Order Number 8910129

Time-marching schemes for solving the Euler equations

Tsai, Ying-Liang Peter, Ph.D.

The Pennsylvania State University, 1988

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

The Pennsylvania State University

The Graduate School

Department of Mechanical Engineering

TIME-MARCHING SCHEMES FOR SOLVING THE EULER EQUATIONS

A Thesis in

Mechanical Engineering

by

Ying-Liang Tsai

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

December 1988

We approve the thesis of Ying-Liang Tsai.

Date of Signature

Charles L. Merkle — October 26 1988

Charles L. Merkle
Professor of Mechanical Engineering
Thesis Advisor
Chair of Committee

Anil K. Kulkarni — October 26, 1988

Anil K. Kulkarni
Associate Professor of Mechanical Engineering

W G Pritchard — 31 Oct 1988

William G. Pritchard
Professor of Mathematics

10/26/88

Savas Yavuzkurt
Associate Professor of Mechanical Engineering

Harold R. Jacobs — Oct 26, 1988

Harold R. Jacobs
Head of the Department of Mechanical Engineering

ABSTRACT

Advanced time-marching algorithms are designed to compute steady-state solutions of the Euler equations. These algorithms include the implicit ADI and the explicit Runge-Kutta multi-stage schemes that utilize central differencing in space. Additional implicit schemes such as the full implicit, diagonal ADI, and various LU schemes are also explored. The computations are made on a body-fitted, generalized coordinate system.

The present work proceeds by first studying the implicit ADI and explicit Runge-Kutta schemes on one- and two-dimensional, incompressible and compressible problems. The converged solutions for both schemes are identical, and their accuracy is confirmed by available analytical solutions. The efficiencies (convergence rates) of these schemes are a major issue of the present work. The results show that the implicit scheme is overwhelmingly superior to the explicit scheme in one dimension, is only slightly more efficient than the explicit scheme in two dimensions, and is less efficient than the explicit scheme in three dimensions.

Along with the flow computations, stability analyses are carefully studied. The stability analysis decomposes the residual errors via the Fourier transformation and determines their amplification. This powerful tool predicts the convergence of a scheme with accuracy. Important parameters such as the CFL number and artificial dissipation coefficients (artificial viscosities) are most appropriately determined by the stability analysis.

The Euler equations are hyperbolic or can be made hyperbolic in time. It is well-known that the system of equations governing the inviscid compressible flows is hyperbolic. The inviscid incompressible system of equations is also made hyperbolic by the concept of pseudocompressibility, or more precisely, adding an artificial

temporal term to the continuity equation. These two systems are then solved in a generalized fashion.

The boundary conditions for these hyperbolic systems are implemented based on the theory of characteristics. The boundary conditions thus formulated are called the MOC (method of characteristics) boundary conditions. In the implicit ADI scheme the MOC boundary conditions are applied implicitly, and in the Runge-Kutta scheme they are applied in every stage. Substitutional boundary-condition procedures known as the explicit boundary procedures are presented as well for comparisons.

The calculations for three-dimensional incompressible Euler equations and their stability analysis are presented in the last chapter. As an example of engineering applications, flows passing a propeller behind a ship hull are calculated.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

a	One-Dimensional Nozzle Area
A,B,C	Flux Jacobian Matrices
ADI	Alternating-Direction Implicit
c	Sonic Speed
C	Cosine Function of Wavenumber
CFL	Courant-Friedrichs-Lewy Number
D	Propeller Diameter
D	Source Jacobian Matrix
e	Total Energy
E,F,G	Flux Vectors
g	Amplification Factor
G	Amplification Matrix
H	Source Vector
i	$\sqrt{-1}$
I	Identity Matrix
J	Coordinate-Transformation Jacobian
K_T	Thrust Coefficient
K_Q	Torque Coefficient
L	Selection Matrix
m	Number of Equations
M	Mach Number
M	Left Eigenvector Matrix
MOC	Method of Characteristics
n	Revolution per Second

N	Number of Nodal Points
p	Pressure
P	Euler-Explicit Space Operator
$\mathbf{Q}$	Dependent Variable Vector
r,θ,z	Cylindrical Coordinates
R_1,R_2	Riemann Invariants
RHS	Right-Hand-Side Residual
s	Entropy
S	Sine Function of Wavenumber
t	Time
u	x-Direction Velocity
U	ξ-Directional Contravariant Velocity
u_r,u_θ,u_z	Cylindrical Velocity Components
U_0	Ship Speed
v	y-Direction Velocity
V	η-Directional Contravariant Velocity
x,y,z	Cartesian Coordinates
β	Pseudocompressibility Coefficient
γ	Ratio of the Specific Heats
δ	Central-Difference Operator
ϵ	Residual-Smoothing Coefficient
$\epsilon_i^{(2)}$	Second-Order Implicit Artificial Viscosity
$\epsilon_e^{(4)}$	Fourth-Order Explicit Artificial Viscosity
λ	Eigenvalue
$\mathbf{\Lambda}$	Diagonal Eigenvalue Matrix
ξ,η,ζ	Generalized Coordinates

ρ	Density
ω	Rotational Speed of Cylindrical Coordinates
$\mathbf{\Omega}$	Rotational Speed of Body
$\omega_\xi,\omega_\eta,\omega_\varsigma$	Wavenumbers
$\mathbf{\Omega}$	Boundary-Condition Vector

SUBSCRIPTS

c	Compressible
i	Incompressible

SUPERSCRIPTS

n	Time-Step Index

ACKNOWLEDGEMENTS

The author would like to express his deepest appreciation to his advisor, Dr. Charles L. Merkle, for his guidance, assistance, and encouragement throughout the course of this work. Appreciation is also extended to Drs. Anil K. Kulkarni, William G. Pritchard, and Savas T. Yavuzkurt for serving on the Doctoral Committee.

This research was performed under Contract No. N00014-87-K-0123, sponsored by the Office of Naval Research (ONR). The financial support of ONR is acknowledged.

The author wishes to express his gratitude to his parents and loving wife, Hsueh-Chin Hsu, for their devotion and patience.

CHAPTER 1
INTRODUCTION

The purpose of this work is to design algorithms and programs that are capable of simulating steady state inviscid fluid motions and to understand the characteristics of a variety of schemes when applied to the Euler equations. These algorithms can be applied to both incompressible and compressible Euler equations in the same manner.

As an example of engineering applications of these algorithms, flows passing a propeller behind a ship hull are calculated. This strongly three-dimensional problem is made complicated by the non-uniform and rotational inflow which is induced by the existence of the hull. Compared to operations in uniform inflows, wake operation causes a dramatic increase of critical inception indices and strong amplitude modulation of cavitation noise spectra [1-2]. One might justify choosing the incompressible Navier-stokes equations as the governing equations for this problem. However, the solutions of the Euler equations are adequate in preliminary design because the pressure is the most important design data, and is nearly constant across a boundary layer in high Reynolds number flows. With the inviscid solution, a boundary layer code can be used to obtain sufficiently accurate details in the viscous regions.

The incompressible Euler equations are frequently encountered in hydrodynamic and low Mach number aerodynamic calculations. They are obtained by assuming the density to be constant, leaving only the pressure and velocity components as the dependent variables. Under the irrotationality assumption, the velocity components can be replaced by a potential function. This simplifies the problem to

that of solving the Laplace equation. However, when the incoming flow is rotational, the potential function does not exist, and the Euler equations must be solved.

For a compressible flow, the signal propagation speed is $\partial p/\partial \rho$ under an isentropic process. This speed becomes infinite when the density is a constant. Hence the wave propagation speed of an incompressible flow is infinite. This makes the incompressible Euler equations parabolic in time. One possible way of solving them is by introducing stream function and vorticity [3]. Unfortunately, this method can be used only on two-dimensional problems. An alternative would be to use the primitive variables, i.e., the velocity components and the pressure, while treating the continuity equation as a constraint to the velocity components [4]. But then the pressure-velocity coupling becomes a significant difficulty [5]. It can be overcome by using the pressure Poisson equation. However, a Poisson solver must be called in each iteration. This is a heavy burden in solving the problem.

Recent introduction of pseudocompressibility to the incompressible system overcomes all the above difficulties. By adding a time derivative of pressure to the continuity equation, the incompressible Euler equations become hyperbolic [6]. The resulting equations contain all the primitive variables in the time derivative. The artificial time derivative of pressure in continuity equation makes the system transiently incorrect, but in the steady state limit, when the time derivatives vanish, the divergence free ($\nabla \cdot \mathbf{V} = 0$) condition is recovered. Since only the steady state solution is concerned here, this method is applicable.

Several advantages of this algorithm can be found. They are:

1. The single vector hyperbolic equation thus formed can be solved by using most of the well developed time-marching schemes for hyperbolic systems [6-11]. It can be shown that the eigenvalues of the flux matrices are all real. This puts the problem into the framework of the mathematical theory of hyperbolic

equations. Stemming from the early work of Courant, Von Neumann, and Lax, this theory is by now highly developed [7,12-15].

2. It provides a straightforward technique for obtaining pressure from the conservation equations.
3. It is inherently stable and robust, because the pressure and velocity components are obtained simultaneously at the new time level.
4. Its stability analysis can be performed, thus making it feasible to predict, understand, and control the convergence rate.
5. It has excellent adaptivity; it can be applied to one, two, or three dimensions, inviscid or viscous, laminar or turbulent problems without major overhaul.

The concept of pseudocompressibility makes the incompressible Euler equations mathematically similar to the compressible Euler equations. Both of them can be written in a conservative form, and both of them are hyperbolic in time. This enables us to apply various time-marching schemes originally designed for the compressible Euler equations. The question remains as to what time-marching schemes should be used.

Time-marching schemes have become an important segment of Computational Fluid Dynamics (CFD) since Moretti and Abbett published the first practical solution for the supersonic blunt-body problem two decades ago [16]. A large number of time-marching schemes were published during this period. Schemes such as Lax-Wendroff scheme [7] and MacCormack scheme [11] gained their popularity in the early years of CFD. As CFD developed, more efficient schemes became available. Nowadays, Beam and Warming's implicit ADI scheme [8-9], which was subsequently called the LBI (Linearized Block Implicit) scheme by Briley and McDonald [17], and more recently, Jameson's explicit Runge-Kutta multi-stage scheme are widely used algorithms. Some time-marching schemes have been used to solve the incom-

pressible system with success since the invention of pseudocompressibility. Chorin [6] solved the hyperbolized incompressible flow equations using the explicit leap-frog/Dufort-Frankel method. His convergence was rather slow, but it suggested that more efficient time-marching schemes could be devised. Later, Steger and Kutler applied the ADI scheme to solve low Mach number vortex wake flows [18]. This was the first application of the implicit method on this system. More recent applications have also been reported by Choi and Merkle [19-20], and Chang and Kwak [21-22] using the same implicit scheme. Merkle and Tsai [23-24] also applied the explicit Runge-Kutta time-marching scheme to this system.

The present study applies both implicit ADI and explicit Runge-Kutta schemes to the incompressible and compressible Euler equations. Both schemes are implemented by first rewriting the governing equations for the generalized body-fitting coordinate system, and then discretizing the transformed equations by finite-difference methods. Although the original designs of these two schemes were for compressible flows, this work shows that they can be used with equal success for incompressible flows. It is noteworthy that successful results are presented by applying a finite-difference method to the Runge-Kutta scheme, even though it was associated with a finite-volume method in Jameson's first paper in 1981, and followed by most researchers ever since.

The research starts with a series of quasi one-dimensional nozzle flow calculations. One-dimensional flows have limited practical use; however, these computations allow us to assure the implementation of these schemes and to verify the prediction of stability analyses in a simplified problem. They also permit studying the implicit scheme apart from the transient error term introduced by approximate-factorization for multi-dimensional problems. Approximate-factorization error in multiple dimensions appears not only in the interior region, but also on the bound-

aries when MOC (method of characteristics) boundary conditions are employed. In contrast, the application of MOC boundary conditions on one-dimensional problems is especially accurate and justified. Thus, the performance of the MOC boundary conditions can be observed with clarity.

The one-dimensional nozzle problem is a good example of numerical calculations with the presence of source terms. Source terms often cause slow convergence or even instability for a numerical calculation. The governing equations of the one-dimensional nozzle problem contain a momentum source term induced by the nozzle wall pressure. The effect of this particular source term is investigated. As the work proceeds to three dimensions, other source terms may be encountered. For example, if the fluid motion is described in cylindrical coordinates, the curvature of the axes results in momentum source terms. Other source terms such as centrifugal force and Coriolis force can be caused by the rotation of the coordinate system.

Both schemes are then extended to two-dimensional problems. Airfoil, cylinder, and nozzle flows are chosen as sample calculations. The main purposes of this fraction of work are to make comparisons between the efficiencies of the implicit and explicit schemes, and to decide appropriate options for the three-dimensional problems. In one-dimensional problems, the implicit scheme is overwhelmingly superior to the explicit one. The CFL condition imposes no restriction on the implicit scheme, while the maximum CFL number usable in the explicit scheme is quite limited. The only restriction of the implicit scheme comes from the source term. However, this restriction is far less serious than the CFL condition suffered by the explicit scheme. Furthermore, this source term restricts the maximum CFL number in the compressible calculations only, but not in the incompressible ones. In two-dimensional calculations, the implicit scheme loses its dominant advantage. In a two-dimensional implicit scheme, the use of approximate-factorization is usually

unavoidable. It introduces another restriction on the implicit scheme. Although the two-dimensional implicit scheme is still unconditionally stable with approximate factorization, its optimal CFL number degrades to the same level as that of an explicit scheme. Pulliam [25] has compared the convergence speeds of the ADI (the implicit scheme with approximate-factorization) and the Runge-Kutta explicit schemes for compressible flows. He concludes that they are nearly identical in two dimensions [25]. The incompressible results are similar except that the smaller block-matrix size in the incompressible case translates into a preference for the implicit scheme.

Approximate-factorization enforces an even more restrictive stability limit on a three-dimensional problem. Unlike the two-dimensional ADI, the three-dimensional ADI scheme is conditionally stable, with a very restrictive CFL limit. This limit can be alleviated by using numerical dissipation. Compared to the implicit scheme, the explicit scheme retains its stability on a nearly unchanged level from one to three dimensions. The explicit scheme therefore becomes more favorable in three dimensions.

