+\frac{1}{2},k}_{j-\frac{1}{2},k} + [(-y_\xi A^n + x_\xi B^n)\Delta Q^n]\Big|^{j,k+\frac{1}{2}}_{j,k-\frac{1}{2}}$$

$$- Re^{-1}[\overline{M}\frac{\partial}{\partial\eta}(\Delta Q)^n]\Big|^{j,k+\frac{1}{2}}_{j,k-\frac{1}{2}} - (\Delta d)\Big|^{j+\frac{1}{2},k}_{j-\frac{1}{2},k} - (\Delta d)\Big|^{j,k+\frac{1}{2}}_{j,k-\frac{1}{2}} = Res^n_{j,k} \quad (63)$$

In Equation (63),

$$(y_\eta A^n - x_\eta B^n)\Delta Q^n\Big|_{j-\frac{1}{2},k}^{j+\frac{1}{2},k} = \frac{1}{2}(y_\eta A^n - x_\eta B^n)_{j+\frac{1}{2},k}\Delta Q^n_{j+1,k}$$
$$+ \frac{1}{2}\left[(y_\eta A^n - x_\eta B^n)_{j+\frac{1}{2},k} - (y_\eta A^n - x_\eta B^n)_{j-\frac{1}{2},k}\right]\Delta Q^n_{j,k}$$
$$- \frac{1}{2}(y_\eta A^n - x_\eta B^n)_{j-\frac{1}{2},k}\Delta Q^n_{j-1,k}$$

$$(-y_\xi A^n + x_\xi B^n)\Delta Q^n\Big|_{j,k-\frac{1}{2}}^{j,k+\frac{1}{2}} = \frac{1}{2}(-y_\xi A^n + x_\xi B^n)_{j,k+\frac{1}{2}}\Delta Q_{j,k+1}$$
$$+ \frac{1}{2}\left[(-y_\xi A^n + x_\xi B^n)_{j,k+\frac{1}{2}} - (-y_\xi A^n + x_\xi B^n)_{j,k-\frac{1}{2}}\right]\Delta Q^n_{j,k}$$
$$- \frac{1}{2}(-y_\xi A^n + x_\xi B^n)_{j,k-\frac{1}{2}}\Delta Q_{j,k-1}$$

$$\overline{M}\frac{\partial}{\partial\eta}\Delta\widehat{Q}\Big|_{j,k-\frac{1}{2}}^{j,k+\frac{1}{2}} \simeq R\frac{\partial}{\partial\eta}\Delta Q\Big|_{j,k-\frac{1}{2}}^{j,k+\frac{1}{2}}$$
$$\simeq R_{j,k+\frac{1}{2}}(\Delta Q_{j,k+1} - \Delta Q_{j,k}) - R_{j,k-\frac{1}{2}}(\Delta Q_{j,k} - \Delta Q_{j,k-1})$$

where

$$R = \begin{bmatrix} 0 & & \\ & \frac{J\mu}{\rho}\left(\frac{4}{3}y_\xi^2 + x_\xi^2\right) & \\ & & \frac{J\mu}{\rho}\left(y_\xi^2 + \frac{4}{3}x_\xi^2\right) \\ & & \frac{J\mu}{Pr}\frac{\gamma}{\rho}\left(x_\xi^2 + y_\xi^2\right) \end{bmatrix}$$

Substituting the above in Equation (63) we obtain,

$$CL_{j,k}\Delta Q_{j-1,k} + CR_{j,k}\Delta Q_{j+1,k} + CC_{j,k}\Delta Q_{j,k}$$
$$+ CB_{j,k}\Delta Q_{j,k-1} + CT_{j,k}\Delta Q_{j,k+1}$$
$$= Res^n_{j,k} - \frac{J^{-1}_{j-\frac{1}{2},k}}{\Delta\tau}\epsilon^{(4)}_{j-\frac{1}{2},k}\Delta Q_{j-2,k} - \frac{J^{-1}_{j+\frac{1}{2},k}}{\Delta\tau}\epsilon^{(4)}_{j+\frac{1}{2}.k}\Delta Q_{j+2,k}$$
$$- \frac{J^{-1}_{j,k-\frac{1}{2}}}{\Delta\tau}\epsilon^{(4)}_{j,k-\frac{1}{2}}\Delta Q_{j,k-2} - \frac{J^{-1}_{j,k+\frac{1}{2}}}{\Delta\tau}\epsilon^{(4)}_{j,k+\frac{1}{2}}\Delta Q_{j,k+2} \qquad (64)$$

where

$$CL_{j,k} = -\frac{1}{2}(y_\eta A^n - x_\eta B^n)_{j-\frac{1}{2},k} - \frac{J^{-1}_{j+\frac{1}{2},k}}{\Delta\tau}\epsilon^{(4)}_{j+\frac{1}{2},k}I$$
$$- \frac{J^{-1}_{j-\frac{1}{2},k}}{\Delta\tau}\epsilon^{(2)}_{j-\frac{1}{2},k}I - 3\frac{J^{-1}_{j-\frac{1}{2},k}}{\Delta\tau}\epsilon^{(4)}_{j-\frac{1}{2},k}I$$

$$
\begin{aligned}
CR_{j,k} &= \frac{1}{2}(y_\eta A^n - x_\eta B^n)_{j+\frac{1}{2},k} - \frac{J^{-1}_{j+\frac{1}{2},k}}{\Delta\tau}\epsilon^{(2)}_{j+\frac{1}{2},k} I \\
&\quad - 3\frac{J^{-1}_{j+\frac{1}{2},k}}{\Delta\tau}\epsilon^{(4)}_{j+\frac{1}{2},k} I - \frac{J^{-1}_{j-\frac{1}{2},k}}{\Delta\tau}\epsilon^{(4)}_{j-\frac{1}{2},k} I \\
CC_{j,k} &= \frac{1}{2}(y_\eta A^n - x_\eta B^n)_{j+\frac{1}{2},k} - (y_\eta A^n - x_\eta B^n)_{j-\frac{1}{2},k} \\
&\quad + \frac{1}{2}(-y_\xi A^n + x_\xi B^n)_{j,k+\frac{1}{2}} - (-y_\xi A^n + x_\xi B^n)_{j,k-\frac{1}{2}} \\
&\quad + Re^{-1}(R_{j,k+\frac{1}{2}} + R_{j,k-\frac{1}{2}}) + \frac{J^{-1}_{j,k}}{\Delta\tau} I + \frac{J^{-1}_{j+\frac{1}{2},k}}{\Delta\tau}\epsilon^{(2)}_{j+\frac{1}{2},k} I \\
&\quad + 3\frac{J^{-1}_{j+\frac{1}{2},k}}{\Delta\tau}\epsilon^{(4)}_{j+\frac{1}{2},k} I + \frac{J^{-1}_{j-\frac{1}{2},k}}{\Delta\tau}\epsilon^{(2)}_{j-\frac{1}{2},k} I + 3\frac{J^{-1}_{j-\frac{1}{2},k}}{\Delta\tau}\epsilon^{(4)}_{j-\frac{1}{2},k} I \\
&\quad + \frac{J^{-1}_{j,k+\frac{1}{2}}}{\Delta\tau}\epsilon^{(2)}_{j,k+\frac{1}{2}} I + 3\frac{J^{-1}_{j,k+\frac{1}{2}}}{\Delta\tau}\epsilon^{(4)}_{j,k+\frac{1}{2}} I + \frac{J^{-1}_{j,k-\frac{1}{2}}}{\Delta\tau}\epsilon^{(2)}_{j,k-\frac{1}{2}} I + 3\frac{J^{-1}_{j,k-\frac{1}{2}}}{\Delta\tau}\epsilon^{(4)}_{j,k-\frac{1}{2}} I \\
CB_{j,k} &= -\frac{1}{2}(-y_\xi A^n + x_\xi B^n)_{j,k-\frac{1}{2}} - Re^{-1} R_{j,k-\frac{1}{2}} \\
&\quad - \frac{J^{-1}_{j,k+\frac{1}{2}}}{\Delta\tau}\epsilon^{(4)}_{j,k+\frac{1}{2}} I - \frac{J^{-1}_{j,k-\frac{1}{2}}}{\Delta\tau}\epsilon^{(2)}_{j,k-\frac{1}{2}} I - 3\frac{J^{-1}_{j,k-\frac{1}{2}}}{\Delta\tau}\epsilon^{(4)}_{j,k-\frac{1}{2}} I
\end{aligned}
$$

and

$$
\begin{aligned}
CR_{j,k} &= \frac{1}{2}(-y_\xi A^n + x_\xi B^n)_{j,k+\frac{1}{2}} - Re^{-1} R_{j,k+\frac{1}{2}} \\
&\quad - \frac{J^{-1}_{j,k+\frac{1}{2}}}{\Delta\tau}\epsilon^{(2)}_{j,k+\frac{1}{2}} I - 3\frac{J^{-1}_{j,k+\frac{1}{2}}}{\Delta\tau}\epsilon^{(4)}_{j,k+\frac{1}{2}} I - \frac{J^{-1}_{j,k-\frac{1}{2}}}{\Delta\tau}\epsilon^{(4)}_{j,k-\frac{1}{2}} I
\end{aligned}
$$

3.7 Single Point Locally Implicit Scheme

Now, the linearized Equation at node (j,k) can be written in delta form using Equation (64) as,

$$
\begin{aligned}
\Delta Q^n_{j,k} = CC^{-1}\{ & Res^n_{j,k} - \frac{J^{-1}_{j-\frac{1}{2},k}}{\Delta\tau}\epsilon^{(4)}_{j-\frac{1}{2},k}\Delta Q^n_{j-2,k} I - \frac{J^{-1}_{j+\frac{1}{2},k}}{\Delta\tau}\epsilon^{(4)}_{j+\frac{1}{2},k}\Delta Q^n_{j+2,k} I \\
& - \frac{J^{-1}_{j,k-\frac{1}{2}}}{\Delta\tau}\epsilon^{(4)}_{j,k-\frac{1}{2}}\Delta Q^n_{j,k-2} I - \frac{J^{-1}_{j,k+\frac{1}{2}}}{\Delta\tau}\epsilon^{(4)}_{j,k+\frac{1}{2}}\Delta Q_{j,k+2} I - CL_{j,k}\Delta Q^n_{j-1,k} \\
& - CB_{j,k}\Delta Q^n_{j,k-1} - CR^n_{j,k}\Delta Q^n_{j+1,k} - CT_{j,k}\Delta Q^n_{j,k+1}\}
\end{aligned}
\tag{65}
$$

3.8 Inner Iteration

The system of linear Equations (65) for ΔQ form a system which couples all the nodes in the flow field. However, the coupling is strongest in a local zone surrounding a (j,k) node. With local time stepping and reasonable Courant numbers (of about 10), we can solve these linear Equations approximately by a modified Gauss–Seidel iteration technique. The Equation for iterative corrections to ΔQ is written as

$$C \cdot dQ_{j,k} = Res^n_{j,k} - L_{j,k}(\Delta Q) \tag{66}$$

$$\Delta Q^{m+1}_{j,k} = \Delta Q^{(m)}_{j,k} + \omega_{in} \;\; dQ_{j,k} \;\; , m = 0, - - -, 3 \tag{67}$$

where C is a diagonal matrix defined as a modification to the coefficient matrix $CC_{j,k}$ in Equation (65) and ω_{in} is a relaxation parameter.

$$\begin{aligned} C_{j,k} = {} & CFL(\frac{J^{-1}_{j,k}}{\Delta\tau})(0.5 + 1/CFL) + \frac{J^{-1}_{j+\frac{1}{2},k}}{\Delta\tau}\epsilon^{(2)}_{j+\frac{1}{2},k} I \\ & + 3\frac{J^{-1}_{j+\frac{1}{2},k}}{\Delta\tau}\epsilon^{(4)}_{j+\frac{1}{2},k} I + \frac{J^{-1}_{j-\frac{1}{2},k}}{\Delta\tau}\epsilon^{(2)}_{j-\frac{1}{2},k} I + 3\frac{J^{-1}_{j-\frac{1}{2},k}}{\Delta\tau}\epsilon^{(4)}_{j-\frac{1}{2},k} I + \frac{J^{-1}_{j,k+\frac{1}{2}}}{\Delta\tau}\epsilon^{(2)}_{j,k+\frac{1}{2}} I \\ & + 3\frac{J^{-1}_{j,k+\frac{1}{2}}}{\Delta\tau}\epsilon^{(4)}_{j,k+\frac{1}{2}} I + \frac{J^{-1}_{j,k-\frac{1}{2}}}{\Delta\tau}\epsilon^{(2)}_{j,k-\frac{1}{2}} I + 3\frac{J^{-1}_{j,k-\frac{1}{2}}}{\Delta\tau}\epsilon^{(4)}_{j,k-\frac{1}{2}} I \end{aligned} \tag{68}$$

ΔQ on the right side of Equation (66) takes the latest available iterate. The inner iteration for ΔQ can be computed rapidly since the dQ corrections are explicit scalar Equations. Four symmetric inner iterations are performed for each time step, with each iteration starting at a different corner of the computational boundary as shown in Figure 10 and sweeping the entire flow field. The modified C matrix of the Equation (68) is designed to compensate for the fact that in a Gauss–Seidel iteration the information at half of the nodes surrounding (j,k) is at the latest level while the other half is at the previous level.

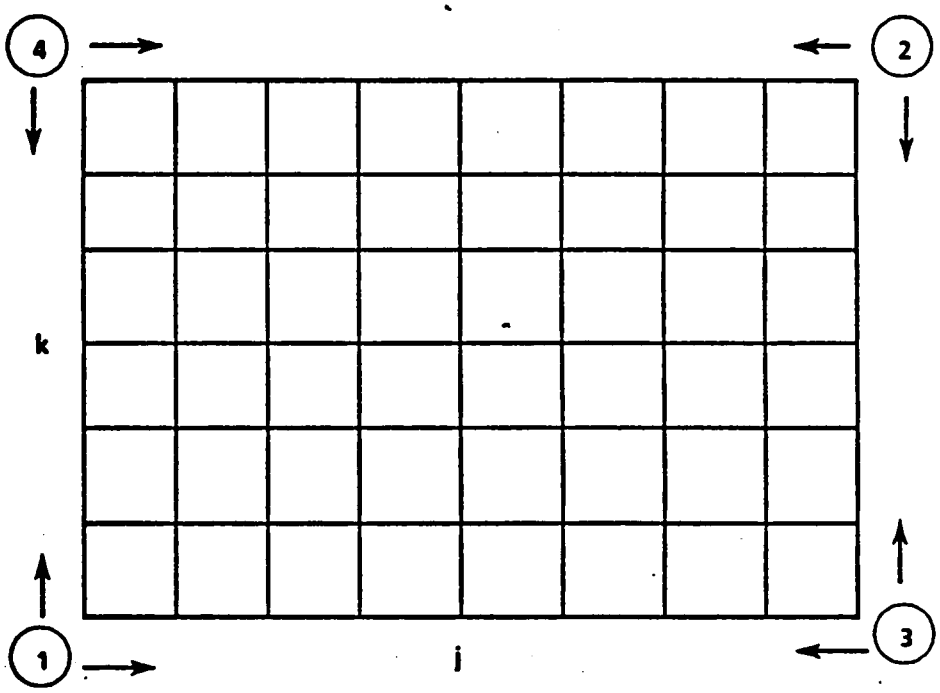


Figure 10: Schematic of Symmetric Sweeping

The stability and convergence of the scheme do not seem to be sensitive to the order of the iteration sweeps as long as a symmetric cycle is performed. ω_{in} is an inner relaxation parameter of the order of 0.9-1.1 and depends upon the type of flow. Another relaxation parameter ω_{out} is introduced at the end of each time step.

$$Q^{n+1} = Q^n + \omega_{out}\Delta Q \tag{69}$$

3.9 Time Step

Local time stepping is used to accelerate convergence to a steady state solution. The time step is scaled in ξ and η directions.

In ξ direction,

$$\Delta t_\xi = [c + |y_\eta u - x_\eta v|/SK] \cdot SK \tag{70}$$

$$SK = (x_\eta^2 + y_\eta^2)^{\frac{1}{2}}$$

and in η direction,

$$\Delta t_\eta = [c + | - y_\xi u + x_\xi v|/SJ]SJ \tag{71}$$

$$SJ = (x_\xi^2 + y_\xi^2)^{\frac{1}{2}} \tag{72}$$

The total time step Δt is given by,

$$\Delta t = c(SJ + SK) + |U1| + |V1| \tag{73}$$

where

$$U1 = y_\eta u - x_\eta v$$

$$V1 = -y_\xi u + x_\xi v$$

$$c = \text{speed of sound}$$

3.10 Multigrid Technique

The present scheme is locally implicit and globally explicit. It has the proper smoothing properties for high-frequency components, and a multigrid iteration scheme is incorporated to speed up the convergence of the scheme for steady state problems. At a particular grid level, h, one or more time step calculations are made and the solution is updated. The solution Q_h is injected to a coarser grid level, $2h$, by the volume weighed average:

$$Q_{2h} = \left(\sum v_h Q_h\right) / v_{2h} \tag{74}$$

and the residual is injected by the summation over the fine grid cells,

$$R_{2h} = \sum Res_h \tag{75}$$

It may be noted that injection is made only in the interior. The boundary is treated separately. Two kinds of boundary injection have been tried out: 1) freezing the conservative variables and pressure on the outer boundary and 2) taking the volume weighted average of the flow variables on the outer boundary. Both methods have been found to be effective, but the first method gives slightly faster convergence for subsonic flow.