CHAPTER 2
MATHEMATICAL BASIS

In this chapter, the mathematical background for numerically solving a hyperbolic system is briefly reviewed. As mentioned earlier, the Euler equations are hyperbolic or can be made hyperbolic in time. The transient solutions of a hyperbolic system propagate at finite speeds, which are defined by the eigenvalues of the hyperbolic system, and their loca are known as characteristic lines. In the numerical calculations that follow, the boundary conditions are formulated based on the understanding of the characteristics. The mathematical basis for the stability analysis is also discussed in this chapter.

2.1 Hyperbolic Systems

A general, first-order, source-free quasi-linear PDE system takes the following form [27],

$$\mathbf{\Gamma}\frac{\partial \mathbf{Q}}{\partial t} + \sum_{i=1}^{M} \mathbf{A}_i \frac{\partial \mathbf{Q}}{\partial x_i} = \mathbf{0} \tag{2.1}$$

where

$$\mathbf{Q} = \begin{pmatrix} q_1 \\ q_2 \\ \vdots \\ q_N \end{pmatrix} = \mathbf{Q}(x_1, x_2, \ldots, x_M, t)$$

and $\mathbf{A}$'s and $\mathbf{\Gamma}$ are $N \times N$ matrices which are functions of M space variables and time $(x_1, x_2, \ldots, x_M, t)$. A system of higher order can always be reduced to this form by treating the first order derivatives as new dependent variables successively. For simplicity, only one-dimensional systems ($M = 1$) given as

$$\mathbf{\Gamma}\frac{\partial \mathbf{Q}}{\partial t} + \mathbf{A}_1 \frac{\partial \mathbf{Q}}{\partial x} = \mathbf{0} \tag{2.2}$$

will be discussed in this section.

Equation (2.2) is premultiplied by inverse of $\mathbf{\Gamma}$ to get

$$\frac{\partial \mathbf{Q}}{\partial t} + \mathbf{A}\frac{\partial \mathbf{Q}}{\partial x} = \mathbf{0} \tag{2.3}$$

where

$$\mathbf{A} = \mathbf{\Gamma}^{-1}\mathbf{A}_1$$

The eigenvalues of $\mathbf{A}$ determine the type of Eq. (2.3). If all the eigenvalues are real, this system is hyperbolic in time. If one or more of the eigenvalues are complex, then it is either mixed parabolic-hyperbolic or purely parabolic in time.

Assume that there exists a similarity transformation such that

$$\mathbf{M}^{-1}\mathbf{A}\mathbf{M} = \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_N \end{pmatrix} \equiv \mathbf{\Lambda} \tag{2.4}$$

where the λ's are eigenvalues of $\mathbf{A}$, and $\mathbf{M}$ is an $N \times N$ transformation matrix. Premultiplying Eq. (2.3) by $\mathbf{M}^{-1}$ gives

$$\mathbf{M}^{-1}\frac{\partial \mathbf{Q}}{\partial t} + \mathbf{M}^{-1}\mathbf{A}\mathbf{M}\mathbf{M}^{-1}\frac{\partial \mathbf{Q}}{\partial x} = \mathbf{0} \tag{2.5}$$

By defining a new unknown vector, $\hat{\mathbf{Q}}$, through the differential equation,

$$d\hat{\mathbf{Q}} = \mathbf{M}^{-1}d\mathbf{Q} \tag{2.6}$$

Eq. (2.5) can be written as

$$\frac{\partial \hat{\mathbf{Q}}}{\partial t} + \mathbf{\Lambda}\frac{\partial \hat{\mathbf{Q}}}{\partial x} = \mathbf{0} \tag{2.7}$$

The vector $\hat{\mathbf{Q}}$ is the characteristic function and does not assume any explicit form. Equation (2.7) is a system containing several scalar equations. Thus we have actually decomposed Eq. (2.3) into N first order scalar hyperbolic equations.

In order to verify the existence of $\mathbf{M}$, consider the equation for eigenvectors of $\mathbf{A}$,

$$\mathbf{AV} = \mathbf{V\Lambda} \tag{2.8}$$

where the $\mathbf{V}$ is composed of the column eigenvectors, $\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_N$.

$$\mathbf{V} = (\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_N) \tag{2.9}$$

Premultiplying Eq. (2.8) by $\mathbf{V}^{-1}$ gives

$$\mathbf{V}^{-1}\mathbf{AV} = \mathbf{V}^{-1}\mathbf{V\Lambda} = \mathbf{\Lambda} \tag{2.10}$$

By comparing Eq. (2.4) and Eq. (2.10), one finds that $\mathbf{M}$ is simply $\mathbf{V}$.

The purpose of using $\mathbf{M}$ is to decouple the system so that we can select the proper scalar equations of Eq. (2.7) to be replaced by boundary conditions. Consider the left boundary of a one-dimensional domain. Assume that locally the first K eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_K$ are negative and the rest are positive. The first K equations of Eq. (2.7) describe the propagation of information from the interior onto the boundary. These K equations describe the propagation of information from the domain, and therefore should be retained. These equations are represented by proper finite-differencing while the rest of the equations are discarded and replaced by suitable boundary conditions. This can be done by premultiplying Eq. (2.7) by a selection matrix, $\mathbf{L}$.

$$\mathbf{L}\left(\frac{\partial \hat{\mathbf{Q}}}{\partial t} + \mathbf{\Lambda}\frac{\partial \hat{\mathbf{Q}}}{\partial x} = \mathbf{0}\right) \tag{2.11}$$

where $\mathbf{L}$ is a diagonal matrix with the first K diagonal elements as 1 and the remaining elements as 0. In the numerical calculations, $\mathbf{Q}$ rather than $\hat{\mathbf{Q}}$ is iterated. Therefore, $\mathbf{Q}$ must be recovered from Eq. (2.11) by

$$\mathbf{LM}^{-1}\left(\frac{\partial \mathbf{Q}}{\partial t} + \mathbf{A}\frac{\partial \mathbf{Q}}{\partial x} = \mathbf{0}\right) \tag{2.12}$$

The $N - K$ dummy equations are then replaced by the specified boundary conditions.

Treating the boundary conditions via the method of characteristics (MOC) as derived above was proposed by Rai-Chaussee [28] and Chakravarthy [29]. This method is utilized in the present work because of its stability and accuracy. Some other treatments either reflect disturbances or require additional assumptions. The importance of the boundary conditions can never be overemphasized. Improper treatment of boundary conditions can cause incorrect results, slow convergence, or even instability. Hence the finite-difference schemes in this work mainly use the MOC boundary conditions.

Equation (2.3) is non-conservative. In most finite-difference applications, the conservative form is preferred. The governing equations of a one-dimensional problem may be given in the following conservative form,

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{E}(\mathbf{Q})}{\partial x} = \mathbf{0} \tag{2.13}$$

where $\mathbf{E}$, the flux vector, is a function of the primitive variables, i.e., the elements in $\mathbf{Q}$. By the chain rule, one gets

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{E}}{\partial \mathbf{Q}} \frac{\partial \mathbf{Q}}{\partial x} = \mathbf{0} \tag{2.14}$$

The flux Jacobian, $\partial \mathbf{E}/\partial \mathbf{Q}$, in Eq. (2.14) thus corresponds to the matrix $\mathbf{A}$ in Eq. (2.3).

2.2 Stability Analysis

For stability analyses we use Fourier analysis. Fourier analysis has been a powerful tool for solving partial differential equations since it was invented almost two centuries ago. Nowadays, it is also utilized in analyzing the stability of a

numerical scheme. By definition, the Fourier transformation of $u(x,t)$ for a fixed t is [30]

$$\bar{u}(\omega,t) = \int_{-\infty}^{\infty} u(x,t)e^{-i\omega x}dx \tag{2.15}$$

in which $u(x,t)$ must be square summable

$$\int_{-\infty}^{\infty} u^2(x,t)dx < \infty \tag{2.16}$$

The inverse transformation is

$$u(x,t) = \int_{-\infty}^{\infty} \bar{u}(\omega,t)e^{i\omega x}\frac{d\omega}{2\pi} \tag{2.17}$$

For functions defined at discrete points, the discrete Fourier transformation is defined analogous to the Fourier transformation:

$$\bar{u}^n(\omega) = \Delta x \sum_{-\infty}^{\infty} u_i^n e^{-i\omega x_i} \tag{2.18}$$

The inverse transformation is defined as

$$u_i^n = \frac{\Delta\omega}{2\pi} \sum_{-\pi/\Delta x}^{\pi/\Delta x} \bar{u}^n(\omega)e^{i\omega x_i} \tag{2.19}$$

where $\bar{u}^n(\omega)$ represents the amplitude of the wave mode with wavenumber ω at time level n.

The stability analysis is performed by calculating the amplification factor, $g(\omega)$, of all the wave modes. The amplification factor is defined by

$$g(\omega) = \frac{\bar{u}^{n+1}(\omega)}{\bar{u}^n(\omega)} \tag{2.20}$$

and is dependent on the discretization scheme used. If $|g|$ is less than unity for all wavenumbers, then all residual wave modes are dissipative and the scheme is stable. If a scheme is stable and consistent, it is convergent [15,31]. On the other hand, the scheme is unstable if $|g|$ is greater than one for even one single wave mode.

Consider the advective model problem,

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0 \tag{2.21}$$

on an infinite domain. This equation is a representative scalar equation of Eq. (2.7), with the wave propagation speed, c, equivalent to the eigenvalue. Here stability analyses of three schemes applied to Eq. (2.21) are given, namely, Euler explicit, leapfrog, and Euler implicit. Details of stability analyses for many other schemes can be found in Vichnevetsky and Bowles' book [30].

The discretized form of Eq. (2.21) for the Euler explicit scheme is,

$$u_i^{n+1} - u_i^n = -\frac{c\Delta t}{2\Delta x}(u_{i+1}^n - u_{i-1}^n) \tag{2.22}$$

Performing a Fourier transformation, we get

$$\bar{u}^{n+1} - \bar{u}^n = -\frac{CFL}{2}(e^{i\omega\Delta x}\bar{u}^n - e^{-i\omega\Delta x}\bar{u}^n) \tag{2.23}$$

where the Courant number, CFL, is defined by

$$CFL = \frac{c\Delta t}{\Delta x} \tag{2.24}$$

Note that in principle $\omega\Delta x$ in Eq. (2.23) ranges from $-\pi$ to π. However, the period of the amplification factor, g, is usually π. Therefore, it is sufficient to confine $\omega\Delta x$ to 0 to π. Divide Eq. (2.23) by $\bar{u}^n$ to get

$$g = 1 - \frac{CFL}{2}(e^{i\omega\Delta x} - e^{-i\omega\Delta x}) = 1 - iCFL\sin(\omega\Delta x) \tag{2.25}$$

The magnitude of g is thus

$$|g| = \sqrt{1 + CFL^2\sin^2(\omega\Delta x)} \tag{2.26}$$

Equation (2.26) shows that $|g|$ is always greater or equal to one. The Euler explicit scheme is therefore unconditionally unstable for the hyperbolic equation (2.21).

This is true even if the fourth-order artificial dissipation is added to the right hand side of Eq. (2.22). Although the fourth-order artificial dissipation can effectively damp out high-wavenumber modes, it has very little effect on low-wavenumber modes. Stability analysis of Eq. (2.22) with the fourth-order artificial dissipation on the right hand side can reveal this.

Next consider the leapfrog scheme:

$$u_i^{n+1} - u_i^{n-1} = -\frac{c\Delta t}{\Delta x}(u_{i+1}^n - u_{i-1}^n) \tag{2.27}$$

The stability analysis gives

$$g = iCFL\sin(\omega\Delta x) \pm \sqrt{-CFL^2\sin^2(\omega\Delta x) + 1} \tag{2.28}$$

This scheme is stable provided

$$\left|\frac{c\Delta t}{\Delta x}\right| \leq 1 \tag{2.29}$$

This is the famous Courant-Friedrichs-Lewy, or CFL, condition [32].

The finite-difference equation for an implicit scheme can be written as

$$u_i^{n+1} - u_i^n = -\frac{c\Delta t}{2\Delta x}(u_{i+1}^{n+1} - u_{i-1}^{n+1}) \tag{2.30}$$

The amplification factor is given by

$$(1 + iCFL\sin(\omega\Delta x))g = 1 \tag{2.31}$$

and its magnitude is,

$$|g| = \frac{1}{\sqrt{1 + CFL^2\sin^2(\omega\Delta x)}} \tag{2.32}$$

which indicates that the implicit scheme is unconditionally stable.

Note that the amplification factor of the highest-wavenumber mode, $|g(\pi)|$, is one in all three schemes. This implies that this wave mode stays forever. In order

to obtain a steady state solution, additional dissipative terms are added to the right hand side of Eq. (2.21). Although spurious, the dissipative terms provide a fast route to reach the steady state. The dissipative terms do affect the steady state solution, but the error introduced by them can be minimized if chosen carefully.

Fourth-order artificial dissipation is frequently used for hyperbolic systems [30]:

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = -\frac{\epsilon_e^{(4)}}{8}\frac{\Delta x^4}{\Delta t}\frac{\partial^4 u}{\partial x^4} \tag{2.33}$$

where $\epsilon_e^{(4)}$ is the artificial dissipation coefficient. The subscript 'e' represents 'explicit' since it is multiplied to the terms at the known time-level. The fourth-order explicit dissipation can effectively smear out high-wavenumber disturbances, and keeps the accuracy of Eq. (2.21) up to $O(\Delta x^2)$ because it itself is of $O(\Delta x^4/\Delta t)$. The second-order accuracy is deteriorated only when Δt varies as Δx^2, a condition that never occurs in this work. Actually, when the 'constant CFL' condition is imposed, Δt is proportional to Δx. Thus when the number of grid points is doubled under the 'constant CFL' condition, the error introduced by the fourth-order dissipation becomes 1/8 of the original. Meanwhile, as shown by the amplification factor, the damping power of the dissipative term in Eq. (2.33) remains unchanged under grid refinement if the CFL number and $\epsilon_e^{(4)}$ are fixed.

With the added fourth-order viscosity, the amplification factor of Eq. (2.33) for the implicit scheme is

$$\left(1 + iCFL\sin\left(\omega\Delta x\right)\right) g = 1 - \frac{\epsilon_e^{(4)}}{2}\left(\cos\left(\omega\Delta x\right) - 1\right)^2 \tag{2.34}$$

and its magnitude is given as

$$|g| = \frac{1 - \frac{\epsilon_e^{(4)}}{2}\left(\cos(\omega\Delta t) - 1\right)^2}{\sqrt{1 + CFL^2\sin^2(\omega\Delta x)}} \tag{2.35}$$

At the highest wavenumber where $\omega\Delta x = \pi$, $|g| = 1 - 2\epsilon_e^{(4)}$. If $\epsilon_e^{(4)}$ is chosen to be 0.5, the amplification factor of the highest-wavenumber wave goes to zero,

suggesting that the error at this wavenumber will vanish in one iteration. In fact, $\epsilon_e^{(4)} = 0.5$ is used throughout the one-dimensional computation. The following example demonstrates the accuracy of stability prediction.

We now use the implicit scheme to solve Eq. (2.33) for $0 \leq x \leq 1$ with $c = 1$ and $\epsilon_e^{(4)} = 0.5$. The boundary condition is $u = 1$ at $x = 0$. Wave modes exist only at $\omega\Delta x = \pi/k$, where $k = 1, 2, \cdots, N$, and N is the number of grid points. Since the magnitude of the amplification factor monotonically decreases within the range $0 \leq \omega\Delta x \leq \pi$, the wave mode $\omega\Delta x = \pi/N$ has the slowest dissipation rate and should be controlling the convergence speed. Hence we can predict the convergence rate by calculating $g(\pi/N)$. Figure 1 shows such comparisons. The comparisons in Fig. 1 are made by using $N = 10$ and $CFL = 10$, 100, and 1000. The solid lines are predicted by the stability analysis through calculating $g(\pi/N)$, and the circles are numerical calculation results. They agree very well.

The stability analysis for a hyperbolic system of equations is performed in a fashion analogous to the scalar analysis. The amplification matrix, G, is defined by

$$\mathbf{Q}^{n+1} = \mathbf{G}\mathbf{Q}^n \tag{2.36}$$

The equation for the amplification matrix can be obtained for a specific scheme. One generally gets

$$\mathbf{L}_1\mathbf{G} = \mathbf{L}_2 \tag{2.37}$$

$$\mathbf{G} = \mathbf{L}_1^{-1}\mathbf{L}_2 \tag{2.38}$$

It is the eigenvalues of G that determine the stability of the scheme. When the absolute values of all eigenvalues are less than or equal to unity, the scheme is stable.

Two comments are made at this point. First, the stability analysis above is for linear systems. Stability analysis for non-linear systems can be made in the same

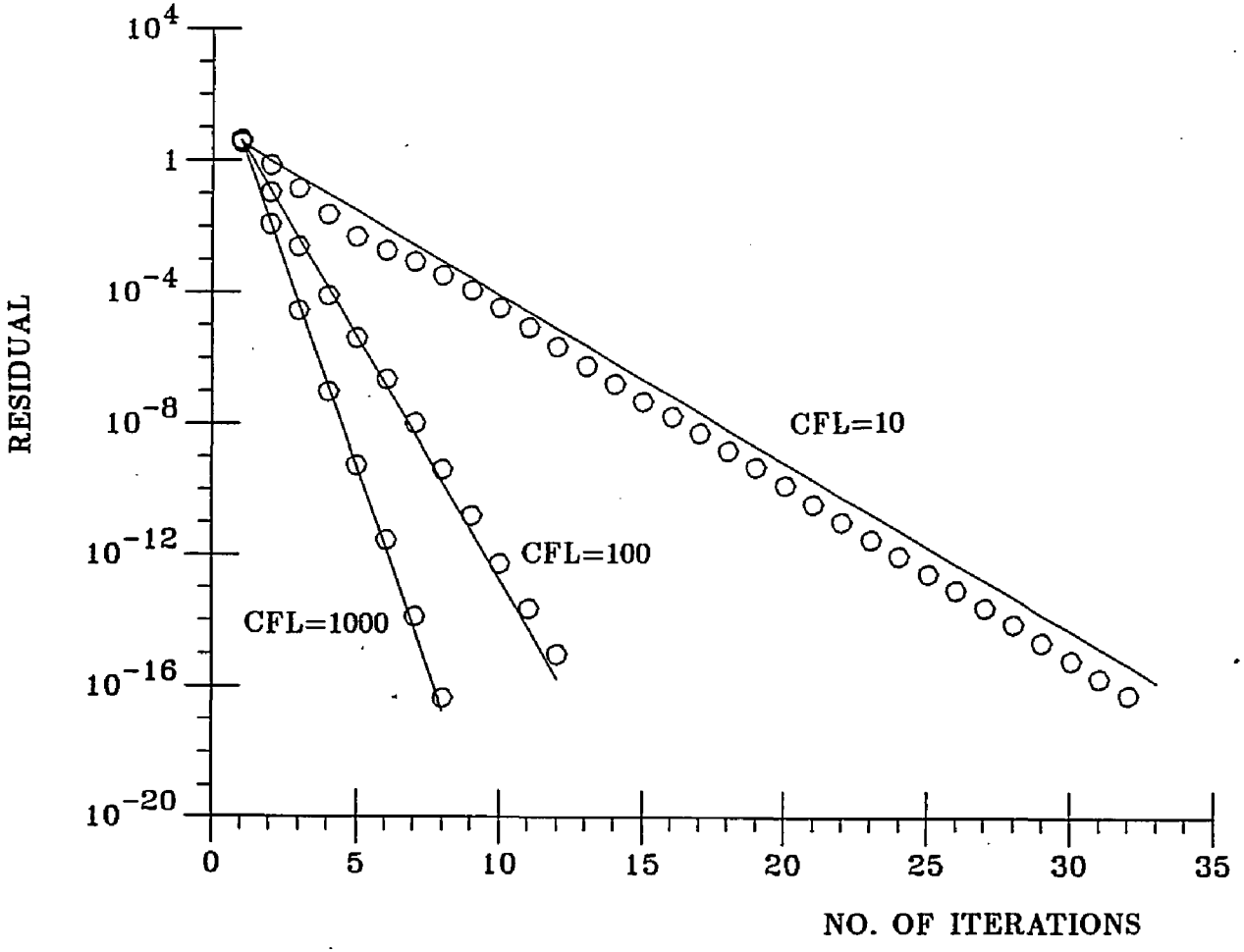

Figure 1: Comparison of Convergence Rates between Prediction by Stability Analysis and Actual Calculations.

way by assuming constant coefficient matrices. However, when the amplitudes of the waves are large compared to the steady state solution, the waves are essentially non-linear. When this happens, the linear stability results may not be accurate because the linear-wave assumption is invalid. This often occurs at the early stage of computation when the error introduced by the initial guess is large, but the approximation becomes better and better as the solution converges. Second, the stability analysis described above is a local analysis. It assumes an infinite domain which excludes the effect of boundary conditions. To make a complete stability analysis that includes the effects of the boundary conditions, a matrix method must be invoked [31].

CHAPTER 3

ONE-DIMENSIONAL PROBLEMS

3.1 Governing Equations

It is well known that the one-dimensional unsteady Euler equations of an compressible flow through a nozzle are hyperbolic in time. On the other hand, the incompressible Euler equations are parabolic in time. However, the incompressible Euler equations can also be made hyperbolic by adding an artificial time-derivative to the continuity equation. As a consequence, the two systems of equations can be written in a generalized vector form:

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{E}(\mathbf{Q})}{\partial x} = \mathbf{H} \tag{3.1}$$

where $\mathbf{Q}$ is the vector of dependent variables, $\mathbf{E}$ is the flux vector, and $\mathbf{H}$ is the source vector. This vector equation is the same as Eq. (2.13) except for the addition of the source vector.

For compressible flows, the vectors, $\mathbf{Q}$, $\mathbf{E}$, and $\mathbf{H}$ are given as

$$\mathbf{Q}_c = \begin{pmatrix} \rho a \\ \rho u a \\ e a \end{pmatrix}, \quad \mathbf{E}_c = \begin{pmatrix} \rho u a \\ (\rho u^2 + p) a \\ (e + p) u a \end{pmatrix}, \quad \mathbf{H}_c = \begin{pmatrix} 0 \\ p \frac{da}{dx} \\ 0 \end{pmatrix} \tag{3.2}$$

where the subscript 'c' represents 'compressible', and ρ, u, e, p, and a stand for density, velocity, total energy, pressure, and area respectively. The source term in the momentum equation comes from the pressure force on the nozzle wall. This term can cause numerical instability as will be shown later in this chapter. The total energy in Eq. (3.2) is defined by

$$e = \frac{p}{\gamma - 1} + \frac{1}{2} \rho u^2$$

This compressible system must be accompanied by an equation of state to equalize the number of equations and the number of variables. Here the perfect gas relation is used.

For incompressible flows, the vectors are designated by the subscript 'i'. They are

$$\mathbf{Q}_i = \begin{pmatrix} pa/\beta \\ ua \end{pmatrix}, \qquad \mathbf{E}_i = \begin{pmatrix} ua \\ (u^2+p)a \end{pmatrix}, \qquad \mathbf{H}_i = \begin{pmatrix} 0 \\ p\frac{da}{dx} \end{pmatrix} \tag{3.3}$$

where p is the pressure divided by the density, u is the velocity, and β is a constant. The number of equations is reduced by one since the energy equation is uncoupled from the system in the incompressible case. Thus only two dependent variables are involved in Eq. (3.3), namely, p and u.

The value of β is important to the convergence rate. In the early pseudocompressibility works [6,18], β was chosen to be large in order to maintain temporal accuracy. However, if only the steady state solution is concerned, β should be chosen such that a fast convergence rate can be obtained. As will be seen in the following chapter, the disturbances of the two-dimensional incompressible hyperbolic system propagate at speeds u, $u+\sqrt{u^2+\beta}$, and $u-\sqrt{u^2+\beta}$. If β is either too large or too small compared to u^2, the system becomes stiff, and approximate-factorization is ill-conditioned. In one-dimensional problems, the disturbances propagate at speeds $u+\sqrt{u^2+\beta}$, and $u-\sqrt{u^2+\beta}$. Since the left-hand-side matrix is directly inverted without approximate-factorization the stiffness is not a problem. Yet if β is too small compared to u^2, the propagation of the two waves is very non-symmetric and the convergence could be slow. On the other hand, a large value of β makes two eigenvalues so large that the time-step size is greatly reduced to satisfy the CFL condition. Therefore, β should be of the order of u^2 to obtain fast convergence.

3.2 Implicit Scheme

The Euler equations have been solved extensively using time-marching schemes.

Schemes such as Lax-Wendroff [7] and MacCormack [11] both have been used for decades with success. More recently, Beam and Warming's successful application of implicit scheme on Euler equations [8-9] has received widespread attention. The implicit scheme has been proven to converge much faster than Lax-Wendroff and MacCormack schemes. The complete numerical domain of dependence for implicit scheme removes the CFL restriction such that a much larger time step can be applied. On the other hand, time-step sizes for Lax-Wendroff and MacCormack schemes must satisfy a restrictive CFL condition. Although an implicit scheme needs more computational effort for each time step, the overall computation time is less.

3.2.1 Time-Stepping Procedure

The finite-difference form of Eq. (3.1) for the implicit scheme is obtained by taking a central difference for the space operator at time level $(n+1)$ and a backward difference for the time operator. The result is

$$\mathbf{Q}_i^{n+1} - \mathbf{Q}_i^n + \frac{\Delta t}{2\Delta x}(\mathbf{E}_{i+1}^{n+1} - \mathbf{E}_{i-1}^{n+1}) = \Delta t \mathbf{H}_i^{n+1} \tag{3.4}$$

By applying a linearization procedure,

$$\mathbf{E}^{n+1} \simeq \mathbf{E}^n + \mathbf{A}^n \Delta \mathbf{Q}$$

$$\mathbf{H}^{n+1} \simeq \mathbf{H}^n + \mathbf{D}^n \Delta \mathbf{Q}$$

Eq. (3.4) can be written in a delta-form as

$$\begin{aligned}(\mathbf{I} - \Delta t \mathbf{D}_i^n)\Delta \mathbf{Q}_i + \frac{\Delta t}{2\Delta x}(\mathbf{A}_{i+1}^n \Delta \mathbf{Q}_{i+1} - \mathbf{A}_{i-1}^n \Delta \mathbf{Q}_{i-1}) \\ = \Delta t \mathbf{H}_i^n - \frac{\Delta t}{2\Delta x}(\mathbf{E}_{i+1}^n - \mathbf{E}_{i-1}^n)\end{aligned} \tag{3.5}$$

where $\mathbf{I}$ is the identity matrix, $\mathbf{A}$ and $\mathbf{D}$ are the Jacobians defined by

$$\mathbf{A} \equiv \frac{\partial \mathbf{E}}{\partial \mathbf{Q}}, \qquad \mathbf{D} \equiv \frac{\partial \mathbf{H}}{\partial \mathbf{Q}}$$

and $\Delta\mathbf{Q}$ is the difference of $\mathbf{Q}$ between two time levels.

$$\Delta\mathbf{Q} \equiv \mathbf{Q}^{n+1} - \mathbf{Q}^n$$

Written in a compact form, Eq. (3.5) becomes

$$(\mathbf{I} - \Delta t\mathbf{D} + \Delta t\delta_x\mathbf{A}\cdot)\Delta\mathbf{Q} = RHS \tag{3.6}$$

where δ_x is the central-difference operator. For example,

$$\delta_x(\mathbf{A}\Delta\mathbf{Q}) = \frac{1}{2\Delta x}\left[(\mathbf{A}\Delta\mathbf{Q})_{i+1} - (\mathbf{A}\Delta\mathbf{Q})_{i-1}\right]$$

The right-hand-side of Eq. (3.6) (denoted as RHS) represents the steady state solution. As the numerical procedure converges, $\Delta\mathbf{Q}$ vanishes, and the steady state solution is obtained.

The Jacobians $\mathbf{A}$ and $\mathbf{D}$ are given as

$$\mathbf{A}_c = \begin{pmatrix} 0 & 1 & 0 \\ \frac{\gamma-3}{2}u^2 & (3-\gamma)u & \gamma-1 \\ (\gamma-1)u^3 - \frac{\gamma e u}{\rho} & \frac{\gamma e}{\rho} - \frac{3\gamma-3}{2}u^2 & \gamma u \end{pmatrix},$$

$$\mathbf{D}_c = \begin{pmatrix} 0 & 0 & 0 \\ \frac{\gamma-1}{2}\frac{1}{a}\frac{da}{dx}u^2 & -(\gamma-1)\frac{1}{a}\frac{da}{dx}u & (\gamma-1)\frac{1}{a}\frac{da}{dx} \\ 0 & 0 & 0 \end{pmatrix} \tag{3.7}$$

for compressible flow, and

$$\mathbf{A}_i = \begin{pmatrix} 0 & 1 \\ \beta & 2u \end{pmatrix}, \qquad \mathbf{D}_i = \begin{pmatrix} 0 & 0 \\ \beta\frac{1}{a}\frac{da}{dx} & 0 \end{pmatrix}$$

for incompressible flow.