The pressure, p, is injected to the coarser grid levels just like the conserved variables. A forcing function FF, is defined as the difference between the injected residual and the residual calculated on that grid using the injected solution.

$$FF_{2h} = R_{2h} - Res(Q_{2h}) \tag{76}$$

A corresponding forcing pressure, FP, is defined in a manner similar to the forcing function, FF. The residual on the coarse grid is defined by

$$\overline{Res_{2h}} = Res_{2h} + FF_{2h} \tag{77}$$

The linearized Equation corresponding to Equation (65) on the coarse grid is

$$L(\Delta Q)_{2h} = \overline{Res_{2h}} \tag{78}$$

and is solved for one or more time steps by the locally implicit scheme as outlined earlier. The process is repeated until the coarsest grid solution is computed. In the present formulation, multigrid solutions are obtained with only one time step calculation at each grid level per cycle with a fixed local Courant number. The coarse grid information is passed back to the finer grids as follows: The difference in the updated solution and the injected solution on the coarse grid is interpolated in computational space at the grid points of the next finer mesh and is added to the solution which has been previously computed at that level. In the present formulation, corrections are passed back from all the coarse grids to the fine grid without recomputing the solutions at the intermediate grids. The pressure is the dominant driver of a flow field and it has been found necessary to inject the fine grid pressures on the coarser grids to prevent the shock location from oscillating during a multigrid cycle.

3.11 Results and Discussion

Elliptic and hyperbolic grid generators have been used to construct C-grids around NACA0012 and RAE2822 airfoils respectively. Figure 11 shows the C-grid around NACA0012 airfoil and Figure 12 shows the closeup view of the grid of Figure 11 near the leading edge. Test cases have been chosen which have either experimental or numerical results in the literature. The first case chosen is a subsonic flow over NACA0012 airfoil with a free stream Mach number of 0.5 and Reynolds number of 5000 at a zero degree angle of attack. The grid around this airfoil requires only moderate stretching for low Reynolds number flows. Figure 13 shows the convergence history, in terms of log (Res) versus the number of work units. The residue drops three and one half orders of magnitude in 200 multigrid

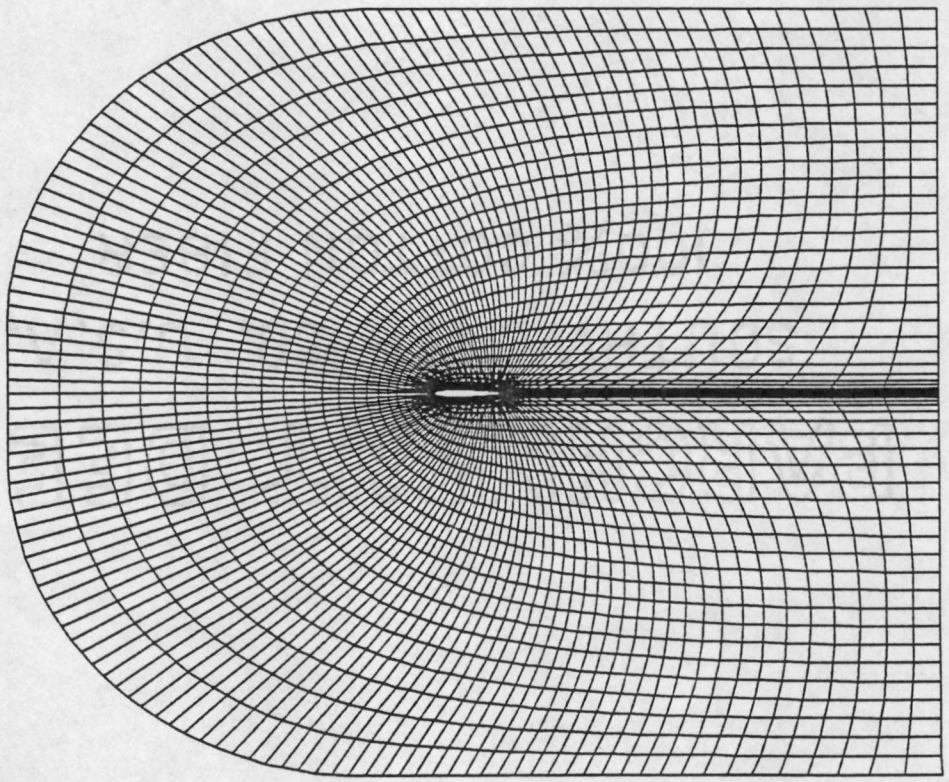


Figure 11: C–grid Around NACA 0012 Airfoil 128 x 32
(Elliptic grid generator)

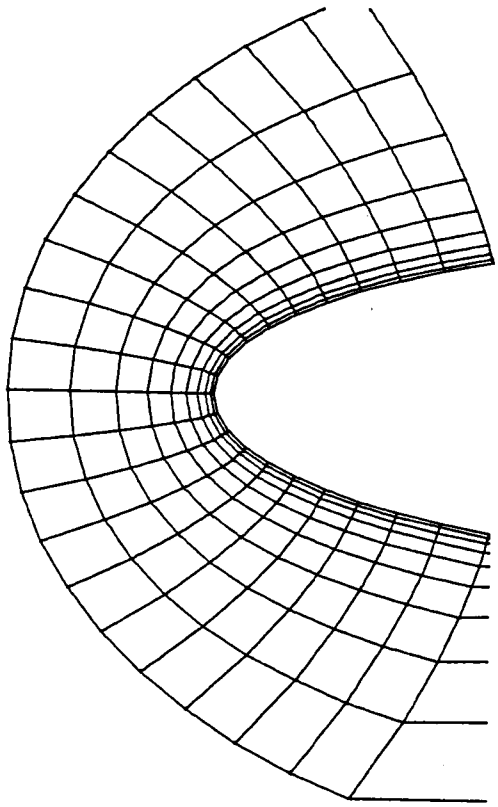


Figure 12: Closeup View of C-grid Around NACA 0012 Airfoil Near the Leading Edge

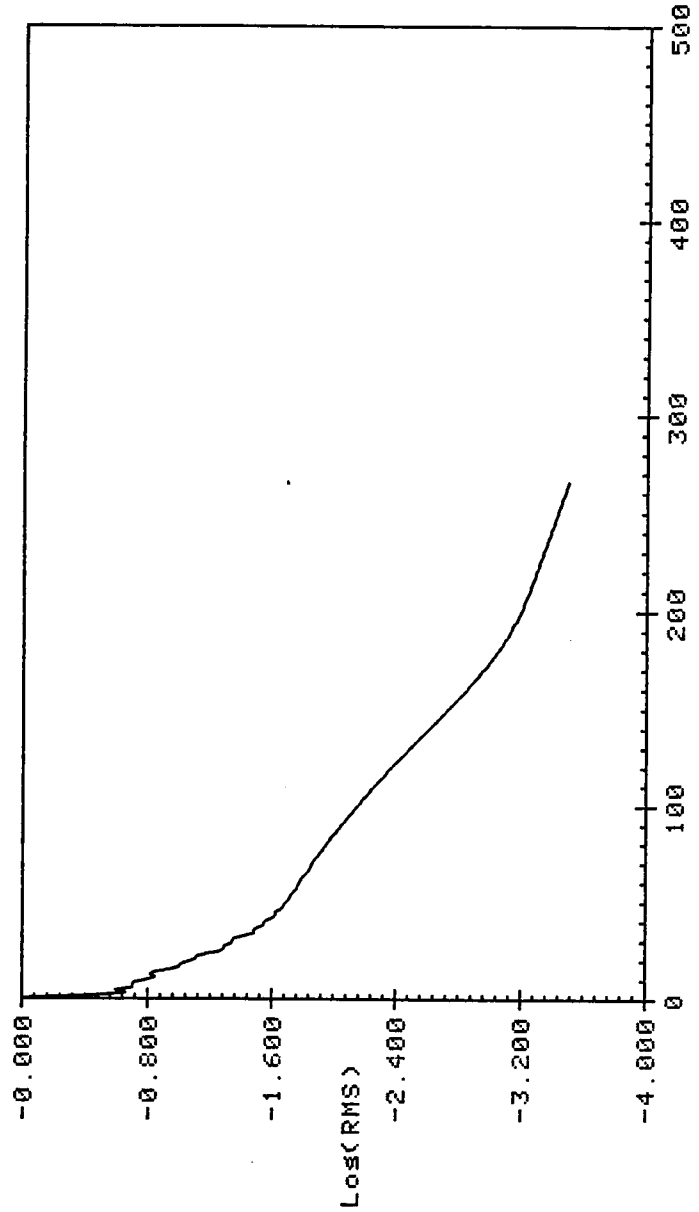


Figure 13: Convergence History of Test Case 1 ($Re = 5000$, $M_\infty = 0.5$, $\alpha = 0.0$)

cycles or 262 work units. Figure 14 and 15 show the Mach and pressure contours. The convergence rate is comparable to that of Jameson's [10]. The Mach contours are in good agreement Jameson's results. Only one time step has been computed at each grid level during the injection cycle and in the type of multigrid used the flow variables have been frozen on the outer boundary throughout the injection cycle. Cook, McDonald and Firmin [23] have conducted extensive experimental studies on RAE2822 airfoil and presented the data for a variety of test cases. Calculations have been made for select cases and the results are compared with the experimental results. For these cases a C-grid (128x32) is generated by a hyperbolic grid generator where the grid spacing normal to the body varies from 0.00001 to 2.0 chord lengths. The grid is as shown in Figure 16. Figure 17 shows the comparison of pressure coefficient between numerical and experimental cases. This is the case where free stream Mach number is 0.676, Reynolds number is 5.7 x 10^6 and the angle of attack is 2.4 degrees. It is expected that the pressure distribution on the suction surface would be closer to the experiment if we used a larger grid size. Figure 18 and 19 show the Mach and pressure contours for the above case. The next test case selected is a flow of free stream Mach number of 0.6, Reynolds number of 6.3 x 10^6 and an angle of attack of 2.57 degrees. Figure 20 compares the pressure distribution and Figures 21 and 22 show the Mach and pressure contours. Having obtained satisfactory results for shock free flows, it has been decided to test the code with transonic flows. The test case chosen is the flow with a free stream Mach number of 0.725, Reynolds number of 6.5 x 10^6 and an angle of attack of 2.92 degrees. Figure 23 shows the comparison of the numerical and experimental pressure distributions. Here again it is expected that the shock can be made sharper by increasing the grid resolution. Figure 24 shows the Mach contours where the shock location can be clearly seen. Figure 25

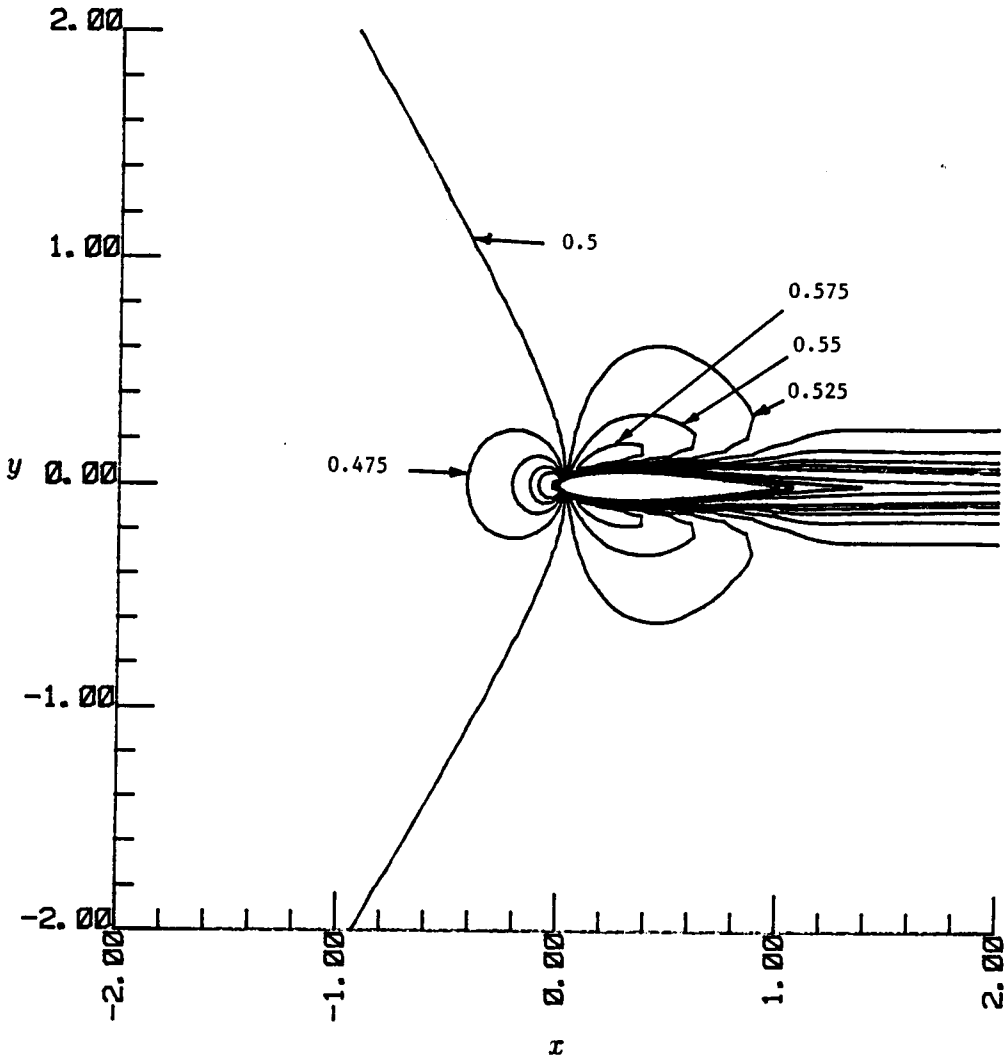


Figure 14: Mach Contours of Test Case 1

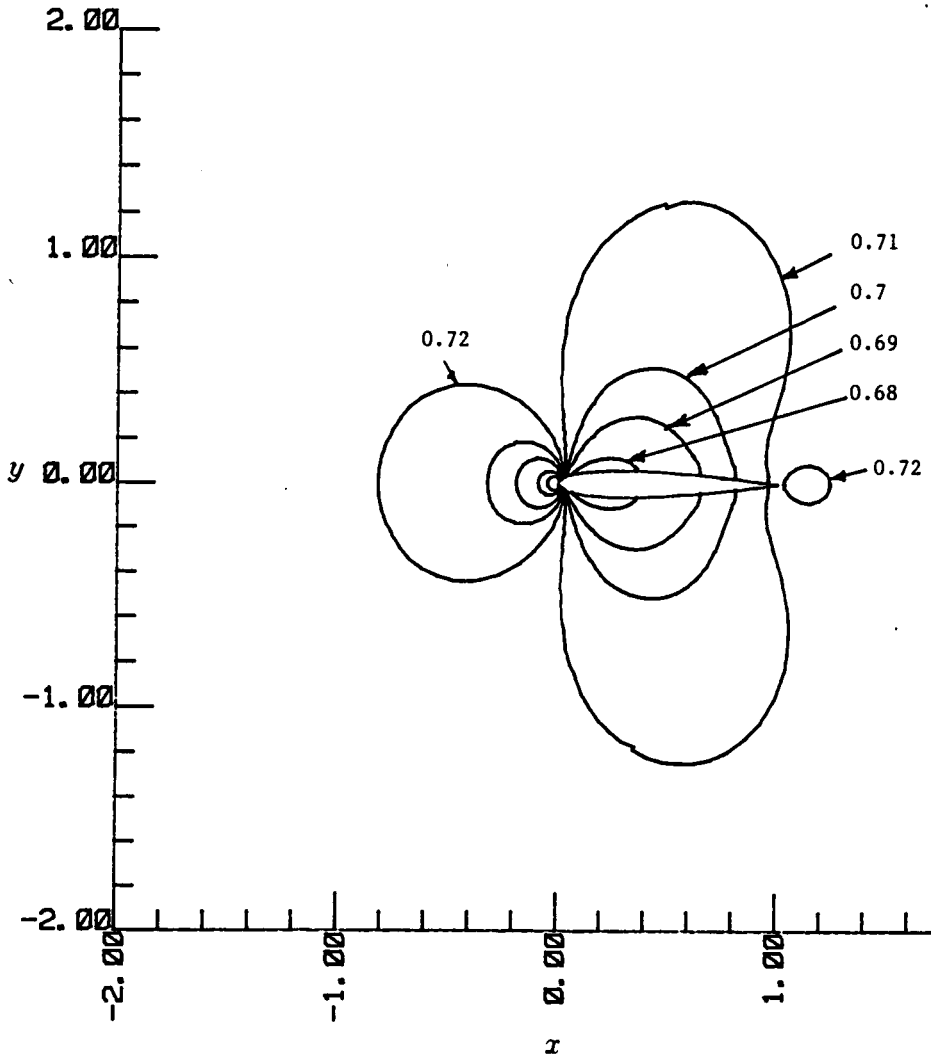


Figure 15: Pressure Contours of Test Case 1

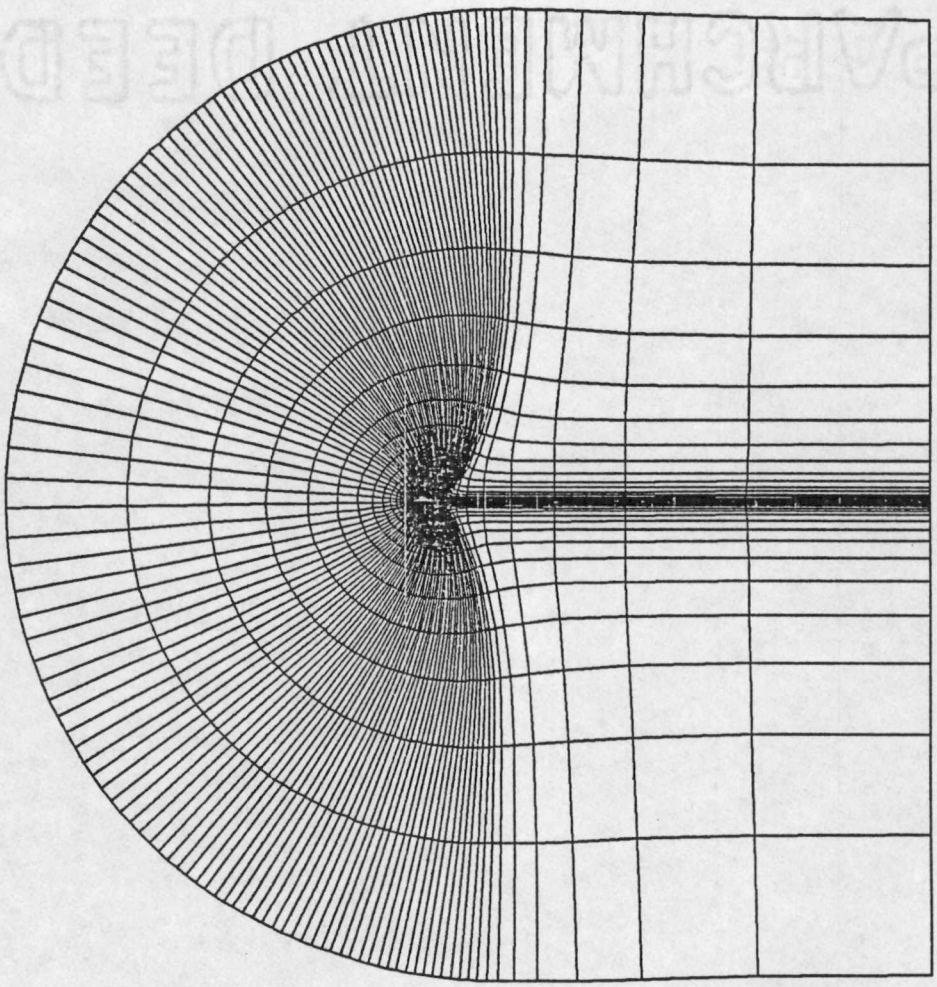


Figure 16: C–grid Around RAE 2822 Airfoil 128 x 32
(Hyperbolic Grid Generator)