This method is first-order accurate in time, assuming that Δt is a constant throughout the domain for each time level. Schemes with higher order time-accuracy such as Crank-Nicolson scheme [31] are available, but they generally converge more slowly. If only the steady state solution is desired, the major concern is fast convergence instead of time-accuracy. In fact, one can define a generalized implicit scheme as

$$(\mathbf{I} - \theta\Delta t\mathbf{D} + \theta\Delta t\delta_x\mathbf{A}\cdot)\Delta\mathbf{Q} = RHS \qquad 3.8$$

such that $\theta = 0, 1/2, 1$ represents Euler explicit, Crank-Nicolson, and Euler implicit respectively. If the source term is absent, the scheme is unconditionally unstable when θ is less than $1/2$, and unconditionally stable when θ is greater or equal to $1/2$. However, strongest temporal dissipation occurs when θ is one. As Eq. (3.8) shows, all values of θ lead to identical steady-state solution upon convergence ($RHS = 0$).

The selection of time-step size is essential to fast convergence. If one insists on using constant Δt, it is logical to search through the domain for the largest eigenvalue to determine Δt. Nevertheless, this is unnecessarily conservative; the result is slow convergence. Since timewise accuracy is not an issue, Δt could vary from point to point. The time-step size, Δt, can be determined based on a local maximum eigenvalue. In order to keep every grid point converging at approximately the same rate, the local CFL number is kept constant. The time-step size is then determined by

$$\Delta t = \frac{CFL\Delta x}{|\lambda_{max}|} \tag{3.9}$$

where CFL is a constant number and $|\lambda_{max}|$ is the local maximum eigenvalue. This 'constant CFL' strategy has been proven to be stable and effective.

3.2.2 Implicit Boundary Conditions

The boundary condition procedure used herein was originally proposed by Chakravarthy [29] and Rai-Chaussee [28]. As addressed in Section 2.1, a hyperbolic system such as Eq. (3.1) can be decoupled into a number of first-order scalar equations if premultiplied by the inverse of eigenvalue matrix $\mathbf{M^{-1}}$. The eigenvalue-sign of each scalar equation determines the direction of the characteristic line. If a characteristic line is outgoing to the boundary, the boundary procedure must be able to extract that information from the interior of the domain. If a characteristic line is incoming onto the boundary, a boundary condition must be specified.

To apply an implicit MOC boundary condition in finite-difference form, Eq.

(3.6) is multiplied by the matrix $\mathbf{LM^{-1}}$ as

$$\mathbf{LM^{-1}}\left[(\mathbf{I} - \Delta t\mathbf{D} + \Delta t\delta_x\mathbf{A}\cdot)\,\Delta\mathbf{Q} = RHS\right] \tag{3.10}$$

where the selection matrix, **L**, is a diagonal matrix with 0's and 1's in the diagonal line such that the outgoing characteristics are multiplied by '1' and 'selected'. The spatial-difference operators on the left- and right-hand-sides must be represented by one-sided differencing on the boundaries. The rows in Eq. (3.10) multiplied by diagonal zeroes of the selection matrix represent incoming characteristics, and are to be replaced by boundary conditions.

If expressed in vector form, the specified boundary conditions can be written as

$$\mathbf{\Omega}(\mathbf{Q}) = 0 \tag{3.11}$$

where $\mathbf{\Omega}$ contains the same number of non-zero elements as the number of incoming characteristics, and these elements are located on the same rows as the zeroes of the matrix **L**. It is intended that $\mathbf{\Omega}$ be fixed on the boundaries. By truncating the Taylor series to the first order, one gets

$$\mathbf{\Omega}^{n+1} - \mathbf{\Omega}^n = \frac{\partial\mathbf{\Omega}}{\partial\mathbf{Q}}\Delta\mathbf{Q} = 0 \tag{3.12}$$

The finite-difference equation for a boundary point is thus obtained by summing Eq. (3.10) and Eq. (3.12).

As an example, consider one-dimensional compressible nozzle flows. The flux Jacobian **A** is given in Eq. (3.7). It has three real distinct eigenvalues,

$$\lambda_{1,2,3} = u, u + c, u - c$$

where c is the local sonic speed. If the nozzle inlet flow is subsonic, λ_1 and λ_2 are positive, while λ_3 is negative. The selection matrix is then

$$\mathbf{L} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

for the inlet boundary. The two incoming characteristics are to be replaced by two boundary conditions. Usually, experimentally realizable conditions of total pressure and total temperature are used. If the inlet flow is supersonic, no outgoing characteristic line exists. The selection matrix, $\mathbf{L}$, is thus a 3×3 null matrix. Total pressure, total temperature, and Mach number can be specified at the upstream boundary.

The δ_x operator in the left-hand-side of Eq. (3.10) must be replaced by one-sided differencing since all grid points are located on one side of the boundary. Three-point differencing is usually chosen to retain second-order accuracy in space. This destroys the block tridiagonal structure of the left-hand-side matrix. Nevertheless, the desirable block tridiagonal property can be restored by simple algebraic manipulation.

The inverse of eigenmatrix, $\mathbf{M}^{-1}$, is given as

$$\mathbf{M}_c^{-1} = \begin{pmatrix} -\frac{\gamma-1}{2}\frac{u^2}{c^2} + 1 & (\gamma-1)\frac{u}{c^2} & -(\gamma-1)\frac{1}{c^2} \\ \frac{\gamma-1}{4}\frac{u^2}{c^2} - \frac{1}{2}\frac{u}{c} & -\frac{\gamma-1}{2}\frac{u}{c^2} + \frac{1}{2}\frac{1}{c} & \frac{\gamma-1}{2}\frac{1}{c^2} \\ -\frac{\gamma-1}{4}\frac{u^2}{c^2} + \frac{1}{2}\frac{u}{c} & -\frac{\gamma-1}{2}\frac{u}{c^2} - \frac{1}{2}\frac{1}{c} & \frac{\gamma-1}{2}\frac{1}{c^2} \end{pmatrix}$$

for one-dimensional compressible flows and

$$\mathbf{M}_i^{-1} = \begin{pmatrix} -\frac{1}{2}(u - \sqrt{u^2+\beta}) & \frac{1}{2} \\ \frac{1}{2}(u + \sqrt{u^2+\beta}) & -\frac{1}{2} \end{pmatrix} \tag{3.13}$$

for one-dimensional incompressible flows.

3.2.3 Stability Analysis

The stability analysis for a scalar hyperbolic equation, Eq. (2.33), with fourth-order artificial viscosity added has been discussed in Section 2.2. The magnitude of the amplification factor was shown as a function of $\omega\Delta x$, CFL number, and $\epsilon_e^{(4)}$ (see Eq. (2.35)). Here we show the effect of CFL number on $|g|$ in Fig. 2. One observes that a large time step can reduce the amplification factor and thus enhance

the convergence rate. However, the highest-wavenumber is neutrally stable for any CFL number when no viscosity is added. The effect of adding fourth-order viscosity is illustrated in Fig. 3. Figure 3 shows the amplification-factor curves with different values of viscosity for a CFL number of 10. These curves indicate that fourth-order viscosity has stronger effect at high-wavenumbers, and that an optimal value for $\epsilon_e^{(4)}$ is 0.5. These curves in Fig. 2 and Fig. 3 for a scalar equation can give a rough estimation of the stability criteria for the vector systems being discussed.

Based on the scalar stability analysis, fourth-order viscosity is added to Eq. (3.1) in an attempt to dissipate high-wavenumber modes as

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{E}(\mathbf{Q})}{\partial x} - \frac{\epsilon_i^{(2)}}{8}\Delta x^2 \frac{\partial^3 \mathbf{Q}}{\partial t \partial x^2} = \mathbf{H} - \frac{\epsilon_e^{(4)}}{8}\frac{\Delta x^4}{\Delta t}\frac{\partial^4 \mathbf{Q}}{\partial x^4} \tag{3.14}$$

Here second-order implicit artificial viscosity is also added as a generalization. This implicit artificial viscosity term contains temporal differentiation, and hence does not affect the steady state solution. The delta-form of Eq. (3.14) for an implicit scheme can be written as

$$(\mathbf{I} - \Delta t \mathbf{D} + \Delta t \delta_x \mathbf{A} \cdot - \frac{\epsilon_i^{(2)}}{8}\delta_{xx})\Delta \mathbf{Q} = \Delta t \mathbf{H} - \Delta t \delta_x \mathbf{E} - \frac{\epsilon_e^{(4)}}{8}\Delta x^4 \delta_{xxxx}\mathbf{Q} \tag{3.15}$$

where the right-hand-side is evaluated at a known time-level and central-differencing is applied to all spatial operators. Performing Fourier decomposition gives

$$\left(\mathbf{I} - \Delta t \mathbf{D} + i\frac{\Delta t \mathbf{A}}{\Delta x}\sin(\omega \Delta x) + \frac{\epsilon_i^{(2)}}{4}(1 - \cos(\omega \Delta x))\right)(\mathbf{G} - \mathbf{I})$$
$$= \Delta t \mathbf{D} - i\frac{\Delta t \mathbf{A}}{\Delta x}\sin(\omega \Delta x) - \frac{\epsilon_e^{(4)}}{2}(1 - \cos(\omega \Delta x))^2$$

or,

$$\left(\mathbf{I} - \Delta t \mathbf{D} + i\frac{\Delta t \mathbf{A}}{\Delta x}\sin(\omega \Delta x) + \frac{\epsilon_i^{(2)}}{4}(1 - \cos(\omega \Delta x))\right)\mathbf{G}$$
$$= \left(1 + \frac{\epsilon_i^{(2)}}{4}(1 - \cos(\omega \Delta x)) - \frac{\epsilon_e^{(4)}}{2}(1 - \cos(\omega \Delta x))^2\right)\mathbf{I} \tag{3.16}$$

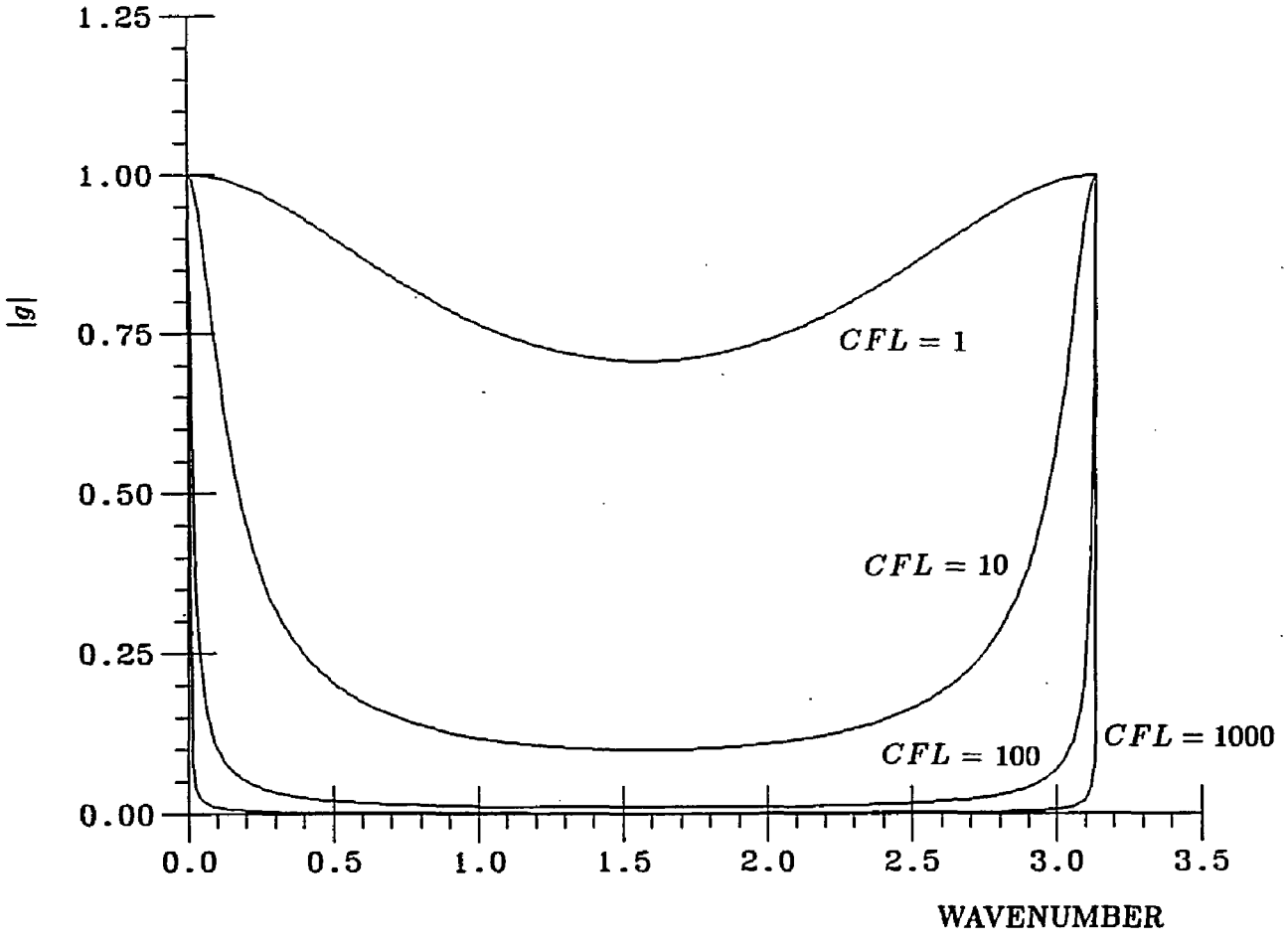

Figure 2: Amplification Factor of the Implicit Scheme for an One-dimensional Scalar Hyperbolic Equation. No Artificial Viscosity Added.