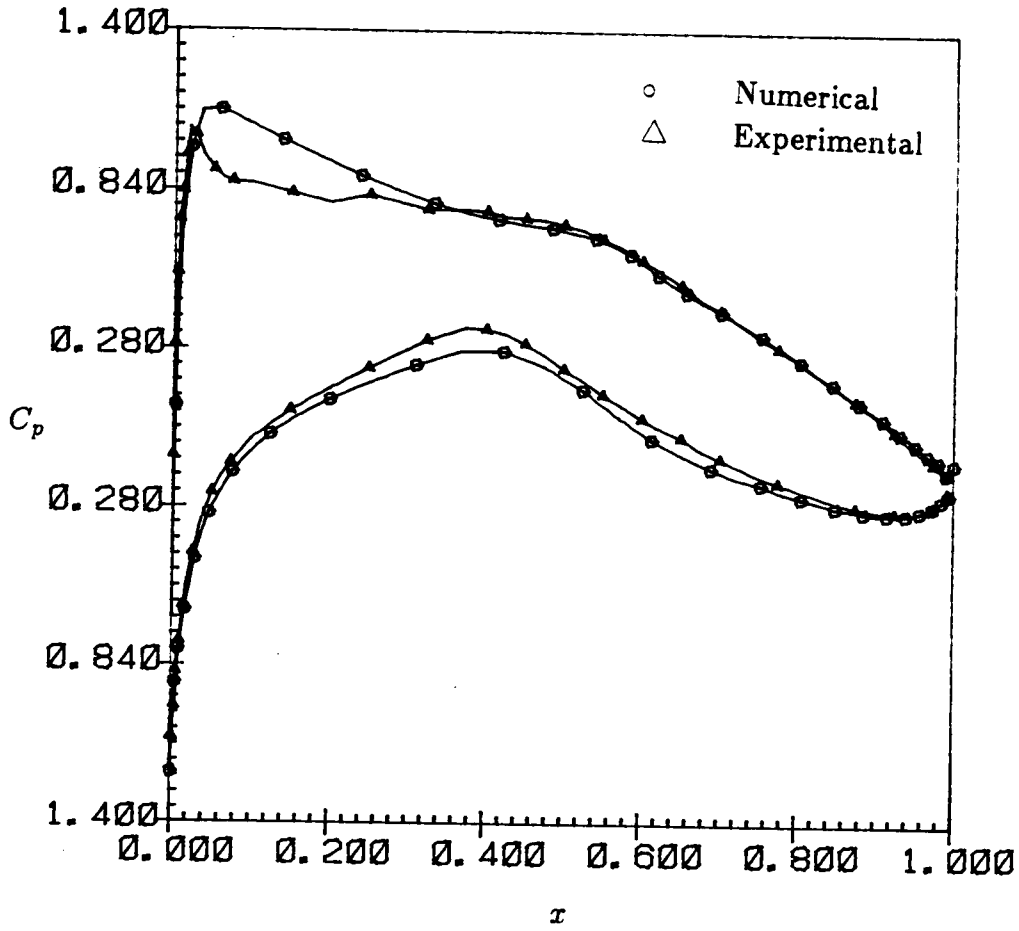


Figure 17: Comparison of Numerical and Experimental Pressure Coefficients for Test Case 2 ($Re = 5.7 \times 10^6$, $M_\infty = 0.676$, $\alpha = 2.4$)

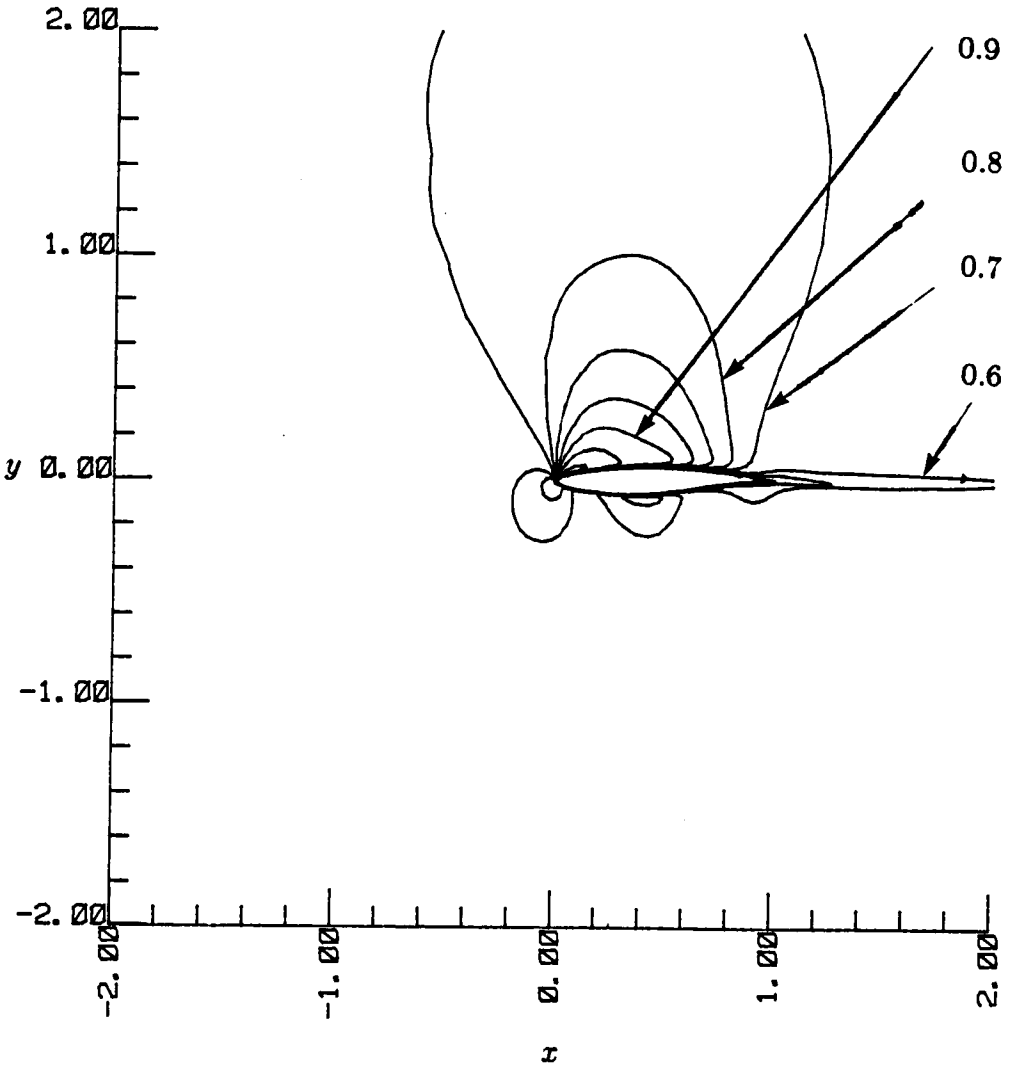


Figure 18: Mach Contours of Flow Field (Test Case 2)

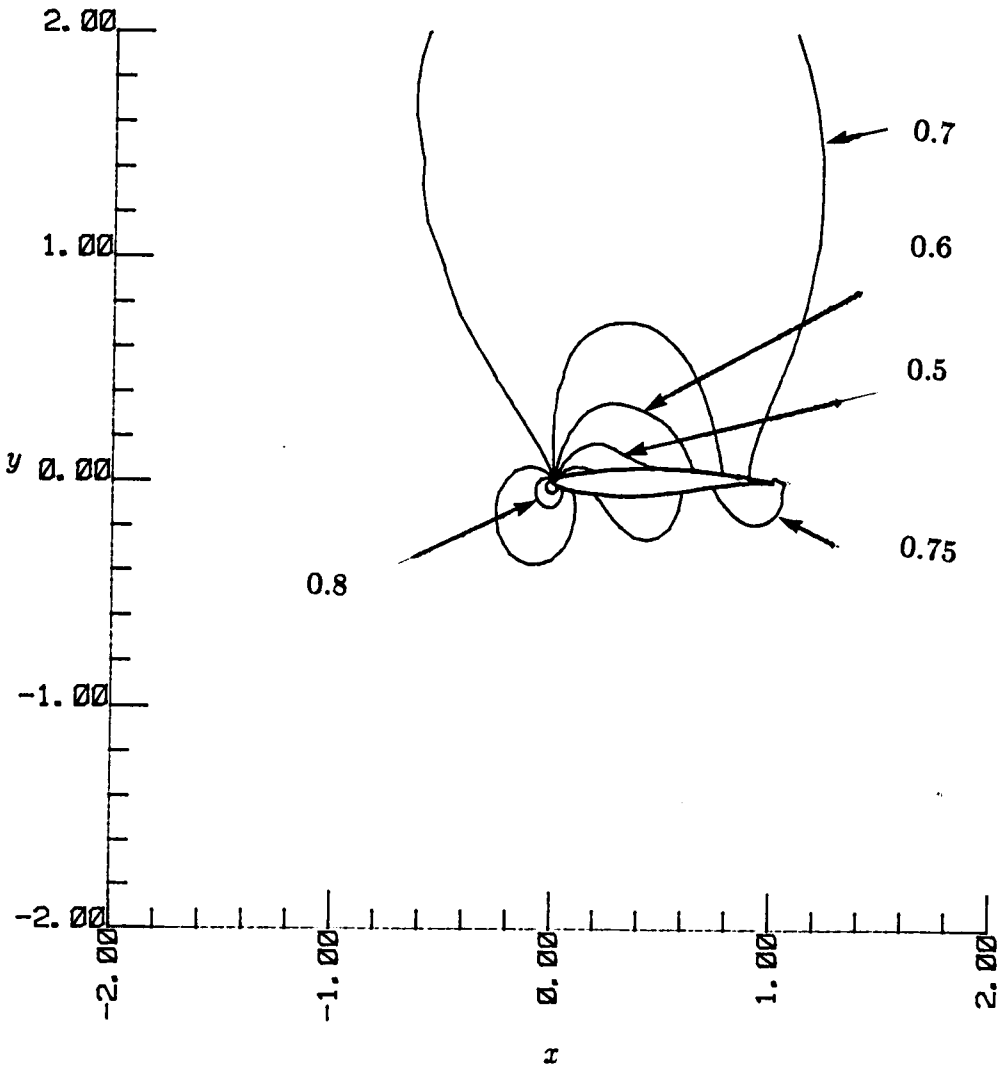


Figure 19: Pressure Contours of Flow Field (Test Case 2)

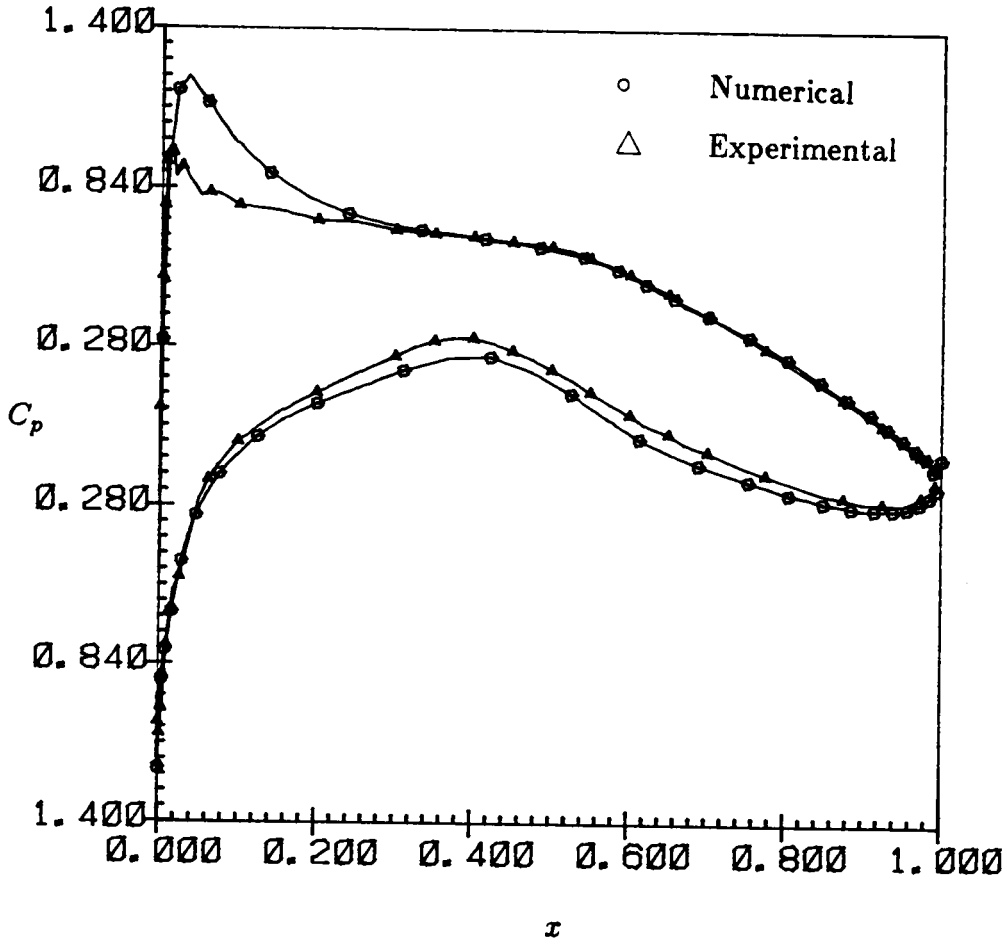


Figure 20: Comparison of Numerical and Experimental Pressure Distributions for Test Case 3 ($Re = 6.3 \times 10^6$, $M_\infty = 0.6,\ \alpha = 2.57$)

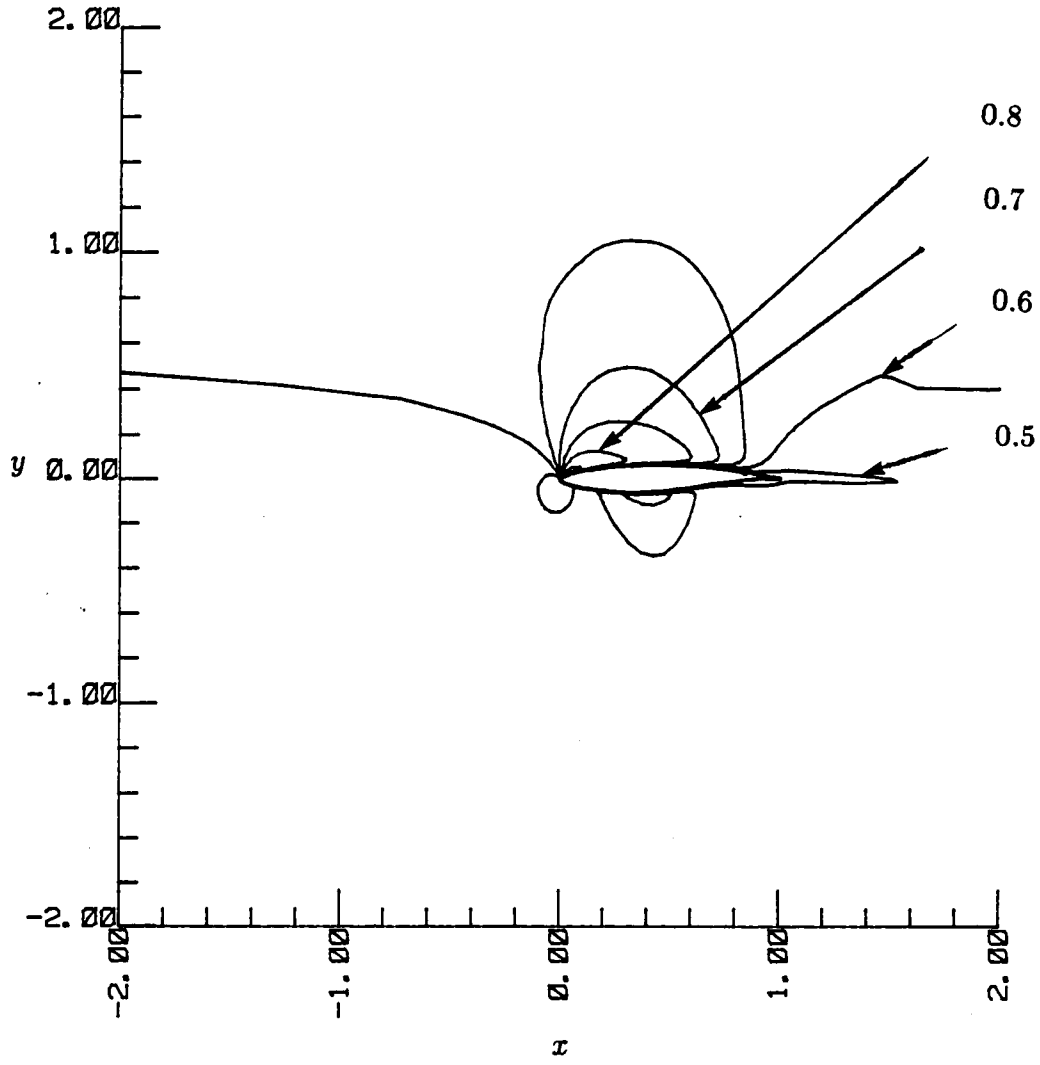


Figure 21: Mach Contours of Flow Field (Test Case 3)

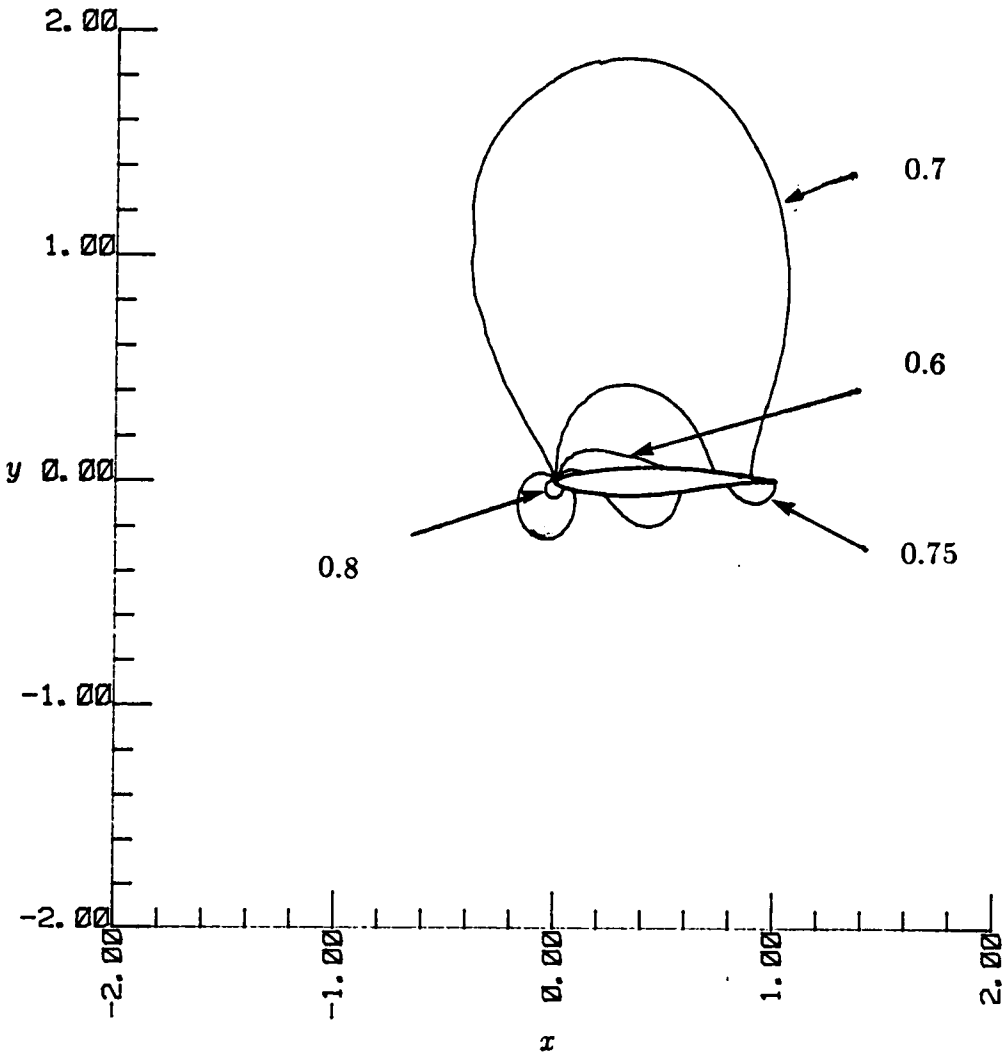


Figure 22: Pressure Contours of Flow Field (Test Case 3)

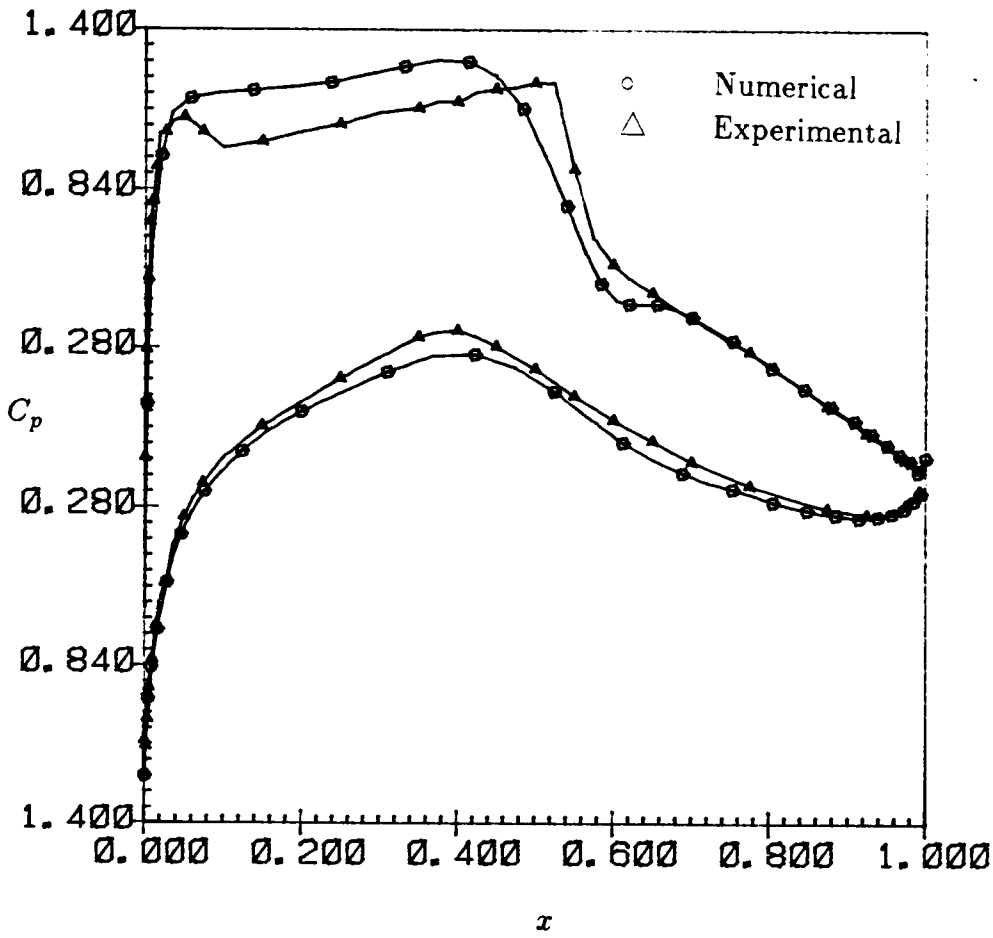


Figure 23: Comparison of Numerical and Experimental Pressure Distributions for Test Case 4 ($Re = 6.5 \times 10^6$, $M_\infty = 0.725$, $\alpha = 2.92$)

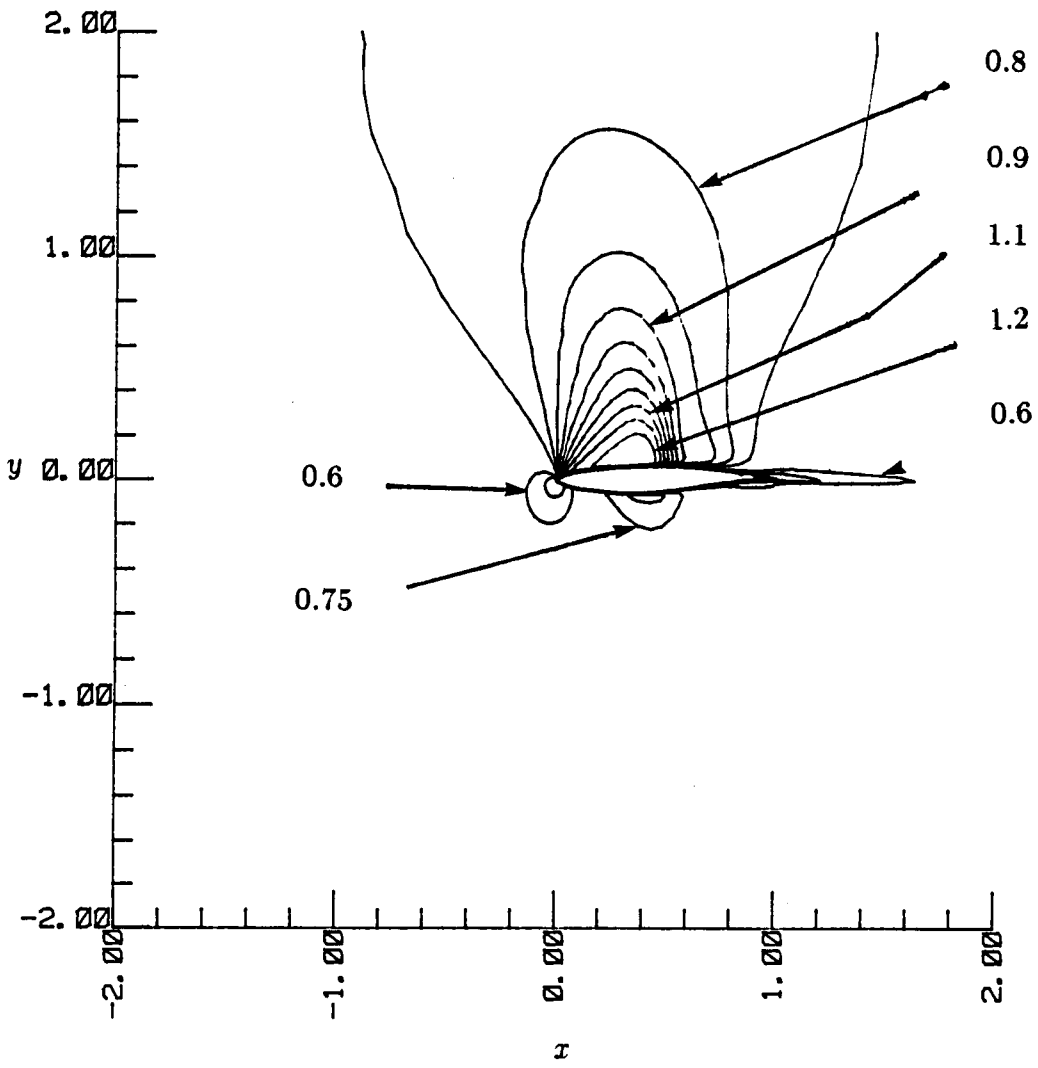


Figure 24: Mach Contours of Flow Field (Test Case 4)

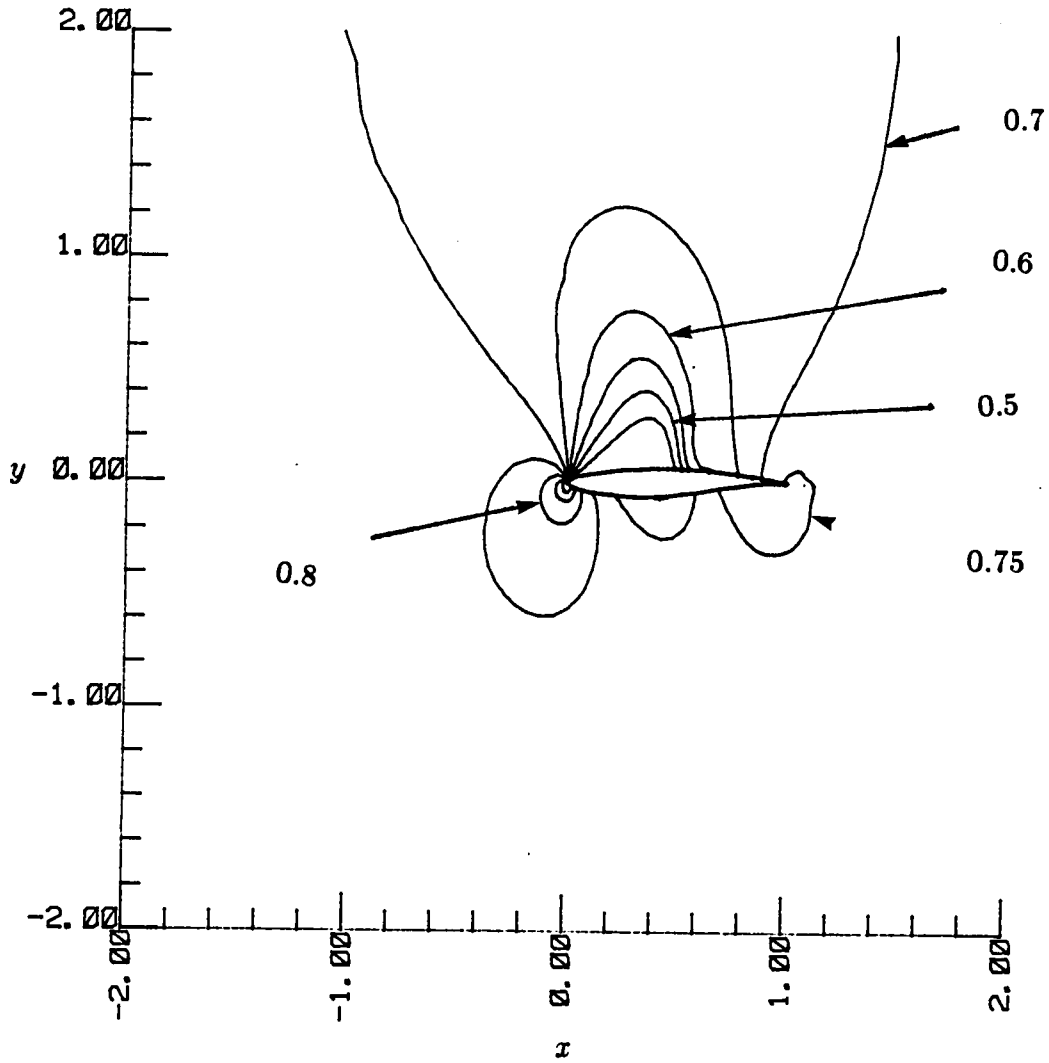


Figure 25: Pressure Contours of Flow Field (Test Case 4)

shows the corresponding pressure contours for the same flow. The next test case is the flow with a smaller Reynolds number, namely, 2.7 x 10^6 and free stream Mach number of 0.73 and angle of attack of 3.19 degrees. Figure 26 compares the surface pressure distribution between experimental data and numerical solution. Figures 27 and 28 show the corresponding Mach and pressure contours for the same flow. In order to evaluate the robustness of the code at low Mach numbers, the code has been run at a free stream Mach number of 0.1, Reynolds number of 1 x 10^6 and at zero angle of attack. The corresponding pressure distribution, Mach and pressure contours are shown in Figures 29, 30 and 31. Having established the robustness of the code from low subsonic flow to transonic flow at reasonable angles of attack it has been decided to test the code for a large angle of incidence. So the code has been run at an angle of attack of 13.5 degrees, Mach number of 0.301 and Reynolds number of 3.91 x 10^6. The corresponding pressure distribution is shown in Figure 32.