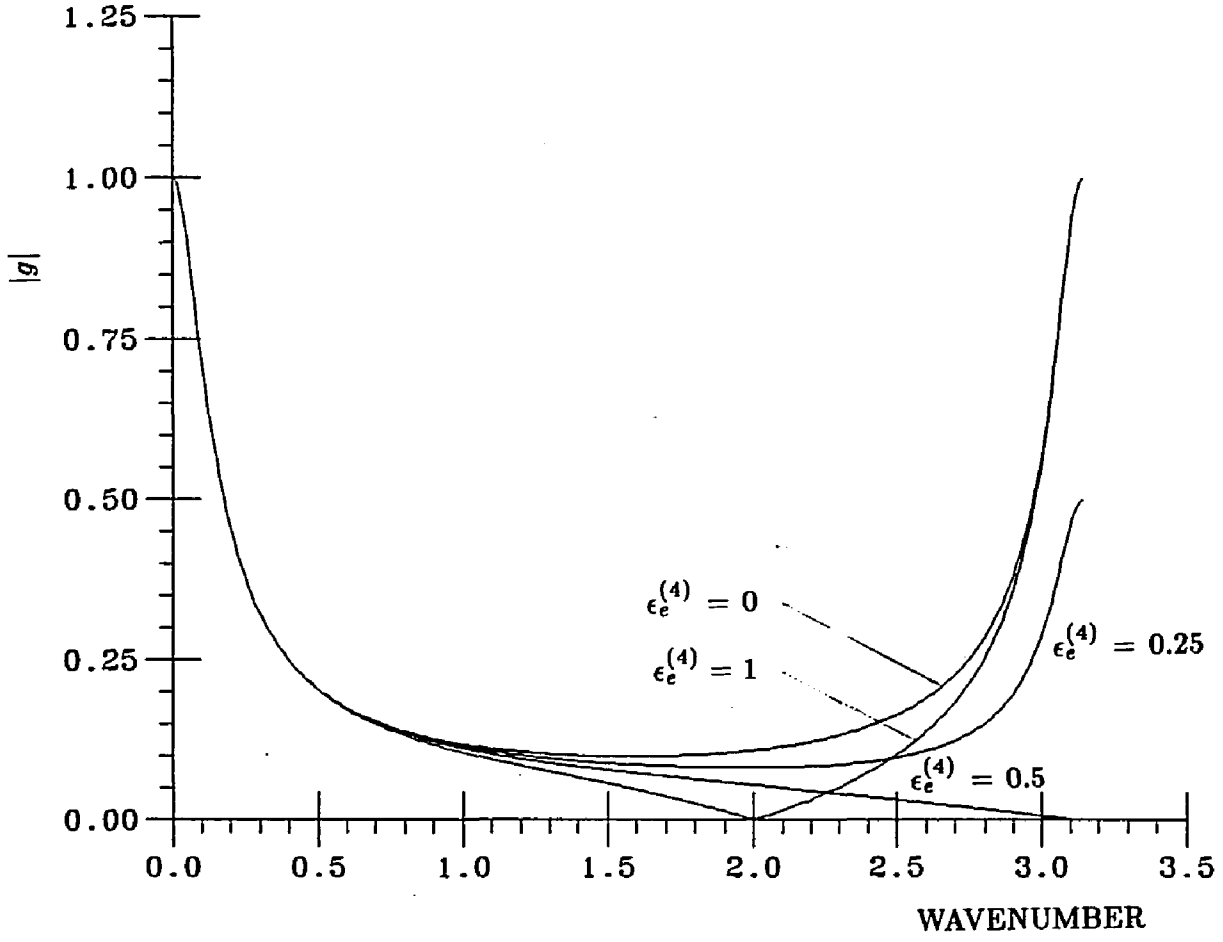

Figure 3: Amplification Factor of the Implicit Scheme for an One-dimensional Scalar Hyperbolic Equation. $CFL = 10$.

where the amplification matrix $\mathbf{G}$ is defined as

$$\mathbf{Q}^{n+1} = \mathbf{G}\mathbf{Q}^n$$

With an appropriate solution available, the matrices $\mathbf{A}$ and $\mathbf{D}$ can be calculated and $\mathbf{G}$ can be obtained numerically.

The stability analysis of a vector system such as Eq. (3.14) is often performed on a computer for specific cases with prescribed CFL number and $\epsilon_e^{(4)}$. A closed-form result is next to impossible if the vector size is more than two. However, an analysis for some specific wavenumbers can be performed. The stability analysis of the scalar equation suggests that the highest-wavenumber ($\omega\Delta x = \pi$) is critical to the implicit method. For this wavenumber, Eq. (3.16) becomes

$$\mathbf{K}_1\mathbf{G} = \mathbf{K}_2 \tag{3.17}$$

where

$$\mathbf{K}_1 = (1 + \frac{\epsilon_i^{(2)}}{2})\mathbf{I} - \Delta t\mathbf{D}, \qquad \mathbf{K}_2 = (1 + \frac{\epsilon_i^{(2)}}{2} - 2\epsilon_e^{(4)})\mathbf{I}$$

Taking the inverse of Eq. (3.17) gives

$$\begin{aligned}\mathbf{G}^{-1} &= \mathbf{K}_2^{-1}\mathbf{K}_1 \\ &= \frac{1}{1 + \epsilon_i^{(2)}/2 - 2\epsilon_e^{(4)}}\mathbf{K}_1\end{aligned} \tag{3.18}$$

For one-dimensional compressible flow,

$$\mathbf{K}_1 = \begin{pmatrix} 1 + \epsilon_i^{(2)}/2 & 0 & 0 \\ -\frac{\gamma-1}{2}\frac{1}{a}\frac{da}{dx}u^2\Delta t & 1 + \epsilon_i^{(2)}/2 + (\gamma-1)\frac{1}{a}\frac{da}{dx}u\Delta t & -(\gamma-1)\frac{1}{a}\frac{da}{dx}\Delta t \\ 0 & 0 & 1 + \epsilon_i^{(2)}/2 \end{pmatrix}$$

Thus the eigenvalues of $\mathbf{G}^{-1}$ are

$$\lambda_{1,2,3}^{-1} = \frac{1}{1 + \epsilon_i^{(2)}/2 - 2\epsilon_e^{(4)}}, \quad \frac{1 + \epsilon_i^{(2)}/2 + (\gamma-1)\frac{1}{a}\frac{da}{dx}u\Delta t}{1 + \epsilon_i^{(2)}/2 - 2\epsilon_e^{(4)}}, \quad \frac{1}{1 + \epsilon_i^{(2)}/2 - 2\epsilon_e^{(4)}}$$

The eigenvalues of G and $\mathbf{G}^{-1}$ are mutually reciprocal. Hence eigenvalues of G are

$$\lambda_{1,2,3} = 1 + \epsilon_i^{(2)}/2 - 2\epsilon_e^{(4)}, \quad \frac{1 + \epsilon_i^{(2)}/2 - 2\epsilon_e^{(4)}}{1 + (\gamma - 1)\frac{1}{a}\frac{da}{dx}u\Delta t}, \quad 1 + \epsilon_i^{(2)}/2 - 2\epsilon_e^{(4)} \tag{3.19}$$

In Eq. (3.19), λ_1 and λ_3 are constants and λ_2 is a variable. When the implicit viscosity is neglected, λ_1 and λ_3 require the range of $\epsilon_e^{(4)}$ to be

$$0 \leq \epsilon_e^{(4)} \leq 1$$

which conforms to the results from scalar stability analysis. The second eigenvalue, λ_2, represents the eigenvalue modified by the source term. Its magnitude is a function of the geometry, the velocity, and the time step. In the converging portion of the nozzle, da/dx is negative, the denominator is reduced, λ_2 is enlarged, and the scheme is destabilized.

Equation (3.19) offers useful information for selecting parameters for calculating compressible nozzle flows. It suggests that in order to obtain a maximum dissipation of the highest-wavenumber, the artificial viscosity should be chosen according to

$$1 + \epsilon_i^{(2)}/2 - 2\epsilon_e^{(4)} = 0$$

It also suggests that the CFL number can be increased when the mesh is refined. With Δt determined by the 'constant CFL' criterion (Eq. (3.9)), λ_2 can be rewritten as

$$\lambda_2 = \frac{1 + \epsilon_i^{(2)}/2 - 2\epsilon_e^{(4)}}{1 + \frac{\gamma-1}{1+M}\frac{1}{a}\frac{da}{dx}CFL\Delta x} \tag{3.20}$$

It is evident that as Δx is reduced, larger CFL number can be used.

For a one-dimensional incompressible flow, eigenvalues of the amplification matrix for the highest wavenumber are

$$\lambda_{1,2} = 1 + \epsilon_i^{(2)}/2 - 2\epsilon_e^{(4)}, \quad 1 + \epsilon_i^{(2)}/2 - 2\epsilon_e^{(4)} \tag{3.21}$$

which shows that the source term of one-dimensional incompressible nozzle flow does not destabilize the implicit scheme on the highest-wavenumber end. Hence the stability characteristic of the incompressible system for the highest-wavenumber is the same as that of the scalar model equation, Eq. (2.33). Thus it is likely that the CFL number is unlimited for the one-dimensional incompressible system. A stability analysis for all wavenumber based on Eq. (3.16) does show this.

3.3 Explicit Scheme

3.3.1 Time-Stepping Procedure

The idea of applying a Runge-Kutta explicit scheme to hyperbolic PDE's is borrowed from the Runge-Kutta scheme for ODE's [33]. An ODE has only one independent variable. The solution of an ODE is obtained by integration from a starting point, or, an initial condition. Since a transient hyperbolic PDE has a first order time-derivative, the concept of Runge-Kutta integration can also be applied with the time-derivative being treated as the independent variable of an ODE. The difference is that there are only source terms in an ODE, while a PDE may contain both source terms and space-derivatives.

The Runge-Kutta scheme for an ODE uses several predictor-corrector stages to integrate accurately. Once the Runge-Kutta scheme is applied to a hyperbolic PDE, the predictor-corrector procedure entrains information from neighboring grid points, and thus broadens the numerical domain of dependence. Hence, the chance of using a larger CFL number is increased. In Runge-Kutta scheme for ODE's, the predictor-corrector procedure is mainly for accuracy of integration. When the steady solution of a PDE is desired, the coefficients in the marching procedure can be chosen such that the fastest convergence can be obtained, without considering the timewise accuracy. On the other hand, this scheme also provides the possibility

of solving a transient problem accurately.

In Jameson's first Runge-Kutta explicit scheme paper [10], he proposed a three-stage and a four-stage procedure. The three-stage procedure as proposed by Jameson is

$$\begin{aligned} \mathbf{Q}^* &= \mathbf{Q}^n - \Delta t\, P\mathbf{Q}^n \\ \mathbf{Q}^{**} &= \mathbf{Q}^n - \frac{\Delta t}{2}(P\mathbf{Q}^n + P\mathbf{Q}^*) \\ \mathbf{Q}^{n+1} &= \mathbf{Q}^n - \frac{\Delta t}{2}(P\mathbf{Q}^n + P\mathbf{Q}^{**}) \end{aligned} \tag{3.22}$$

and the four-stage procedure is

$$\begin{aligned} \mathbf{Q}^* &= \mathbf{Q}^n - \frac{\Delta t}{2} P\mathbf{Q}^n \\ \mathbf{Q}^{**} &= \mathbf{Q}^n - \frac{\Delta t}{2} P\mathbf{Q}^* \\ \mathbf{Q}^{***} &= \mathbf{Q}^n - \Delta t\, P\mathbf{Q}^{**} \\ \mathbf{Q}^{n+1} &= \mathbf{Q}^n - \frac{\Delta t}{6}(P\mathbf{Q}^n + 2P\mathbf{Q}^* + 2P\mathbf{Q}^{**} + P\mathbf{Q}^{***}) \end{aligned} \tag{3.23}$$

where P represents the space-operator. For the one-dimensional nozzle flow, the space-operator can be

$$P\mathbf{Q} = \delta_x \mathbf{E} - \mathbf{H} + \frac{\epsilon_e^{(4)}}{8} \frac{\Delta x^4}{\Delta t} \delta_{xxxx} \mathbf{Q} \tag{3.24}$$

in which the fourth order viscosity is added.

Equation (3.22) requires three space-operations in each iteration, while Eq. (3.23) requires four. However, as can be seen from stability analysis, the four-stage procedure gives a larger CFL number limit, a stronger temporal damping, and therefore a faster convergence. Jameson claimed that the four-stage procedure takes less overall CPU-time than the three-stage procedure for the same level of convergence. As the number of stages increases, both the maximum CFL number and the work in each iteration increase. However, Jameson indicated that only

a marginal improvement can be obtained by increasing the number of stages. In the present work, most Runge-Kutta calculations employ the four-stage procedure. Jameson later on published another form of four-stage stepping procedure [34],

$$\begin{aligned} \mathbf{Q}^{*} &= \mathbf{Q}^{n} - \frac{\Delta t}{4} P\mathbf{Q}^{n} \\ \mathbf{Q}^{**} &= \mathbf{Q}^{n} - \frac{\Delta t}{3} P\mathbf{Q}^{*} \\ \mathbf{Q}^{***} &= \mathbf{Q}^{n} - \frac{\Delta t}{2} P\mathbf{Q}^{**} \\ \mathbf{Q}^{n+1} &= \mathbf{Q}^{n} - \Delta t\, P\mathbf{Q}^{***} \end{aligned} \tag{3.25}$$

It can be shown that Eq. (3.25) is really identical to Eq. (3.23). However, we will be using Eq. (3.25) throughout because it is more convenient to program and takes less computer-memory storage than Eq. (3.23).

3.3.2 Boundary Conditions

The application of the MOC boundary conditions to the implicit scheme was described in Section 3.2.2. Unlike most boundary conditions currently applied to implicit schemes, MOC boundary conditions require no extra assumption. The outgoing information is considered by the formulation comes directly from the governing equations. As a result, it can render residual waves non-reflective at the boundaries so that the convergence rate is not slowed down on the account of boundary conditions. The MOC boundary procedure can be applied to Runge-Kutta schemes with equal success.

The MOC boundary conditions are applied to Runge-Kutta schemes in a fashion that is analogous to that for implicit schemes. Consider one stage of the Runge-Kutta scheme,

$$-P\mathbf{Q} = RHS$$

Similar to the implicit scheme, this finite-difference equation is premultiplied by the inverse of the eigenmatrix, $\mathbf{M}^{-1}$, and a selection matrix, $\mathbf{L}$, in sequence. The result

is

$$\mathbf{LM^{-1}}(-PQ = RHS) \tag{3.26}$$

One may compare Eq. (3.26) with Eq. (3.10) for the implicit scheme. Equation (3.26) can be interpreted as a result of dropping terms containing Jacobians from Eq. (3.10). The dummy rows in Eq. (3.26) are to be filled up by appropriate boundary conditions in the form of Eq. (3.12). Solving this linear system is unrelated to the calculation of interior points, and can be carried out point by point along the boundaries. Equation (3.26) plus Equation (3.12) constitute the boundary condition for one stage in the Runge-Kutta scheme. To satisfy the CFL-condition, this boundary condition is applied to every stage to ensure stability.