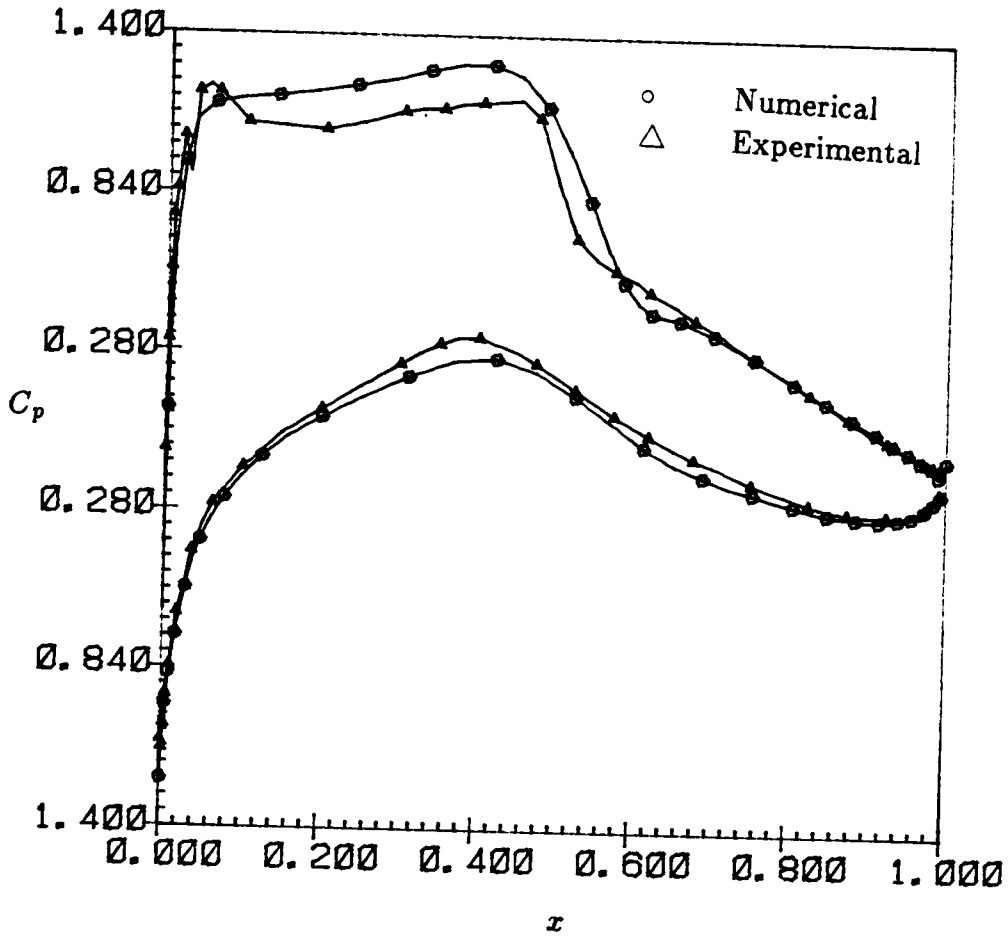


Figure 26: Comparison of Numerical and Experimental Pressure Distributions for Test Case 5 ($Re = 2.7 \times 10^6$, $M_\infty = 0.73$, $\alpha = 3.19$)

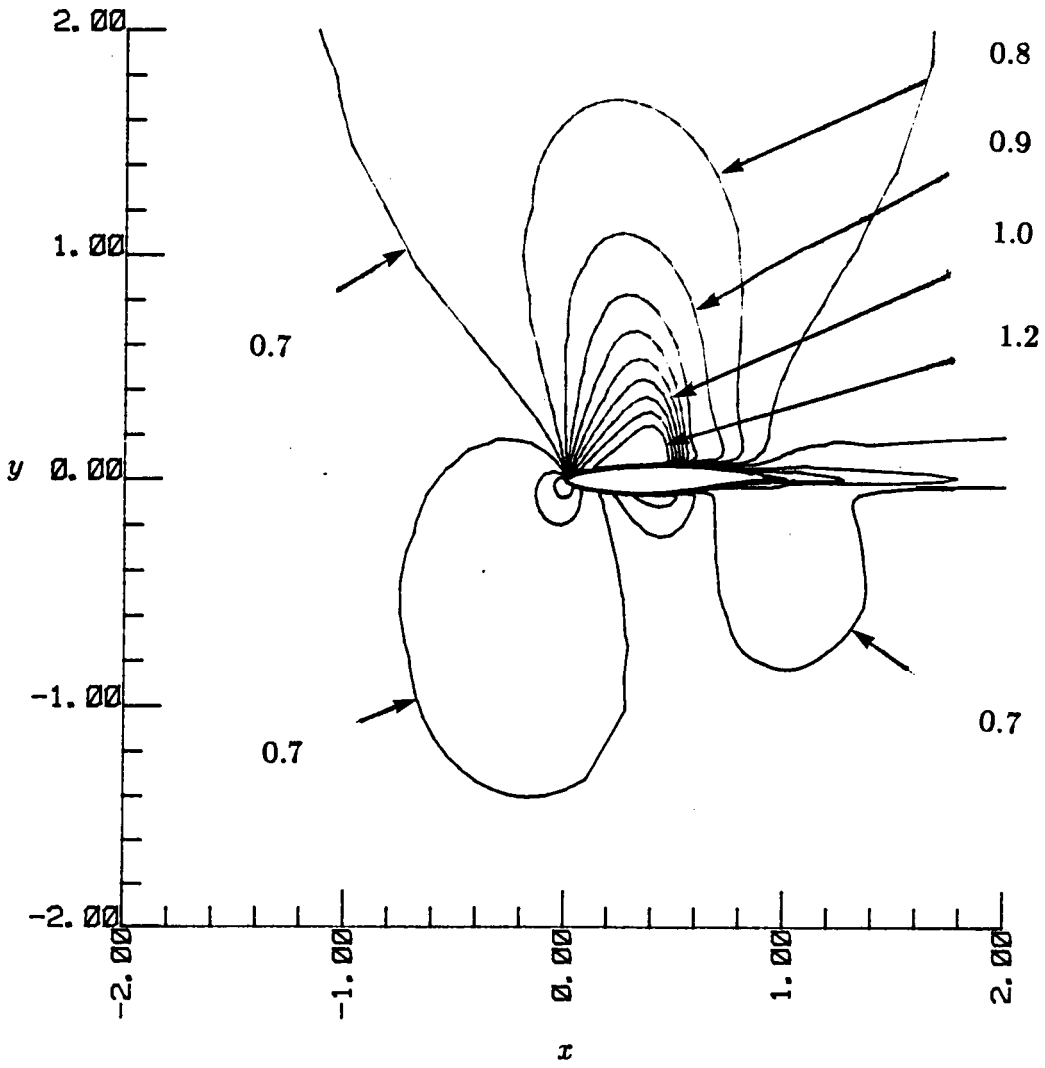


Figure 27: Mach Contours of Flow Field (Test Case 5)

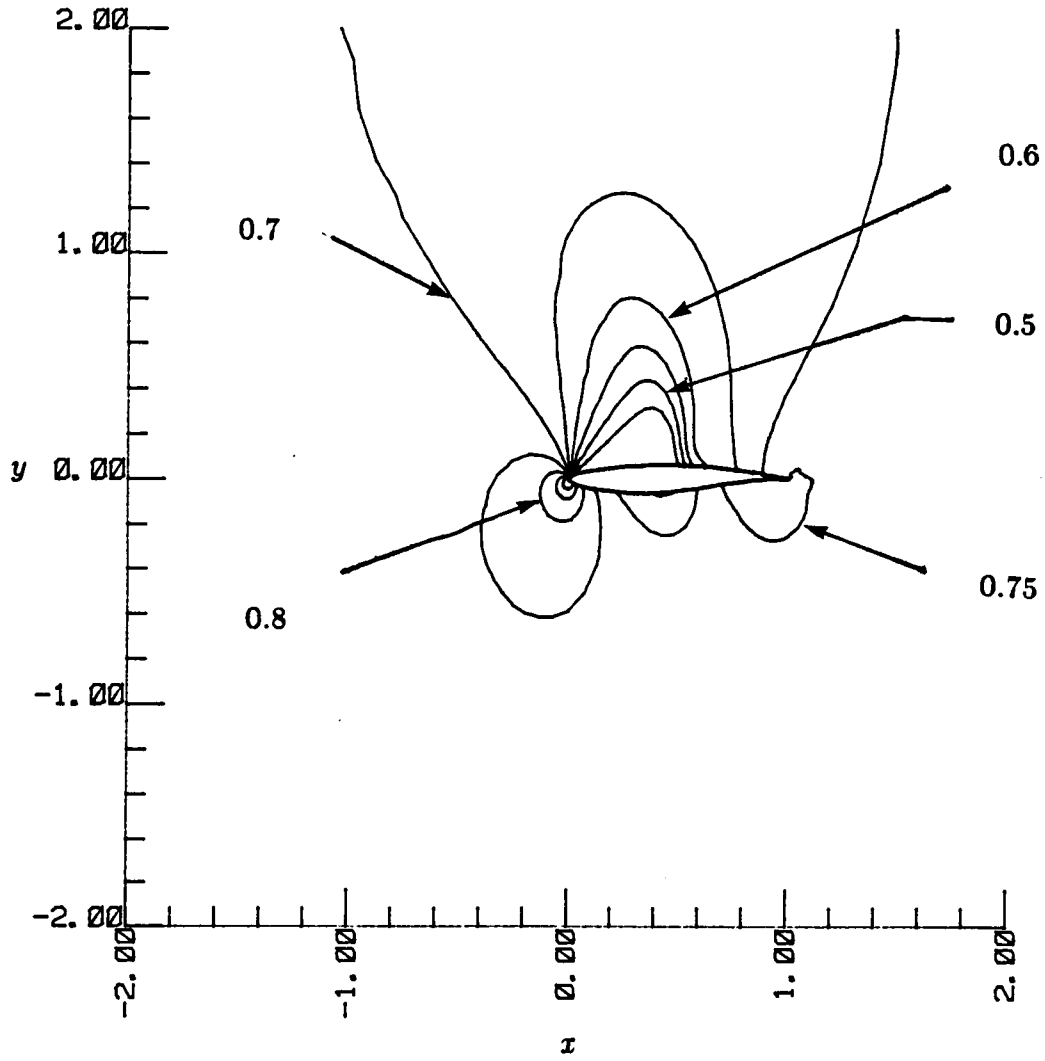


Figure 28: Pressure Contours of Flow Field (Test Case 5)

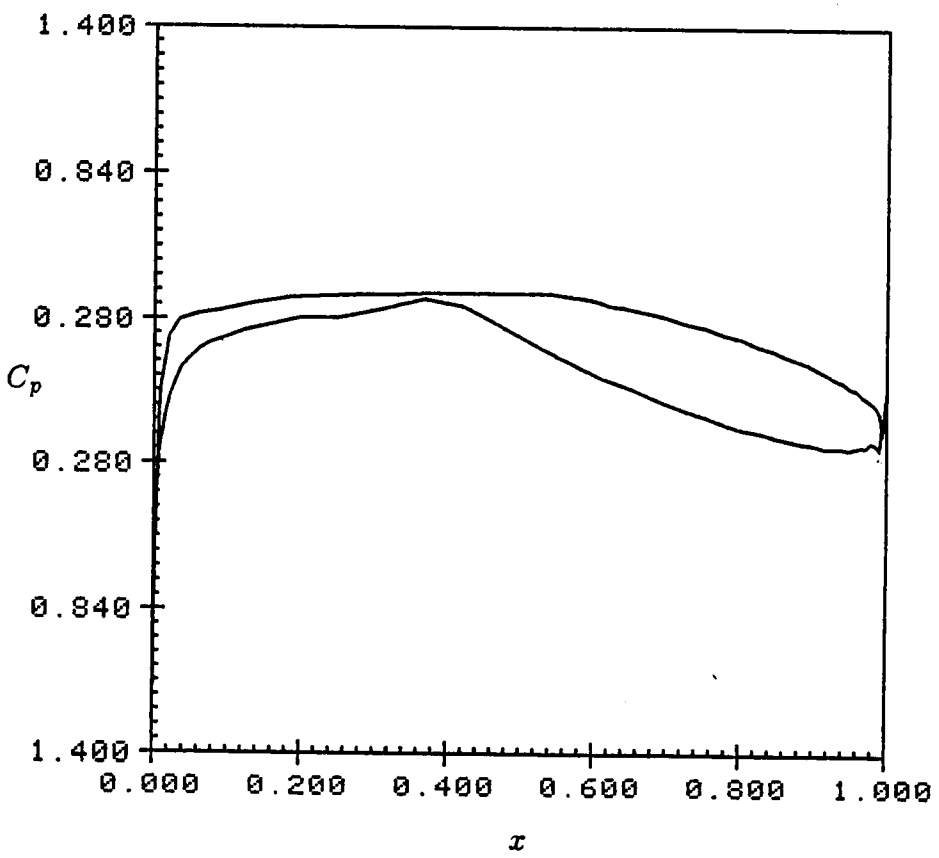


Figure 29: Pressure Distribution of Test Case 6 ($Re = 1 \times 10^6$, $M_\infty = 0.1,\ \alpha = 0.0$)

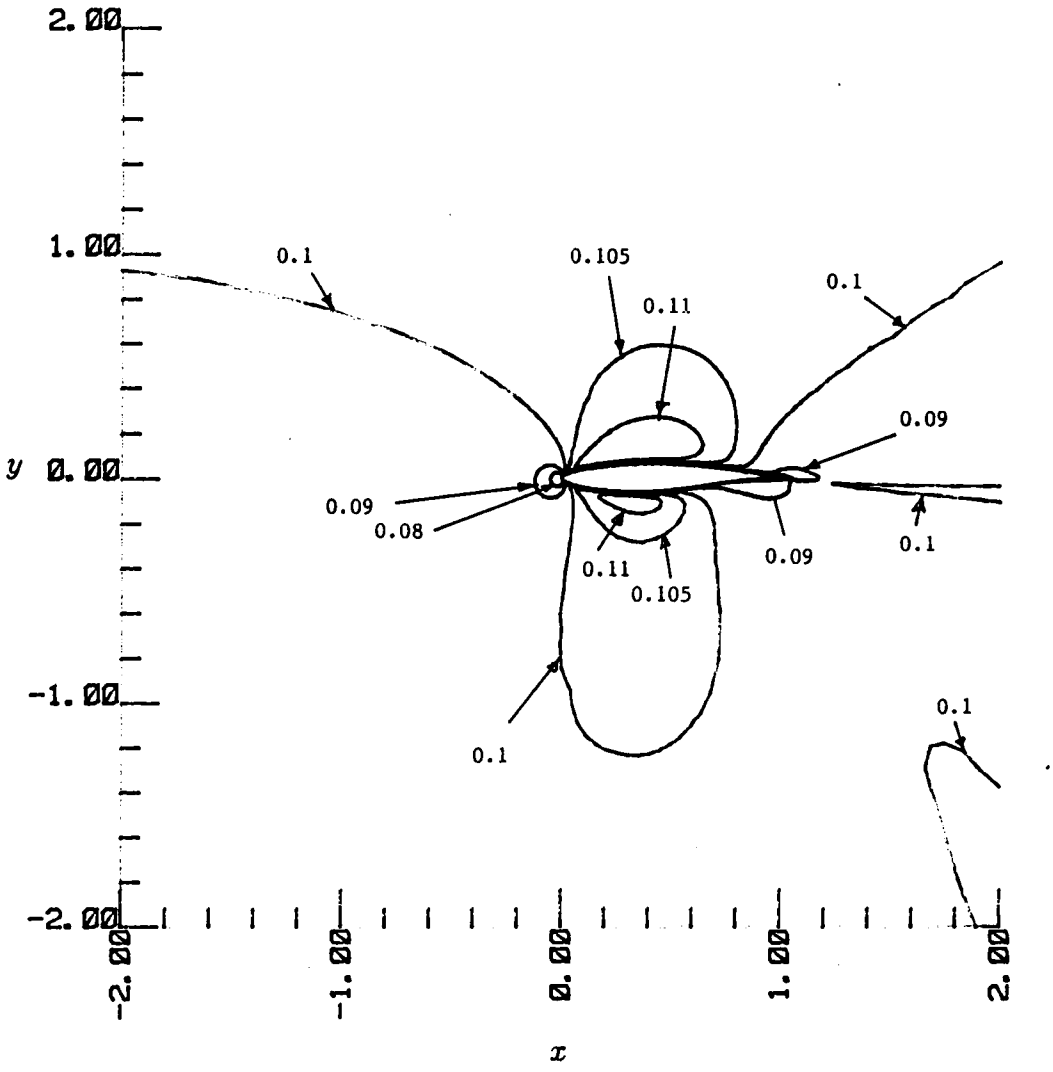


Figure 30: Mach Contours of Flow Field (Test Case 6)

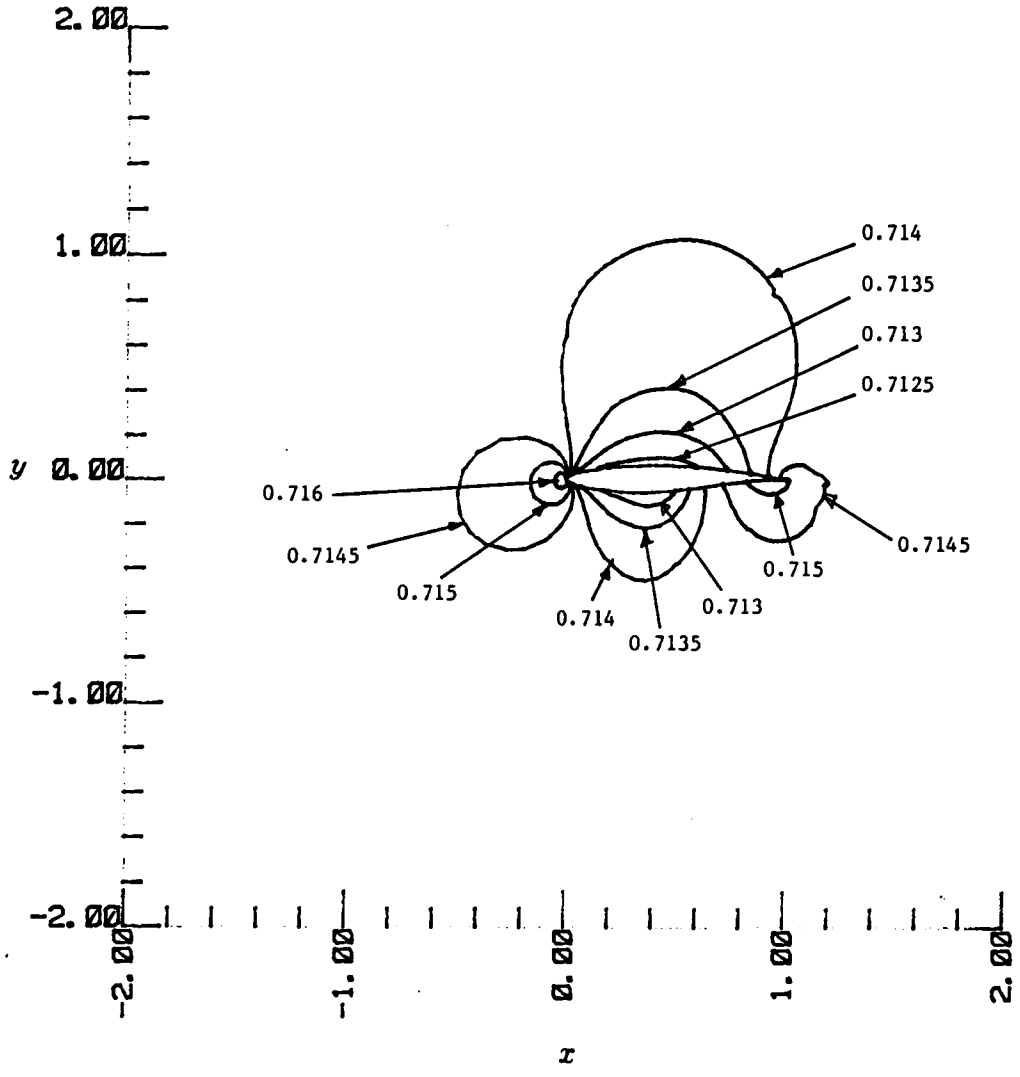


Figure 31: Pressure Contours of Flow Field (Test Case 6)

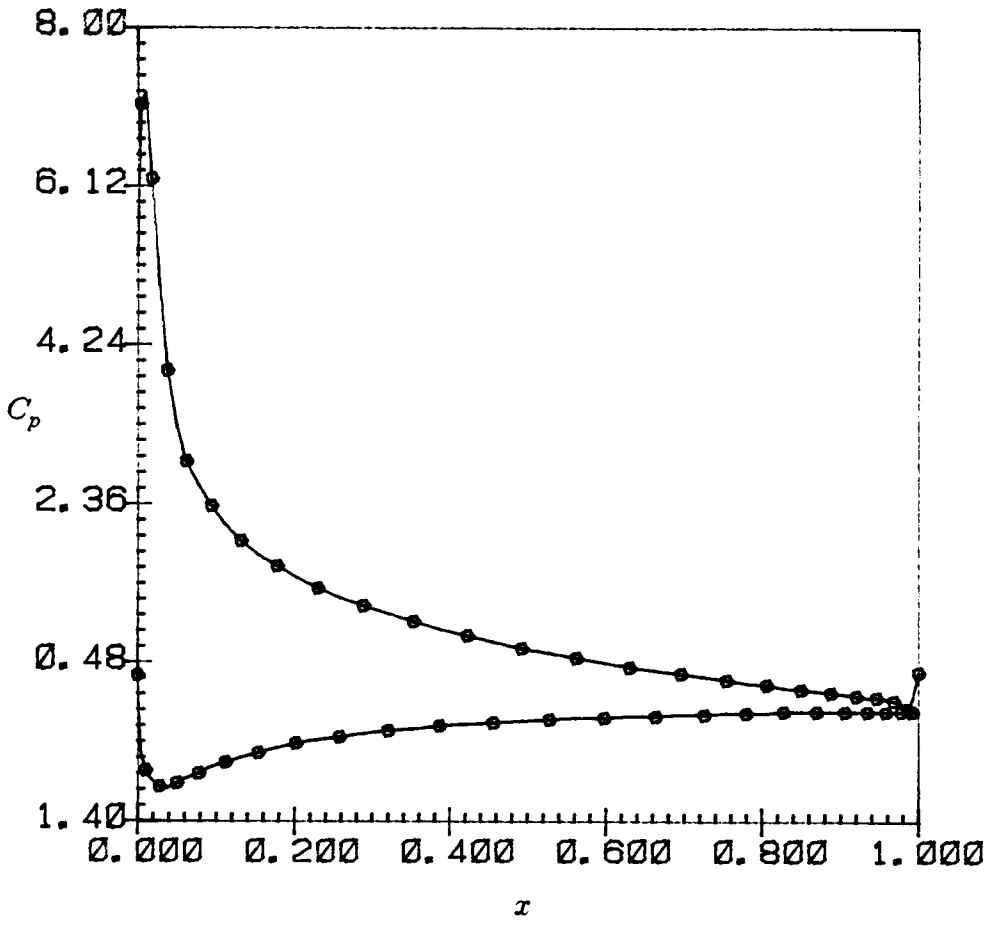


Figure 32: Pressure Distribution of Test Case 7 ($Re = 3.91 \times 10^6$, $M_\infty = 0.301,\ \alpha = 13.5$)

CHAPTER 4

CONCLUSIONS

A locally implicit, globally explicit method has been demonstrated for the solution a of time-dependent model elliptic Equation in one and three dimensions. The same method has been successfully extended to time-dependent Navier-Stokes Equations in generalized curvilinear coordinates with body conforming and stretched grids. It can be applied in finite difference or finite volume formulations in two or three dimensions. Unlike the other implicit schemes, this method does not involve any large Equation solvers. The method works both in single and multigrid modes and is compatible with Jameson's multigrid strategy. There is no restriction on the courant number and the scheme is unconditionally stable.

It has been tested extensively for flows from low subsonic to transonic range and the numerical results have been compared with the experimental results, and the results have been found to be satisfactory. An algebraic turbulence model has been used for high Reynolds number flows. A concept of a new multi-point locally implicit scheme has been introduced and its application to elliptic problems has been found to be satisfactory, both in single and multigrid modes. However, single point scheme seems to be computationally more efficient than multi-point scheme. In conclusion it can be said that a new locally implicit method has been demonstrated to be an efficient technique for solving the Navier-Stokes Equations.

BIBLIOGRAPHY

BIBLIOGRAPHY

1. Jameson, A., "Successes and Challenges in Computational Aerodynamics", AIAA papar 87-1184-cp, 8th Computational Fluid Dynamics Conference, Honolulu, June 1987.

2. Murman, E.M, and Cole, J.D., "Calculation of Plane Steady Transonic Flows", AIAA Journal, 9 , 1971, pp. 114-121.

3. Murman, E.M., "Analysis of Embedded Shock Waves Calculated by Relaxation Methods", AIAA Journal, 12 ,1974, pp. 626-633.

4. Magnus, R., and Yoshihara, H., "Inviscid Transonic Flow Over Airfoils", AIAA Journal, 8 ,1970, pp. 2157-2162.

5. MacCormack, R.W., "The Effect of Viscosity in Hyper- Velocity Impact Cratering", AIAA paper 69-354,1969.

6. Beam, R.W., and Warming, R.F., "An Implicit Finite Difference Algorithm for Hyperbolic System in Conservation Form", J. Comp. Phys., 23, 1976, pp. 87-110.

7. Steger, J.L., "Implicit Finite Difference Simulation of Flow About Arbitrary Two Dimensional Geometries", J. Computational Physics,16 ,1978 , pp. 679-686.

8. Steger, J.L., and Warming, R.F., "Flux vector Splittting of the Inviscid Gas Dynamics Equations with Applications to Finite Difference Methods", J. Comp. Phys., 40, 1981, pp. 263-293.

9. Thomas H. Pulliam, "Euler and Thin-Layer Codes:ARC2D,ARC3D", Computational Fluid Dynamics Workshop held at The University of Tennessee Space Institute, Tullahoma, TN, March 12-16, 1984.

10. Jameson, A., and Turkel, E., "Implicit Schemes and LU Decompositions ", Math. Comp., 37, 1981, pp. 385-397.

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

12. Achi Brandt, "Multi-Level Adaptive Solutions to Boundary Value Problems", Mathematics of Computation, Vol. 31, Number 138, April 1977, pp 333-390.

13. Jameson, A., "Solution of Euler Equations by a Multigrid Method", Applied Math. and Computation, Vol 13, 1983, pp. 327-356.

14. Reddy, K.C., Reddy, J.N., Nayani, S.N.,"A Locally Implicit Scheme for Navier-Stokes Equations", presented at the SSME/CFD Working Group Meeting, NASA Marshall Space Flight Center, April 21-23, 1987.

15. Reddy, K.C., and Jacocks, J.L., "A Locally Implicit Scheme for the Euler Equations", Proceedings of the AIAA 8th Computational Fluid Dynamics Conference, Honolulu, June 1987.

16. Chakravarthy, S.R., "Relaxation Methods for Unfactored Implicit Upwind Schemes", AIAA-84-0165, AIAA 22nd Aerospace Sciences Meeting, January 9-12, Reno, Nevada 1984.

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

18. Jameson, A., "Numerical Solution of the Euler Equations for Compressible Inviscid Fluids", Numerical Methods for the Euler Equations of Fluid Dynamics, Ed. F.Angrand, et al., SIAM, 1985, pp. 199-245.

19. Jameson, A., "Solution of Euler Equations for Two-Dimensional Transonic Flow by a Multigrid Method", Applied Mathematics and Computation, Vol. 13, 1983, pp. 327-355.

20. Sorenson, R.L., "A Computer Program to Generate Two-Dimensional Grids About Airfoils and other Shapes by the Use of Poisson's Equation", NASA TM 81198, May 1980.

21. Steger, J.L., and Chaussee, D.S., "Generation of Body-Fitted Coordinates Using Hyperbolic Partial Differential Equations", SIAM, J. Scientificc and Statistical Computing, 1, 1980, pp. 431-437.

22. Yoon, S., and Antony Jameson, "An LU-SSOR Scheme for the Euler and Navier-Stokes Equations", AIAA 25th Aerospace Sciences Meeting, Reno, Nevada, January 12-15, 1987.

23. Cook, P.H., McDonald, M.A., and Firmin, M.C.P, "Aerofoil RAE2822-Pressure Distributions, and Boundary Layer and Wake Measurements", AGARD-AR-138, 1979.

24. Baldwin, B.S., and Lomax, H., "Thin Layer Approximation and Algebraic Model for Seperated Turbulent Flows", AIAA paper 78-257, January 1978.

25. Martinelli, L., Jameson, A., and Grasso, F., "A Multigrid Method for the Navier-Stokes Equations", AIAA 24th Aerospace Sciences Meeting, Reno, Nevada, January 6-9, 1986.

26. Chima, R.V., Turkel, E., and Schaffer,S., "Comparision of Three Explicit Multigrid Methods for the Euler and Navier-Stokes Equations", AIAA 25th Aerospace Sciences Meeting, Reno, Nevada, January 12-15, 1987.

27. Agarwal, R.K., and Deese, J.E., "Numerical Solution of Shock/Bondary Layer Interaction Using a Finite Volume Runge-Kutta Time-Stepping Scheme", 3rd Symposium on Numerical and Physical Aspects of Aerodynamic Flows, Long Beach, California, 20-24 January, 1985.

28. Agarwal, R.K., and Deese, J.E., "Computation of Transonic Viscous Airfoil, Inlet, and Wing Flowfields", AIAA 17th Fluid Dynamics, Plasma Dynamics, and Lasers Conference, Snowmass, Colorado, June 25-27, 1984.

29. Beam, R.M., and Warming, R.F., "An Implicit Factored Scheme for the Compressible Navier-stokes Equtions", AIAA Journal, April 1978.

30. Coakley, T.J., "Numerical Solution of Viscous Transonic Airfoil Flows", AIAA 25th Aerospace Sciences Meeting, Reno, Nevada, January 12-15, 1987.

31. Maksymiuk, C.M., and Pulliam, T.H., "Viscous Transonic Airfoil Workshop Results Using ARC2D", AIAA 25th Aerospace Sciences Meeting , Reno, Nevada, January 12-15, 1987.

32. Coakley, T.J., "A Compressible Navier-Stokes Code for Turbulent Flow Modeling", Computational Fluid Dynamics Workshop held at The University of Tennessee Space Institute, Tullahoma, TN, March 12-16, 1984.

33. Mavriplis, D., and Jameson, A., "Multigrid solution of the Two-Dimensional Euler Equations on Unstructured Triangular Meshes" , AIAA 25th Aerospace Sciences Meeting, Reno, Nevada, January 12-15 , 1987.

34. Jameson, A., and Baker, T.J., "Improvements to the Aircraft Euler Method", AIAA 25th Aerospace Sciences Meeting, Reno, Nevada, January 12-15, 1987.

35. Antony Jameson, "A Vertex Based Multigrid Algorithm for Three Dimensional Compressible Flow Calculations", ASME symposium on Numerical Methods for Compressible Flow, Annaheim, CA, December 1986

36. South Jr., J.C., and Brandt, A., "Application of a Multi- Level Grid Method to Transonic Flow Calculations", ICAASE Report 76-8, NASA Langley Research Center, Hampton, Virginia, 1976.

37. Agarwal, R.K., and Deese, J.E., "Euler Solutions for Flow Past an Airfoil from Subsonic to High Supersonic Mach Numbers", AIAA 6th Computational Fluid Dynamics Conference, Danvers, Massachusetts, July 13-15, 1983.

38. Stone, H.L., "Iterative Solution of Implicit Approximations of Multidimensional Partial Differential Equations", SIAM J. Numer. Anal., Vol. 5, No. 3, September 1968.

39. Rob Kettler, "Analysis and Comparision of Relaxation Schemes in Robust Multigrid and Preconditional Conjugate Gradient Methods" , edited by Hackbusch and Trottenberg, Springer Verlag, 1982.

40. Anutosh Moitra, Eli Turkel, Ajay Kumar, "Application of a Runge-Kutta Scheme for High Speed Inviscid Internal Flows", ICASE Report No. , NASA Langley Research Center, Hampton, VA.

41. Parter, S.V., and Steuerwalt, M., "Block Iterative Methods for Elliptic and Parabolic Difference Equations", SIAM J. Numer. Anal., Vol 19, No. 6, December 1982.

42. Caughey, D.A., "A Diogonal Implicit Multigrid Algorithm for the Euler Equations", AIAA 25th Aerospace Sciences Meeting, Reno, Nevada, January 12-15, 1987.

43. Thompson, J., "Numerical Grid Generation", J.Thompson Ed., North Holland, 1982.

44. Schlichting, H., "Boundary Layer Theory", Fourth Edition, Mcgraw Hill, New York, 1968.

45. Ratcliff, M.L., "A Locally Implicit Scheme for Euler Equations", M.S. Thesis to be submitted, The University of Tennessee Space Institute, Tullahoma, TN.

46. Reddy, K.C., and Nayani, S.N., "A Locally Implicit Scheme for Elliptic Partial Differential Equations", presented at the SSME/CFD Working Group Meeting, NASA Marshall Space Flight Center, April 8-11, 1986.

APPENDIXES

APPENDIX 1

MULTI-POINT LOCALLY IMPLICIT SCHEME

One Dimensional Diffusion Equation

A 2–point scheme for a one dimensional problem and an 8–point scheme for a three dimensional problem are described along with the stability analysis as opposed to one point scheme described earlier.

Consider the one dimensional diffusion Equation,

$$\frac{\partial u}{\partial t} = \nu \frac{\partial^2 u}{\partial x^2} \tag{79}$$

with arbitrary initial conditions and Dirichlet boundary conditions. Using Finite difference approximation and Euler Implicit Scheme, Equation (1) can be discretized as,

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = \nu \frac{u_{j-1}^{n-1} - 2u_j^{n+1} + u_{j+1}^{n+1}}{\Delta x^2} \tag{80}$$

2–Point Locally Implicit Scheme

Writing the difference Equation at j and $j+1$ and denoting $R = \nu \frac{\Delta t}{\Delta x^2}$,

$$(1+2R)u_j^{n+1} - Ru_{j+1}^{n+1} = u_j^n + Ru_{j-1}^{n+1}$$

$$-Ru_j^{n+1} + (1+2R)u_{j+1}^{n+1} = u_{j+1}^n + Ru_{j+2}^{n+1}$$

Now, approximate Ru_{j+2}^{n+1} by Ru_{j+2}^n and solve for u_j^{n+1} and u_{j+1}^{n+1} ,

$$j = 2, 4, 6, - - - -$$

Delta Form of the Difference Equations

Denoting $\Delta u_j^n = u_j^{n+1} - u_j^n$ (81)

and $Res_j^n = (u_{j-1}^n - 2u_j^n + u_{j+1}^n)R$ (82)

the difference Equations can be written in the Δ form as,

$$(1+2R)\Delta u_j^n - R\Delta u_{j+1}^n = Res_j^n + R\Delta u_{j-1}^n \tag{83}$$

$$-R\Delta u_j^n + (1+2R)\Delta u_{j+1}^n = Res_{j+1}^n + R\Delta u_{j+2}^n \tag{84}$$

Approximations for Δu_{j+2}^n

(i) $$\Delta u_{j+2}^n = 0$$

Loses time accuracy but unconditionally stable. Asymptotic steady state solution is not affected.

(ii) $$\Delta u_{j+2}^n = R^*(u_{j+1}^n - 2u_{j+2}^n + u_{j+3}^n), \quad R^* \simeq 1$$

It was observed that the scheme is time accurate for $R^* = 1$ and stable for any R. Asymptotic steady state solution was not affected and convergence to steady state solution was found to be faster than in case (i).