3.3.3 Stability Analysis

We now carry out the scalar stability analysis for the four-stage Runge-Kutta scheme defined by Eq. (3.25). Stability analysis of the three-stage scheme defined by Eq. (3.22) is also performed for comparison.

Consider the one-dimension scalar wave equation with fourth-order artificial viscosity added, Eq. (2.33). Recall that this equation was used earlier for a stability analysis of the implicit scheme. Define the amplification factor of the operator $\Delta t P$ as Z. We first consider the case with $\epsilon_e^{(4)}$ being equal to zero. In such case,

$$Z = iCFL\ S \tag{3.27}$$

where S represents $\sin(\omega \Delta x)$.

For the three-stage scheme,

$$g^* = 1 - Z$$

$$g^{**} = 1 - \frac{1}{2}(Z + Zg^*)$$

$$g = 1 - \frac{1}{2}(Z + Zg^{**})$$

where

$$g^* \mathbf{Q}^n = \mathbf{Q}^*$$

$$g^{**} \mathbf{Q}^n = \mathbf{Q}^{**}$$

$$g \mathbf{Q} = \mathbf{Q}^{n+1}$$

The amplification factor, g, for the three-stage scheme is then

$$g = 1 - Z + \frac{1}{2}Z^2 - \frac{1}{4}Z^3 \tag{3.28}$$

By substituting Eq. (3.27) into Eq. (3.28), g is obtained as a function of CFL and wave mode,

$$g = (1 - \frac{1}{2}CFL^2 S^2) + i(-CFL\ S + \frac{1}{4}CFL^3 S^3) \tag{3.29}$$

Its magnitude is

$$\begin{aligned} |g| &= \sqrt{(1 - \frac{1}{2}CFL^2 S^2)^2 + (-CFL\ S + \frac{1}{4}CFL^3 S^3)^2} \\ &= \sqrt{1 - \frac{1}{4}(CFL\ S)^4 + \frac{1}{16}(CFL\ S)^6} \end{aligned} \tag{3.30}$$

By differentiating the right-hand-side of Eq. (3.30) once with respect to the wavenumber, one can show that the mid-wavenumber

$$\omega \Delta x = \frac{\pi}{2}$$

gives a maximum value of the amplification factor besides the lowest-wavenumber ($\omega \Delta x = 0$) and the highest-wavenumber ($\omega \Delta x = \pi$). Since g is identically one at the lowest- and the highest-wavenumbers, they need not be considered. At the mid-wavenumber, the amplification factor is

$$|g| = \sqrt{1 - \frac{1}{4}CFL^4 + \frac{1}{16}CFL^6} \tag{3.31}$$

The right-hand-side of Eq. (3.31) reaches one when CFL equals 2. Therefore, the three-stage scheme is conditionally stable for the wave equation. The CFL limit is 2 when no artificial viscosity is added.

The amplification curves of the three-stage Runge-Kutta scheme for some typical CFL number are shown on Fig. 4. One notices that the shapes of the amplification-factor curves for the explicit scheme are quite different from those of the implicit scheme as were shown in Fig. 2. The curves on Fig. 4 are always above a base line of $\sqrt{11/27}(\approx 0.64)$. As CFL increases, $|g|$ for the mid-wavenumber region decreases until the baseline is reached, and then reflects to approach the unstable zone.

Similar analysis for the four-stage scheme shows the amplification factor to be

$$g = 1 - Z + \frac{1}{2}Z^2 - \frac{1}{6}Z^3 + \frac{1}{24}Z^4 \tag{3.32}$$

By substituting Eq. (3.27) into Eq. (3.32), one gets

$$|g| = \sqrt{1 - \frac{1}{72}(CFL\ S)^6 + \frac{1}{576}(CFL\ S)^8} \tag{3.33}$$

Curves of $|g|$ for some typical CFL numbers are shown on Fig. 5. Just as for the three-stage scheme, the mid-wavenumber wave mode is also the most unstable for the four-stage scheme. The CFL limit of the four-stage scheme is found to be $2\sqrt{2}$. The shapes of the curves in Fig. 5 are similar to those for the three-stage scheme, with the baseline being reduced to 0.5.

As discussed in Section 2.2, the convergence rate appears to be controlled by the low wavenumber π/N, with N being the number of grid points. For a reasonably large N, this low wavenumber is very close to zero. Thus the slope of the amplification-factor curves near the lowest-wavenumber is an index of the convergence rate. In Fig. 4 and Fig. 5, one observes the nearly-zero slopes near the lowest-wavenumber. This indicates that the low-wavenumber modes have very small dissipation rates. In contrast, the amplification-factor curves for the implicit scheme (see Fig. 2 and Fig. 3) have much larger slopes near the lowest-wavenumber end. Accordingly, we can predict that the Runge-Kutta scheme will converge much slower

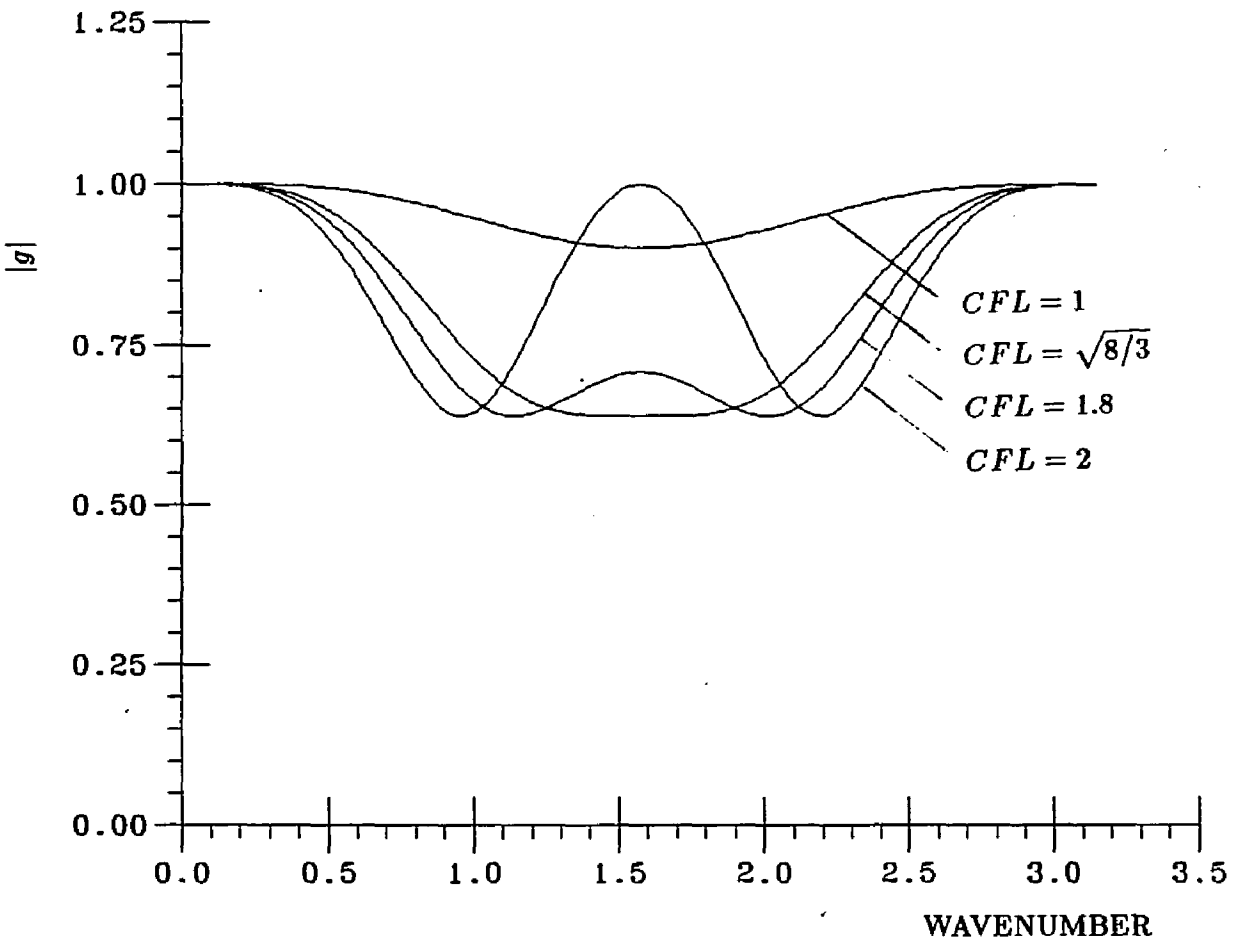

Figure 4: Amplification Factor of the 3-Stage Runge-Kutta Scheme for an One-dimensional Scalar Hyperbolic Equation.
No Artificial Viscosity Added.

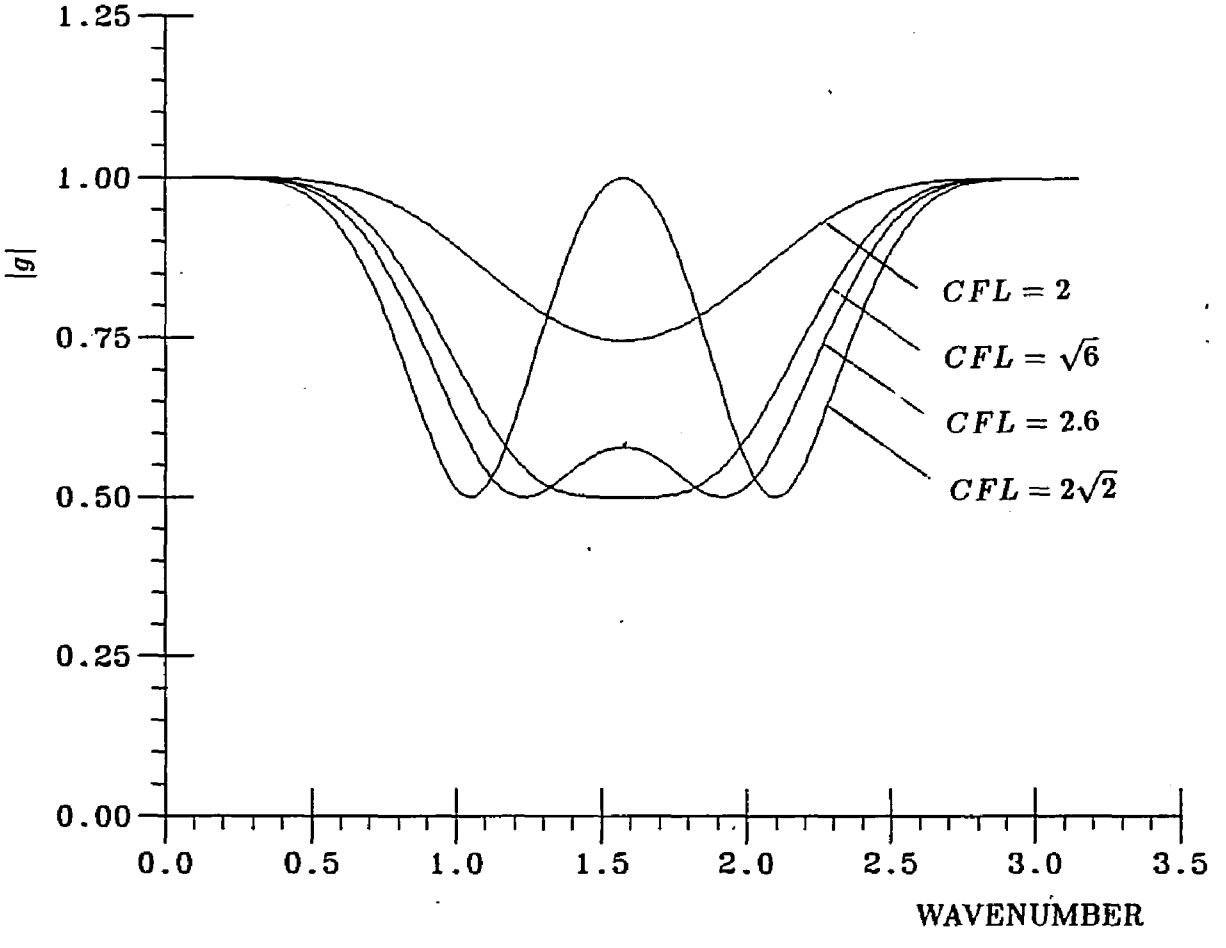

Figure 5: Amplification Factor of the 4-Stage Runge-Kutta Scheme for an One-dimensional Scalar Hyperbolic Equation.
No Artificial Viscosity Added.

than the implicit scheme for one-dimensional problems. The difference between the slopes of the three- and four-stage explicit scheme is less clear. However, since the baseline of curves for four-stage scheme is lower, it seems to have a higher dissipation rate.

Next consider the effect of artificial viscosity. With fourth-order artificial viscosity, Z becomes

$$Z = \frac{\epsilon_e^{(4)}}{2}(C-1)^2 + iCFL\ S \tag{3.34}$$

where C represents $\cos(\omega\Delta x)$. The amplification factor for the four-stage scheme is obtained by substituting Eq. (3.34) into Eq. (3.32). The resulting expression is quite complex, and is not displayed here. The curves for $|g|$ with artificial viscosity added are shown on Fig. 6 and Fig. 7. Figure 6 shows that $\epsilon_e^{(4)}$ can damp out high-wavenumber waves, but has little effect on low-wavenumber modes. It can be shown that the highest-wavenumber modes become unstable if $\epsilon_e^{(4)}$ exceeds approximately 1.4. Figure 7 shows that the CFL limit can increase slightly when artificial viscosity is added. However, no significant change for the amplification factor of the low-wavenumber modes is observed in both figures.

3.3.4 Convergence Acceleration

As shown by the stability analysis in the previous section, the convergence speed of the Runge-Kutta scheme appears to be quite slow. This motivates us to search for methods that accelerate convergence. One possible way to enhance the convergence rate is to increase the CFL limit. In many cases, a larger CFL number gives a smaller amplification factor, and therefore a faster convergence rate. The previous section indicates that adding artificial viscosity can only pull the $|g|$ curve away from neutral stability zone at the highest-wavenumber, but cannot relieve the CFL restriction significantly. Here we discuss one other effective strategy, namely, residual-smoothing [34].