Now, as before solve for Δu_j^n and Δu_{j+1}^n, for $j = 2, 4, 6, - - - - - - -$.

Stability

Denote $u_j = \lim_{n\to\infty} u_j^n$ (85)

and $e_j^n = u_j^n - u_j$ (86)

Substituting, approximation (i) becomes

$$(1+2R)e_j^{n+1} - Re_{j+1}^{n+1} = e_j^n + Re_{j-1}^{n+1}$$

$$-Re_j^{n+1} + (1+2R)e_{j+1}^{n+1} = e_{j+1}^n + Re_{j+2}^n$$

Denote $D = \frac{R}{(1+2R)}$, $0 < D < \frac{1}{2}$ (87)

Solving for e_{j+1}^{n+1},

$$(1-D^2)e_{j+1}^{n+1} = D^2 e_{j-1}^{n+1} + \frac{D^2}{R}e_j^n + \frac{D}{R}e_{j+1}^n + De_{j+2}^n$$

$$(1-D^2)|e_{j+1}^{n+1}| \leq (\frac{D^2}{R} + \frac{D}{R} + D)\|e\|_\infty^n + D^2|e_{j-1}|^{n+1}$$

Now,

$$|e_{j+1}^{n+1}| \leq \|e\|_\infty^n,$$

$$|e_j^{n+1}| \leq \|e\|_\infty^n \ , \ j = 2,4,6, - - - - .$$

and

$$\|e\|_\infty^{n+1} \leq \|e\|_\infty^n \tag{88}$$

Hence, the scheme is unconditionally stable. The convergence history of the one dimensional diffusion Equation for a case of 80 cells is shown in Figure 33 for both single grid and multi-grid cases.

Three Dimensional Diffusion Equation

$$\frac{\partial u}{\partial t} = \nu \nabla^2 u \tag{89}$$

Dirichlet boundary conditions are applied at the grid boundaries and random numbers are chosen as the initial condition.

Writing Euler implicit scheme for the Equation as before,

$$\frac{1}{\Delta t}\Delta u_{i,j,k}^n = \frac{\nu}{h^2}(\delta_x^2 + \delta_y^2 + \delta_z^2)u_{i,j,k}^{n+1} \tag{90}$$

Δ form of the above Equation is,

$$\{I - R(\delta_x^2 + \delta_y^2 + \delta_z^2)\}\Delta u_{i,j,k}^n = Res_{i,j,k}^n$$

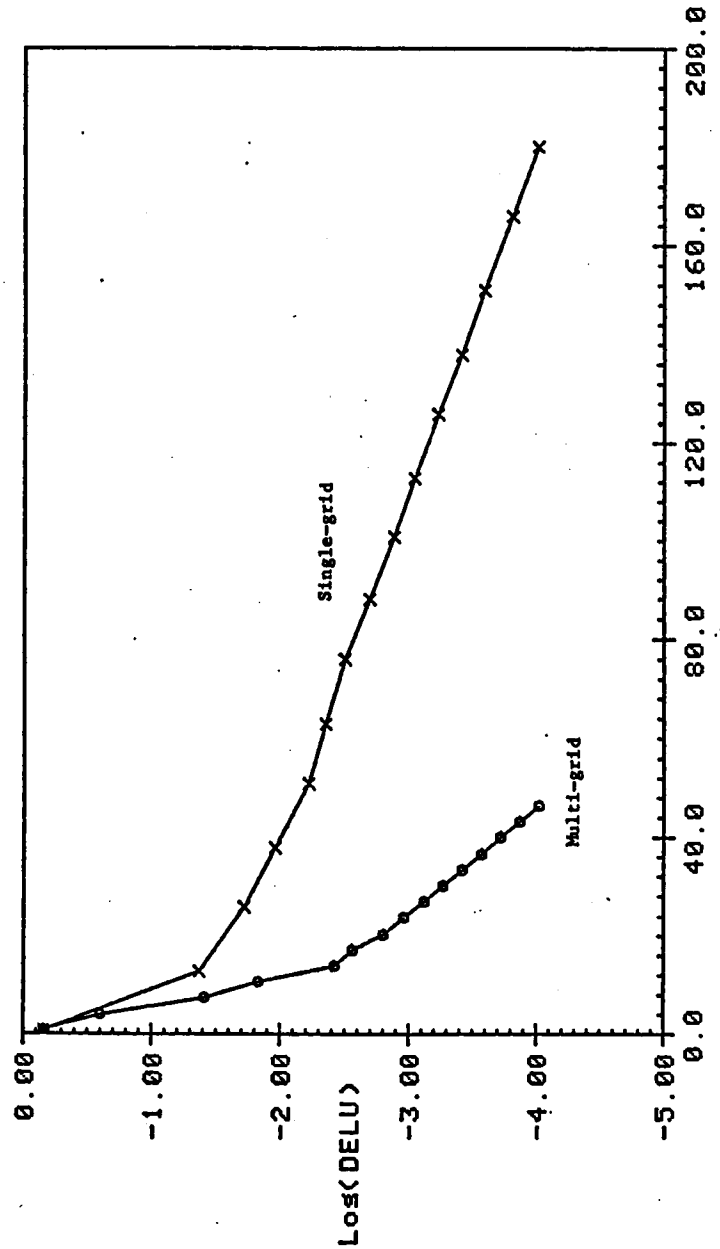


Figure 33: Convergence History of 1–D Diffusion Equation (80 Cells)

where

$$Res^n_{i,j,k} = R(\delta_x^2 + \delta_y^2 + \delta_z^2)u^n_{i,j,k} \tag{91}$$
$$R = \frac{\nu \Delta t}{h^2}$$

and

$$h = \Delta x = \Delta y = \Delta z$$

8-point Locally Implicit Scheme

The schematic representation of the 8-point Locally Implicit scheme is as shown in Figure 34. Writing the difference Equation at node 1 (i, j, k):

$$(1 + 6R)\Delta u_1 - R(\Delta u_2 + \Delta u_3 + \Delta u_5) = RHS_1 \tag{92}$$

where

$$RHS1 = Res_1 + R(\Delta u_{i-1,j,k} + \Delta u_{i,j-1,k} + \Delta u_{i,j,k-1}) \tag{93}$$

Approximate the difference Equation at node $2(i + 1, j, k)$:

$$(1 + 6R)\Delta u_2 - R(\Delta u_1 + \Delta u_4 + \Delta u_6) = RHS2 \tag{94}$$

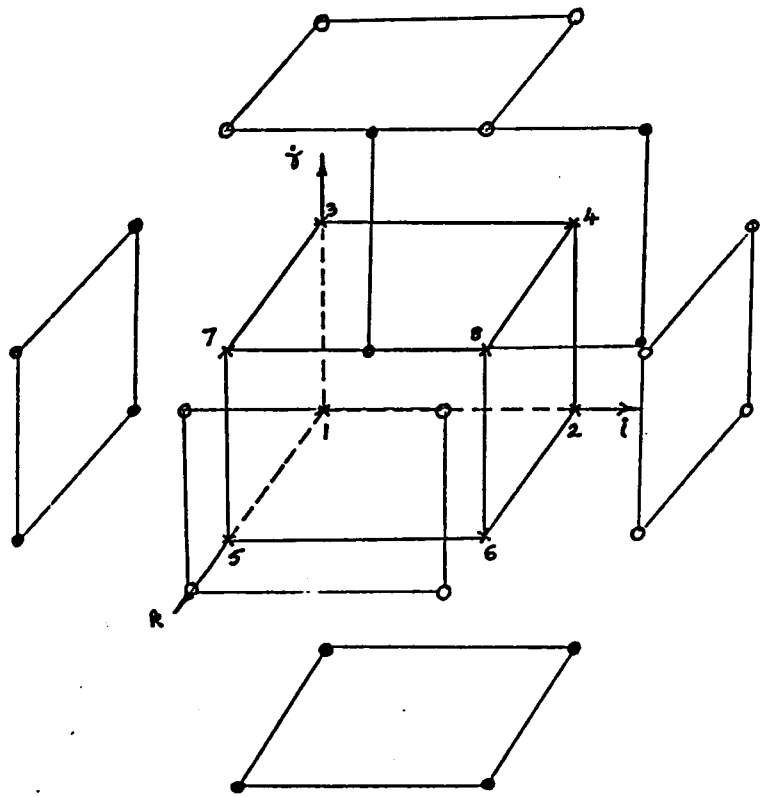
where

$$RHS2 = Res_2 + R(\Delta u_{i+1,j-1,k} + \Delta u_{i+1,j,k-1} + \Delta u^{(e)}_{i+2,j,k}) \tag{95}$$

and

$$\Delta u^{(e)}_{i+2,j,k} = Res^n_{i+2,j,k}/R \tag{96}$$

Similar difference Equations are written at the nodes 3,4,5,6,7,8 (corners of one cell).



- • Latest DELU
- ○ Explicit DELU

Node	
1	i, j, k
2	$i+1, j, k$
3	$i, j+1, k$
4	$i+1, j+1, k$
5	$i, j, k+1$
6	$i+1, j, k+1$
7	$i, j+1, k+1$
8	$i+1, j+1, k+1$

Figure 34: Schematic Representation of Locally Implicit Scheme

Writing in the matrix form,

$$\begin{bmatrix} & & \\ & & \\ & & \\ & & \end{bmatrix} \begin{bmatrix} \Delta u_1 \\ - - - \\ - - - \\ - - - \\ - - - \\ \Delta u_8 \end{bmatrix} = \begin{bmatrix} RHS_1 \\ - - - \\ - - - \\ - - - \\ - - - \\ RHS_8 \end{bmatrix}$$

$\Delta u_1, - - - - - -, \Delta u_8$ are computed for each cell, in the increasing directions of i, j, k and the solution is updated.

$$u_{i,j,k}^{n+1} = u_{i,j,k}^{n} + \omega \Delta u_1^n, \quad \text{etc.} \tag{97}$$

where ω is the relaxation factor.

The convergence histories for the grid sizes of (20 x 20 x 20), (40 x 40 x 40), (80 x 40 x 40) are shown in Figures 35, 36, 37 and 38. Figure 39 shows the comparison of the numerical and exact solutions of a poissons' Equation, the forcing function being $\sin \pi x \sin 2\pi y \sin 3\pi z$. The error margin in the above comparison was found to be less than 0.6%. The plot of % error against grid length is shown in Figure 40 and the convergence history in Figure 41.

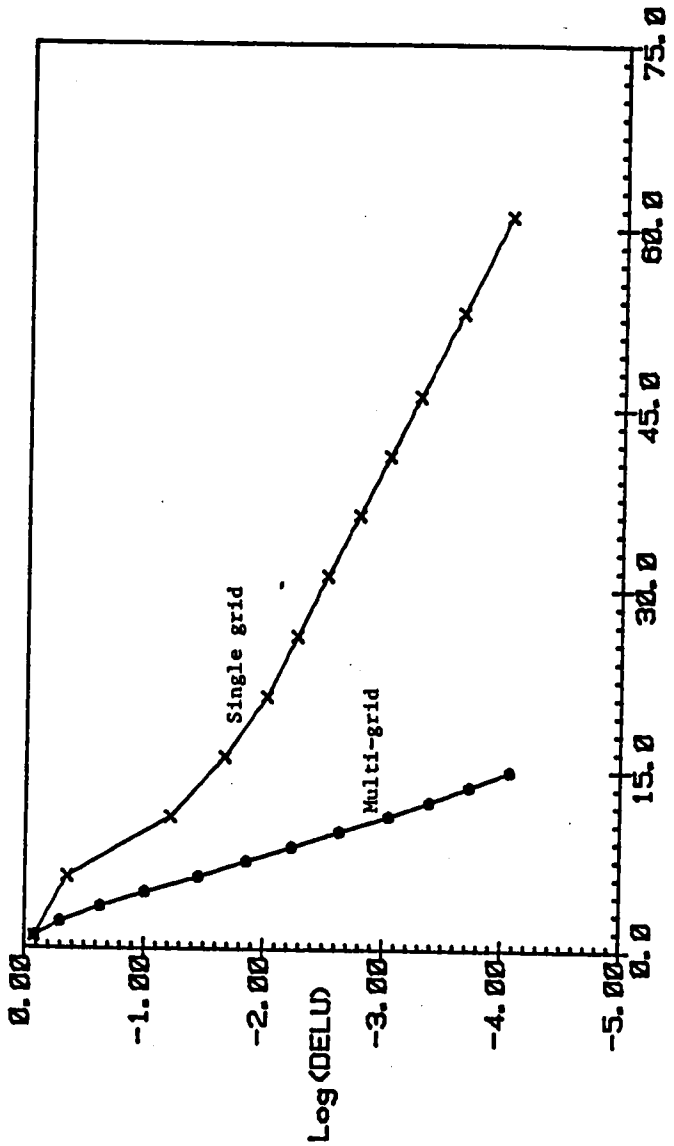


Figure 35: Convergence History of 3–D Diffusion Equation (20 x 20 x 20)

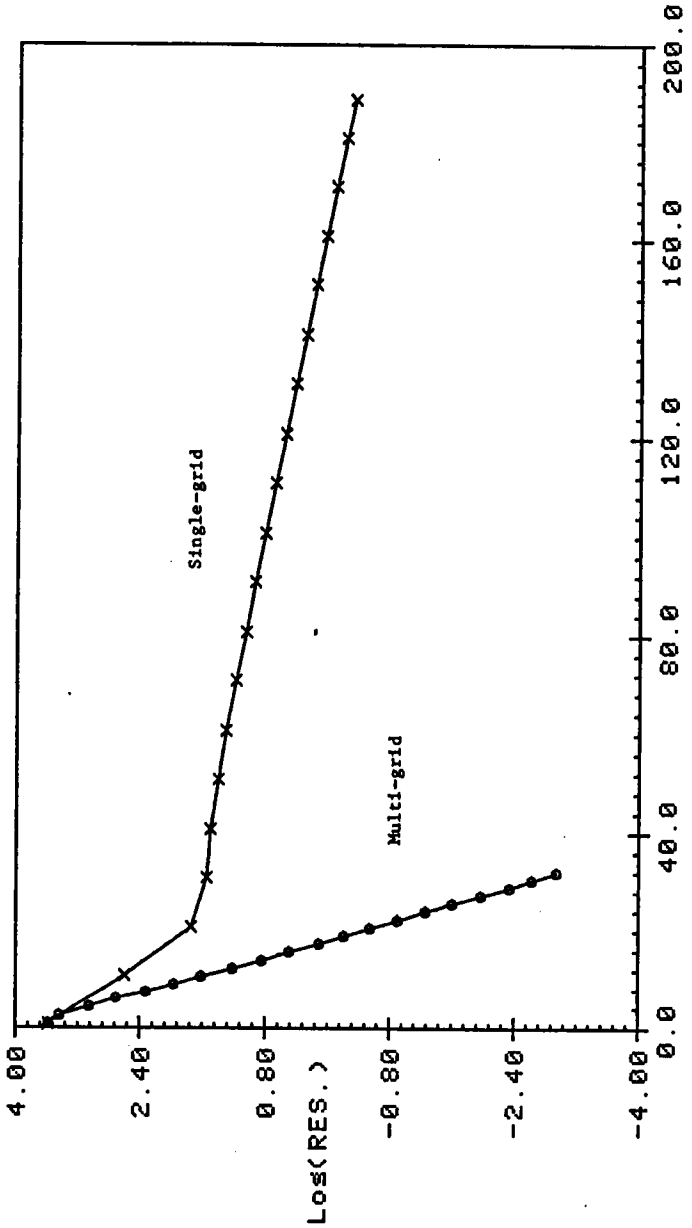


Figure 36: Convergence History of 3–D Diffusion Equation (40 x 40 x 40)

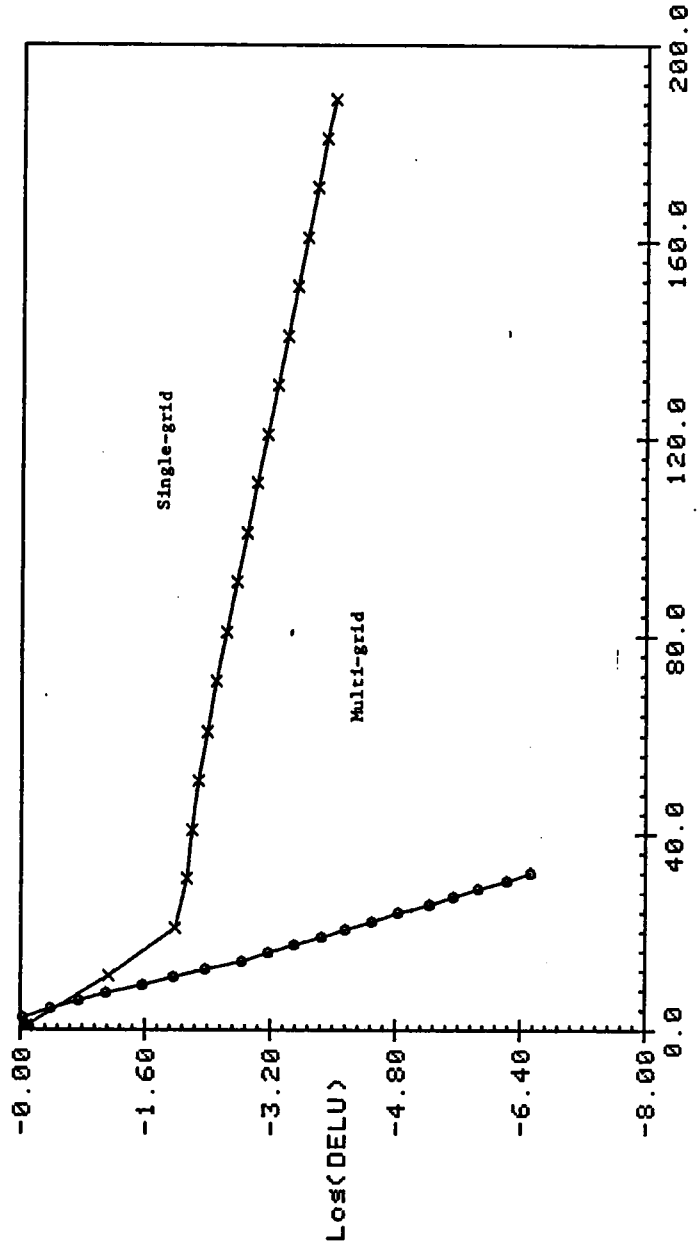


Figure 37: Convergence History of 3-D Diffusion Equation (40 x 40 x 40)

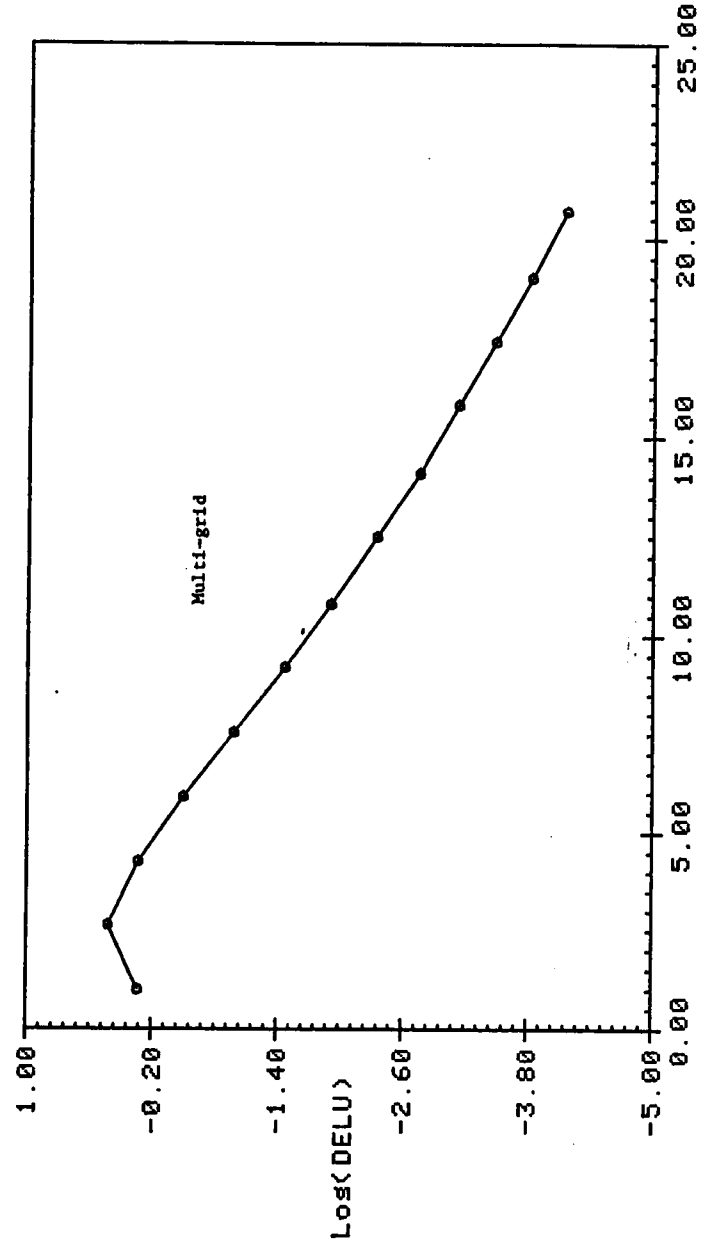


Figure 38: Convergence History of 3-D Diffusion Equation (80 x 40 x 40)

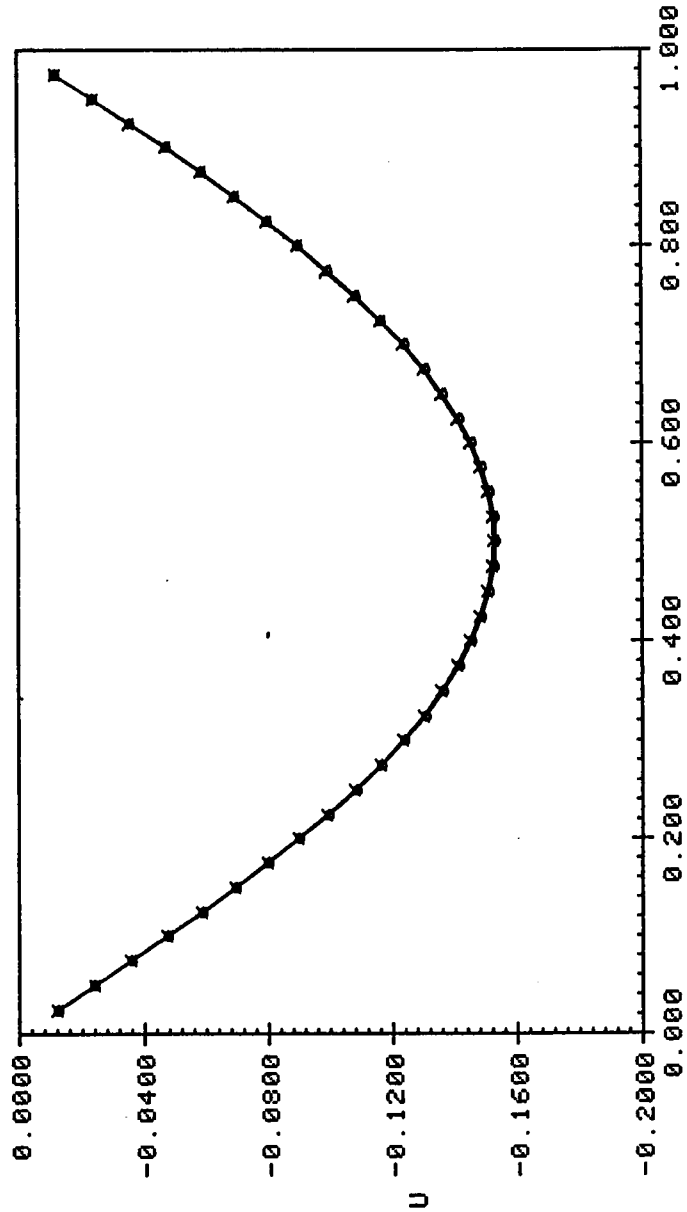


Figure 39: Comparison of the Numerical and Exact Solutions of the Poissons' Equation (40 x 40 x 40)

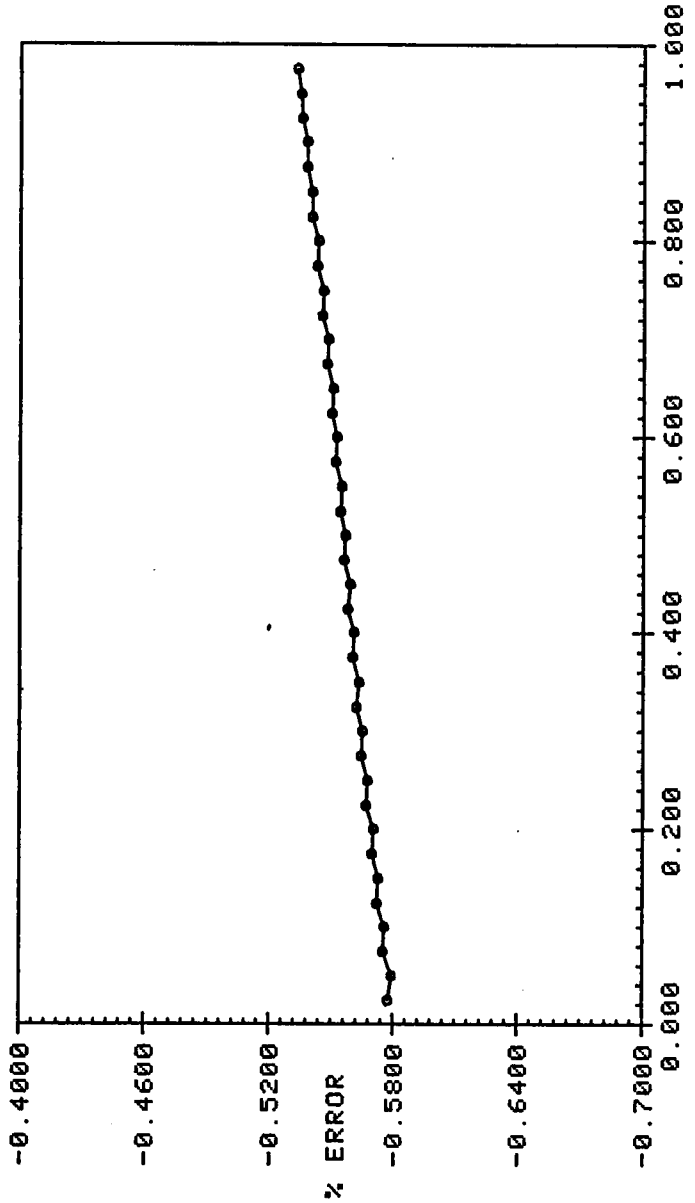


Figure 40: Percentage Error Between Numerical and Exact Solutions of the Poissons' Equation

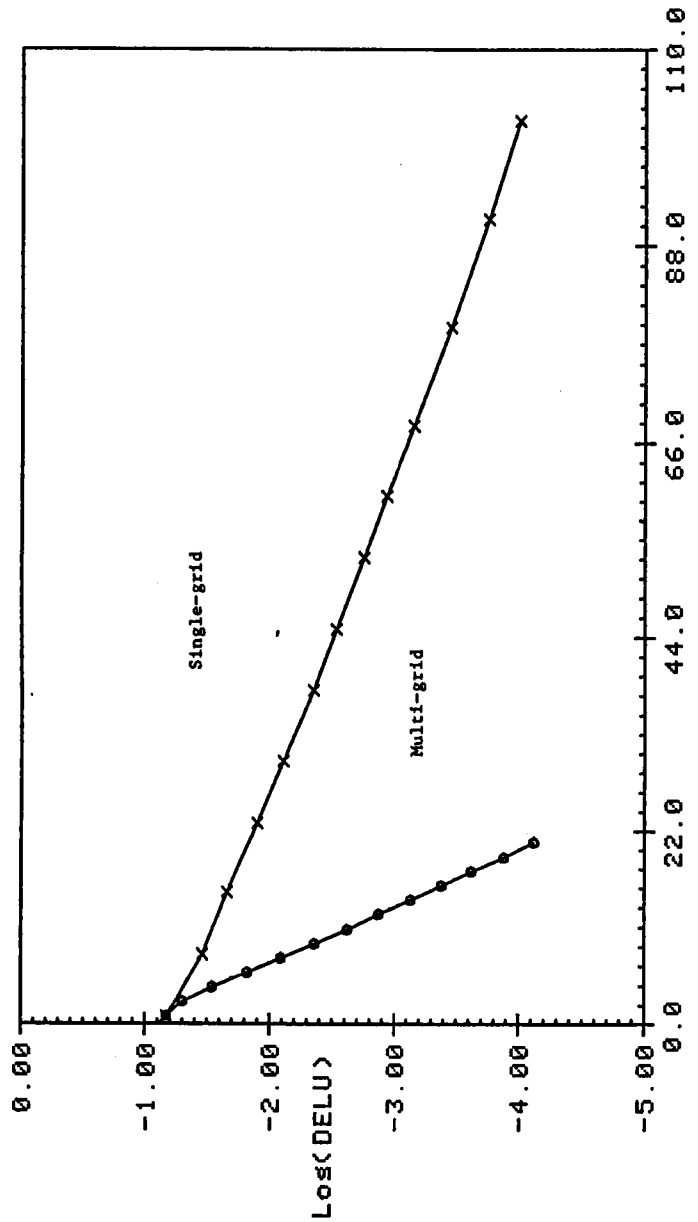


Figure 41: Convergence History of the Test Case With Poissons' Equation

APPENDIX 2

ALGEBRAIC TURBULENCE MODEL

An algebraic eddy viscosity model is included in the code to approximate the effect of turbulence. It is a conventional two-layer model. The inner layer is governed by the Prandtl–Van Driest formulation, and the outer layer follows the modified Clauser approximation. The turbulence model is detailed Baldwin and Lomax [22] and was designed specifically for use with the thin layer approxmiation. The model is most appropriate to attached and mildly separated boundary layers. No attempt is made to model wake regions and massively separated flows. It has been very successful for computing boundary layer growth, shock–boundary layer interation, separation points or lines and other boundary layer properties.

The effects of turbulence are simulated in terms of an eddy viscosity coefficient μ_t. Thus, in stress terms of the laminar Navier–Stokes Equations, the molecular coefficient of viscosity μ is replaced by $\mu + \mu_t$. In heat flux term $k/c_p = \mu/p_r$ is replaced by $\mu/p_r + \mu_t/p_{rt}$. The model avoids the necessity for finding the edge of the boundary layer. The turbulent viscosity μ_t is given by,

$$\mu_t = \begin{cases} (\mu_t)_{\text{inner}} & y \leq y_{\text{crossover}} \\ (\mu_t)_{\text{outer}} & y_{\text{crossover}} < y \end{cases} \tag{95}$$

where y is the normal distance from the wall and $y_{\text{crossover}}$ is the smallest value of y at which values from the inner and outer formulas are equal. The Prandtl–Van Driest formulation is used in the inner region,

$$(\mu_t)_{\text{inner}} = \rho\, l^2\, |\omega| \tag{99}$$

where

$$l = ky\left[1 - \exp\left(-y^+/A^+\right)\right] \tag{100}$$

magnitude of the vorticity $|\omega| = \frac{\partial u}{\partial y} - \frac{\partial v}{\partial x}$ (101)

$$y^+ = \frac{\rho_w u_\tau y}{\mu_w}$$

u = velocity parallel to the solid surface

u_t = friction velocity $\sqrt{\tau_w/\rho_w}$

τ_w = shear stress at the wall

y^+ = law of the wall coordinate

The turbulent viscosity in the outer region is defined by,

$$(\mu_t)_{\text{outer}} = K\ C_{cp}\ \rho\ F_{\text{wake}}\ F_{\text{Kleb}}(y) \tag{102}$$

where K is the Clauser constant, C_{cp} is an additional constant, and

$$F_{\text{wake}} = \left\{ \begin{array}{c} y_{\max}\ F_{\max} \\ \text{or} \\ C_{Wk}\ y_{\max}\ U_{\text{Dif}}^2/F_{\max} \end{array} \right\} \text{ the smaller} \tag{103}$$

The quantity $F_{\max}$ is the maximum value of $F(y)$ that occurs in a profile and $y_{\max}$ is the value of y at which it occurs. The function $F_{\text{Kleb}}(y)$ is the Klebanoff intermittency factor given by,

$$F_{\text{Kleb}}(y) = \left[1 + 5.5\left(\frac{C_{\text{Kleb}}y}{y_{\max}}\right)^6\right]^{-1} \tag{104}$$

the quantity U_{Dif} is the difference between maximum and minimum total velocity in the profile,

$$U_{\text{Dif}} = \left(\sqrt{u^2+v^2}\right)_{\max} - \left(\sqrt{u^2+v^2}\right)_{\min} \tag{105}$$

The outer formulations (Equations (102) and (103)) can be used in the wakes as well as in attached and separated boundary layers. The product $y_{\max}F_{\max}$

replaces $\delta^* u_e$ in the Clauser formulation and the combination $y_{\max} U_{\mathrm{Dif}}^2 / F_{\max}$ replaces δu_{Dif} in a wake formulation. In effect, the distribution of vorticity is used to determine length scales so that the necessity for finding the outer edge of the boundary layer (or wake) is removed.

The effect of transition to turbulence can be simulated by setting μ_t equal to zero everywhere in a profile for which the maximum tentatively computed value of μ_t from the foregoing relations is less than a specified value, that is,

$$\mu_t = 0 \quad \text{if} \quad (\mu_t)_{\substack{\max \text{ in} \\ \text{profile}}} < C_{\mathrm{Mutm}}\, \mu_\infty \tag{106}$$

The constants appearing in the foregoing relations have been determined by requiring agreement with the Cebici formulation for constant pressure boundary layers at transonic speeds. The values determined are,

$$A_+ = 26$$

$$C_{cp} = 1.6$$

$$C_{\mathrm{Kleb}} = 0.3$$

$$C_{\mathrm{Wk}} = 0.25$$

$$k = 0.4$$

$$K = 0.0168$$

$$P_r = 0.72$$

$$P_{rt} = 0.9$$

$$C_{\mathrm{Mutm}} = 14$$

VITA

Sudheer Nath Nayani was born in Anakapalle, India on June 25, 1951, and attended primary school there. He graduated in 1973 from Osmania University, Hyderabad, with a Bachelor of Engineering degree in Mechanical Engineering. He obtained his Master of Technology degree from Indian Institute of Technology, Madras in 1975 majoring in Mechanical Engineering. After graduation he worked as Senior Technical Assistant till February 1976 at Indian Institute of Technology, Madras in their Thermal Turbomachinery Laboratory. He was employed by Bharat Heavy Electricals Limited, Hyderabad in their Research and Development Division from February 1976 to May 1982, in the capacities of Development and Senior Development Engineer. He entered The University of Tennessee Space Institute, Tullahoma in the Summer quarter of 1982 where he majored in Mechanical Engineering.

He was married on October 23, 1980, to the former Miss Usha Koka. They have a son, Sree Vikram.