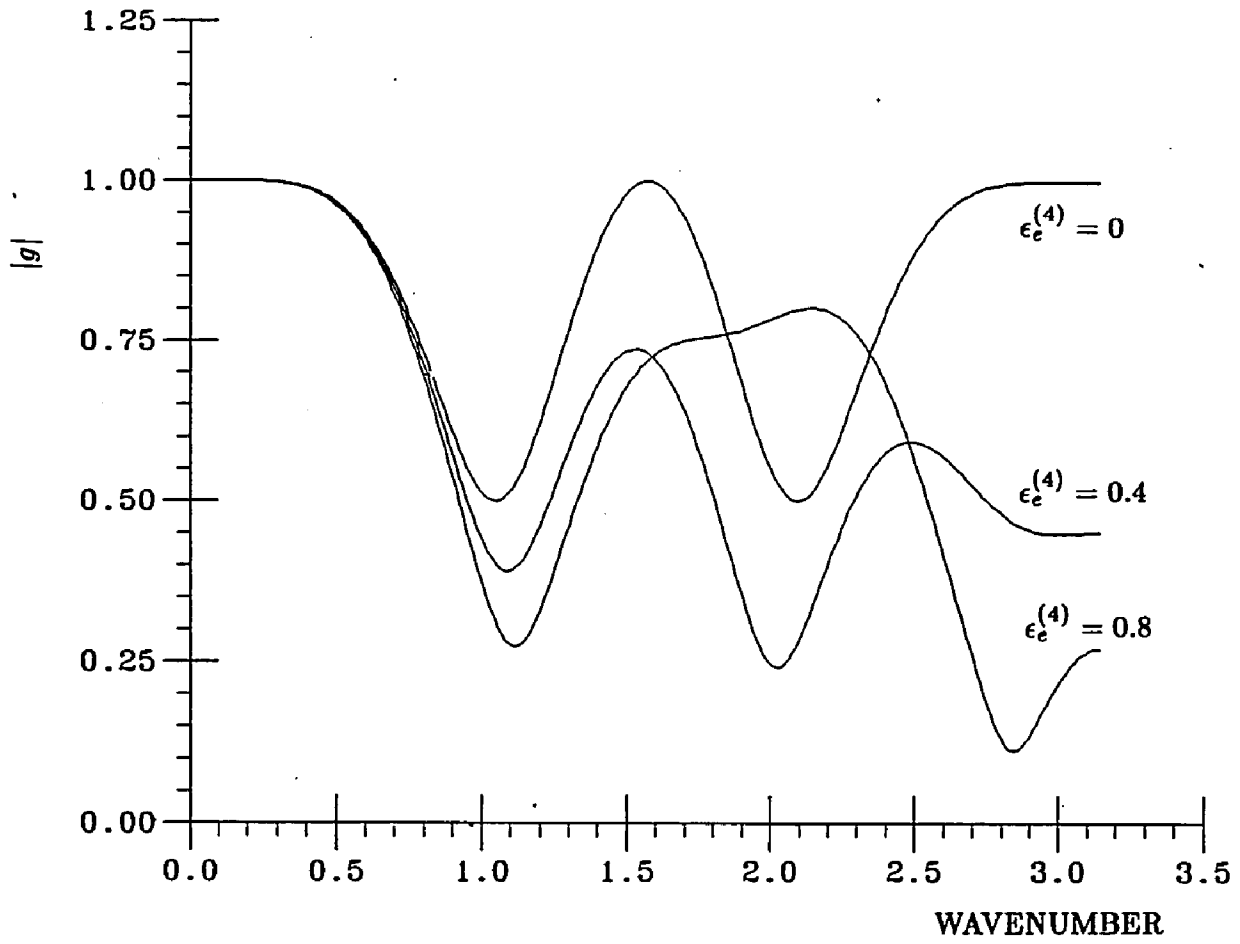

Figure 6: Amplification Factor of the 4-Stage Runge-Kutta Scheme for an One-dimensional Scalar Hyperbolic Equation. $CFL = 2\sqrt{2}$.

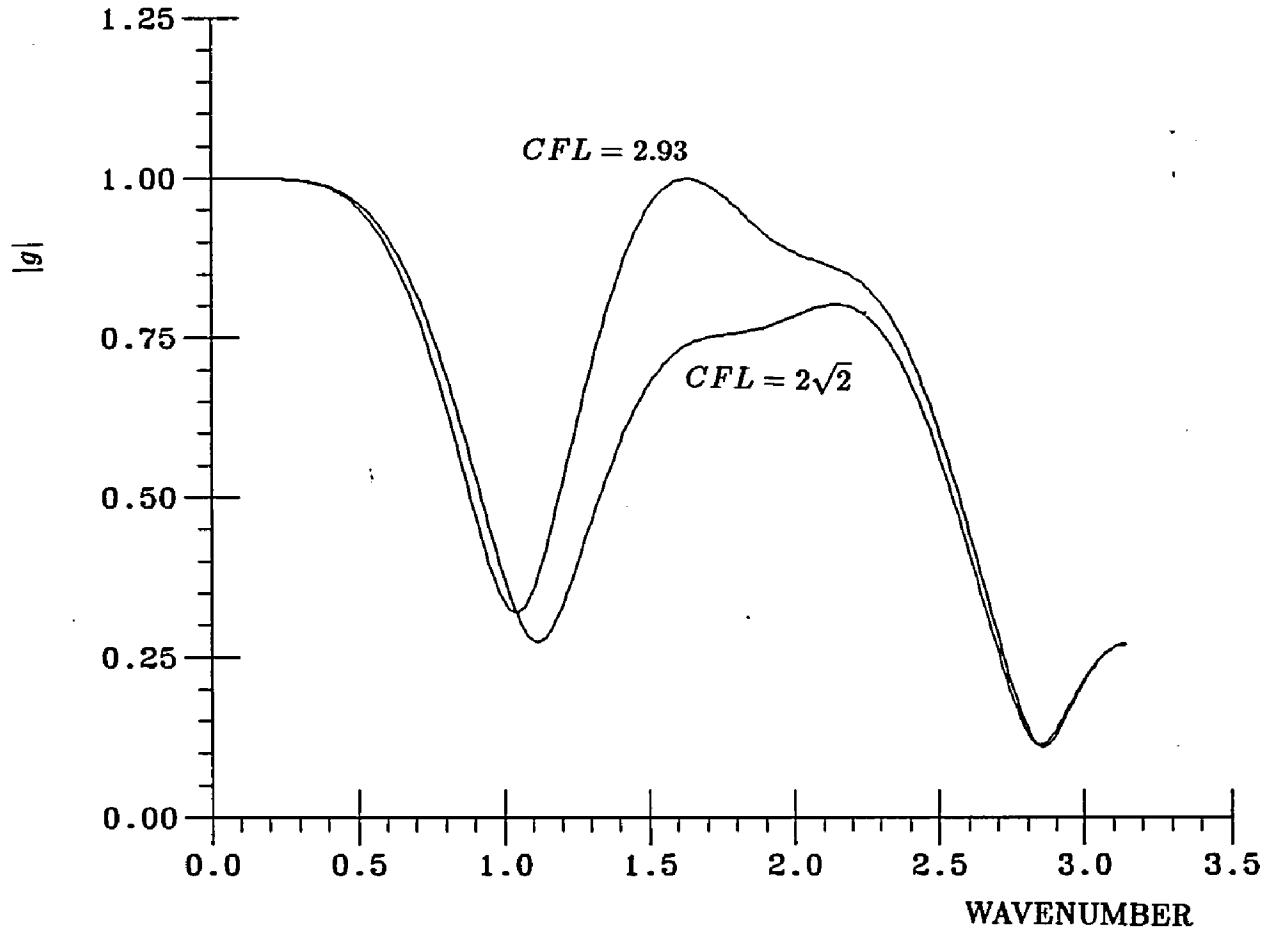

Figure 7: Amplification Factor of the 4-Stage Runge-Kutta Scheme for an One-dimensional Scalar Hyperbolic Equation. $\epsilon_e^{(4)} = 0.8$.

We may replace the residual by

$$\bar{P} \equiv (1 + \epsilon\delta_{xx})P$$

This increases the support of the difference scheme, and should increase the CFL limit. The amplification factor of the new operator, $\bar{P}$, is defined as $\bar{Z}$. It is related to the original amplification factor, Z, by

$$\bar{Z} = [1 - 2\epsilon\,(1 - C)]\,Z$$

If ϵ is selected properly, this method should increase the CFL limit and reduce the amplification factor. At the highest-wavenumber end,

$$\bar{Z} = (1 - 4\epsilon)\,Z$$

If ϵ is outside the range

$$0 \leq \epsilon \leq \frac{1}{2}$$

this smoothing procedure can destabilize the highest-wavenumber wave. Within this range the CFL limit can be increased by about one half.

To remove this restriction on ϵ one may perform the residual-smoothing implicitly, that is, using the following equation,

$$(1 - \epsilon\delta_{xx})\bar{P} = P \tag{3.35}$$

One may also interpret Eq. (3.35) as the implementation of the implicit second-order artificial viscosity to the explicit scheme. The amplification factor is now

$$\bar{Z} = \frac{Z}{1 + 2\epsilon(1 - C)} \tag{3.36}$$

If there is no artificial viscosity added,

$$\bar{Z} = i\frac{CFL\,S}{1 + 2\epsilon(1 - C)} \tag{3.37}$$

Equation (3.37) is substituted into Eq. (3.32) to obtain the amplification factor for the four-stage Runge-Kutta scheme. By differentiating the magnitude of the amplification factor once, one can show that the most unstable wave mode satisfies

$$C = \frac{2\epsilon}{1+2\epsilon}, \qquad S = \frac{\sqrt{1+4\epsilon}}{1+2\epsilon}$$

The amplification factor of each stage for this wave mode is

$$\bar{Z} = i\frac{CFL}{\sqrt{1+4\epsilon}} \tag{3.38}$$

If we define

$$\overline{CFL} \equiv \frac{CFL}{\sqrt{1+4\epsilon}} \tag{3.39}$$

then the rest of the stability analysis would be the same as in the previous section, with CFL being replaced by $\overline{CFL}$. As derived previously, the maximum CFL for the four-stage scheme without residual-smoothing is $2\sqrt{2}$. The maximum CFL is thus $2\sqrt{2}\sqrt{1+4\epsilon}$ with residual-smoothing. Equation (3.39) may also be written in the following form to show the gain of CFL number due to residual-smoothing method,

$$\overline{CFL} = \frac{CFL}{\text{gain}}$$

where

$$\text{gain} = \sqrt{1+4\epsilon}$$

Note here that using a larger CFL number does not always mean having a faster convergence. We should evaluate the convergence rate by the amplification factor. Several $|g|$ curves for some representative CFL numbers with $\epsilon = 1$ are shown in Fig. 8. The curve for the maximum CFL number, $2\sqrt{10}$, is also included. Compared to the curves without residual-smoothing (see Fig. 5), the curves in Fig. 8 have their peaks (the most unstable wavenumber) moved toward the low-wavenumber end. Such movements narrow the 'flat' region around the lowest-wavenumber, and

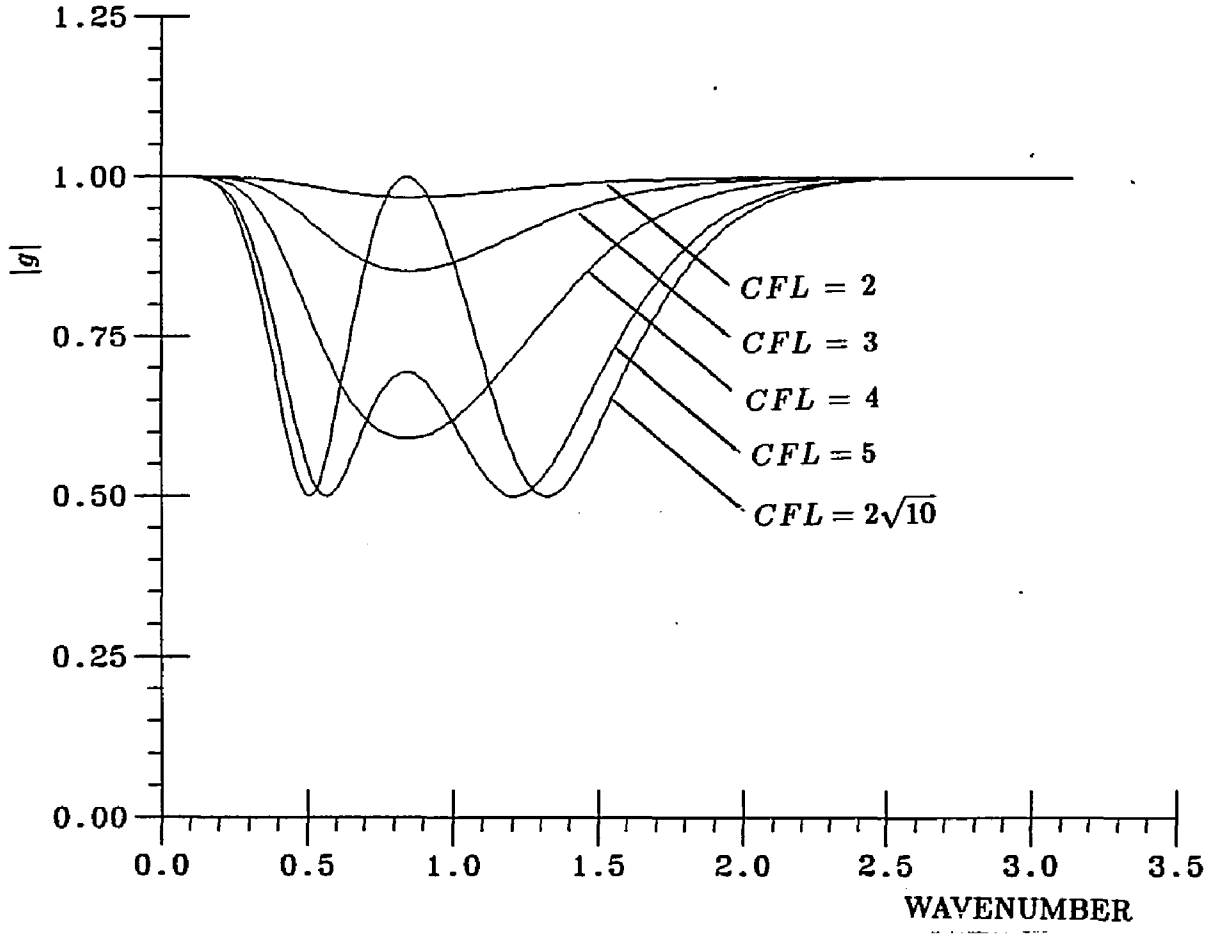

Figure 8: Amplification Factor of the 4-Stage Runge-Kutta Scheme for an One-dimensional Scalar Hyperbolic Equation. Residual-Smoothing Coefficient $\epsilon = 1$.

appear to reduce $|g|$ for the low-wavenumber modes. Heuristically, one may expect a faster convergence rate with residual-smoothing.

One important feature about residual-smoothing is that it allows the addition of larger viscosity. It was pointed out earlier that the maximum $\epsilon_e^{(4)}$ for stability is 1.4 without residual-smoothing. With the application of residual-smoothing ($\epsilon = 1$), the fourth-order viscosity can be increased by a factor of four without destabilizing the scheme. Figure 9 shows the amplification factor for various quantities of viscosity added, with a CFL number of $2\sqrt{10}$ and a residual-smoothing coefficient of one. These curves show that the maximum viscosity coefficient for stability is more than 6. It appears that the addition of viscosity can reduce the $|g|$ values of the low-wavenumber modes and can result in a faster convergence. This will be confirmed by numerical experiments.

Implementing residual-smoothing on the one-dimensional Runge-Kutta scheme is rather simple. The smoothing operator on the left-hand-side of Eq. (3.35) involves inverting a constant scalar-tridiagonal system for each dependent variable. This can be done by performing an LU-decomposition in advance and executing forward-backward substitutions for each dependent variable in each iteration. Inverting this system represents approximately 30-40% of the total CPU-time on a scalar machine for a typical one-dimensional problem.

3.4 Results and discussion

The one-dimensional Euler equations are exceedingly simple, but they can provide considerable insight into the capabilities of a numerical algorithm. As examples both the compressible and incompressible Euler equations were computed by implicit and explicit schemes for flows passing through a convergent-divergent nozzle. Figure 10 shows the area profile of the test nozzle. The area ratio which is defined as the ratio of throat area and the inlet area is 0.8. The profile is a third-order

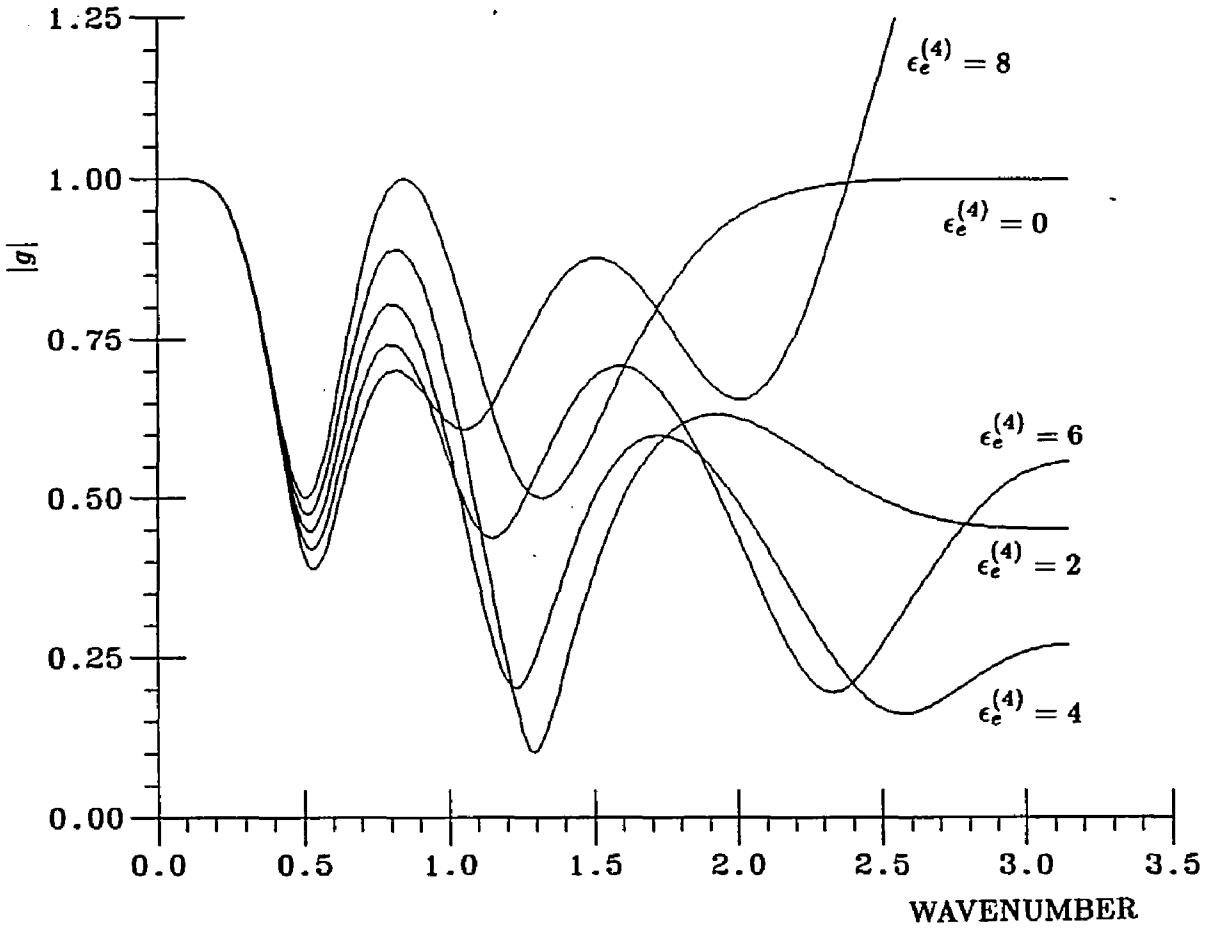

Figure 9: Amplification Factor of the 4-Stage Runge-Kutta Scheme for an One-dimensional Scalar Hyperbolic Equation. Residual-Smoothing Coefficient $\epsilon = 1$ and $CFL = 2\sqrt{10}$.

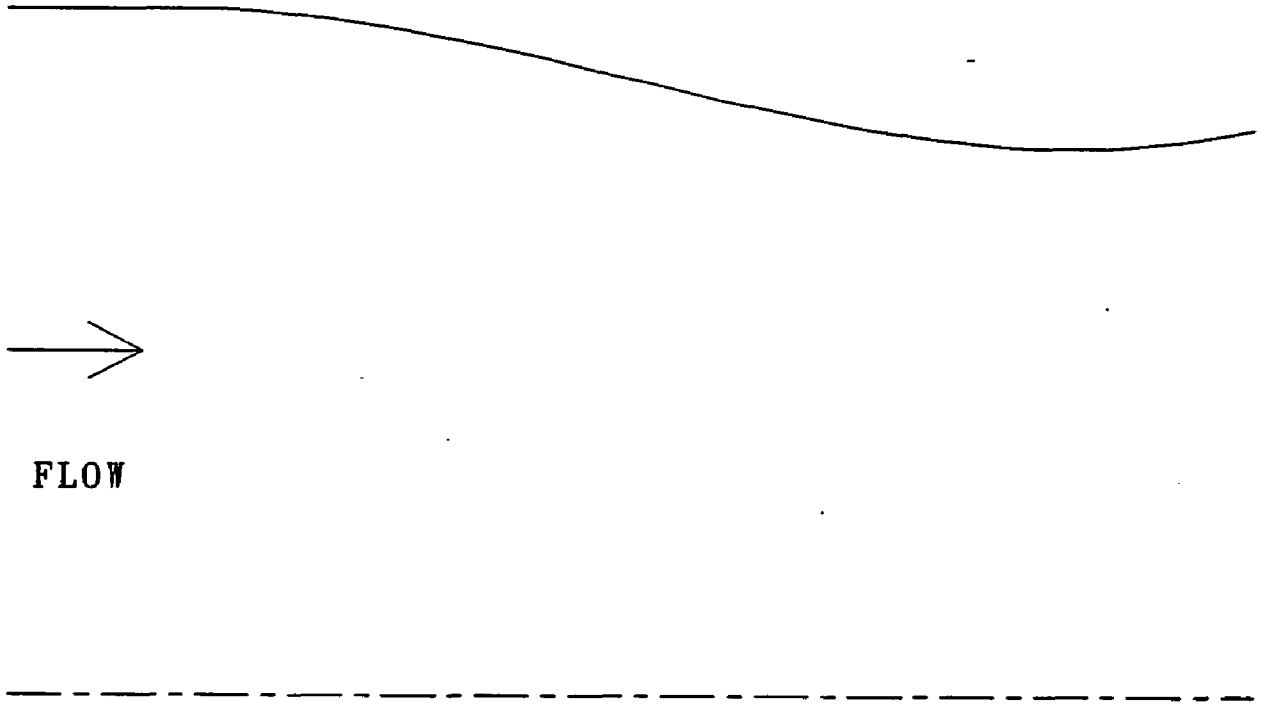

Figure 10: One-Dimensional Nozzle Area Variation. Area Ratio=0.8

polynomial patched with a constant area inlet. This polynomial is given as

$$-2(\mathrm{AR}-1)X^3+3(\mathrm{AR}-1)X^2+1$$

where AR is the area ratio.

The steady state results for compressible and incompressible flows are shown in Fig. 11 and Fig. 12, respectively. The nozzle is represented by 36 grid points with equal spacing. These results are computed by the implicit scheme. Results of the explicit scheme are identical to them, and are not displayed. Figure 11 shows the Mach number distribution of a choked flow, and Fig. 12 shows the pressure distribution of an incompressible flow. The numerical solutions are compared against analytical solutions, and both agree to four digits.

In order to prove that the accuracy of these central-difference schemes is of second-order, a series of calculations are carried out by bisecting the grid spacing successively. In principle, the numerical error should be proportional to the square of the grid spacing, and inversely proportional to the square of the number of grid points,

$$\text{error} = \text{const.}(\Delta x)^2 = \text{const.}(1/N)^2 \tag{3.40}$$

Taking the logarithm of both sides of Eq. (3.40) yields

$$\log_{10}(\text{error}) = \text{const.} - 2\log_{10} N$$

which indicates a linear relationship between $\log_{10}(\text{error})$ and $\log_{10} N$. The computational results of $\log_{10}(\text{error})$ are shown in Fig. 13 in terms of $\log_{10} N$. The numerical error is taken as the average of the absolute value of the difference between the numerical and analytical solutions. Indeed, both curves for compressible and incompressible flows are straight lines with slopes of -2. The second-order accuracy of the schemes is confirmed.

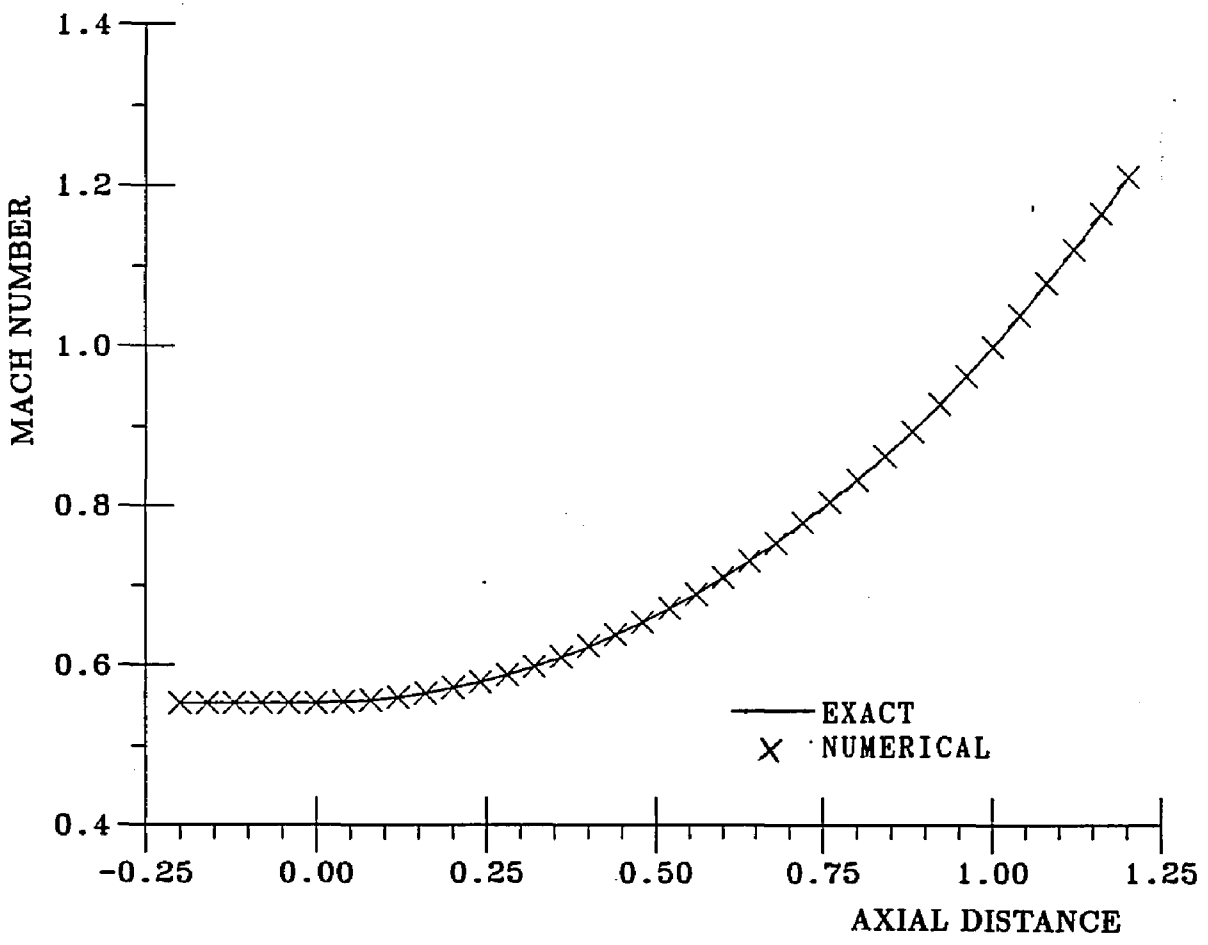

Figure 11: Mach Number Solution of a One-Dimensional Compressible Nozzle Flow.

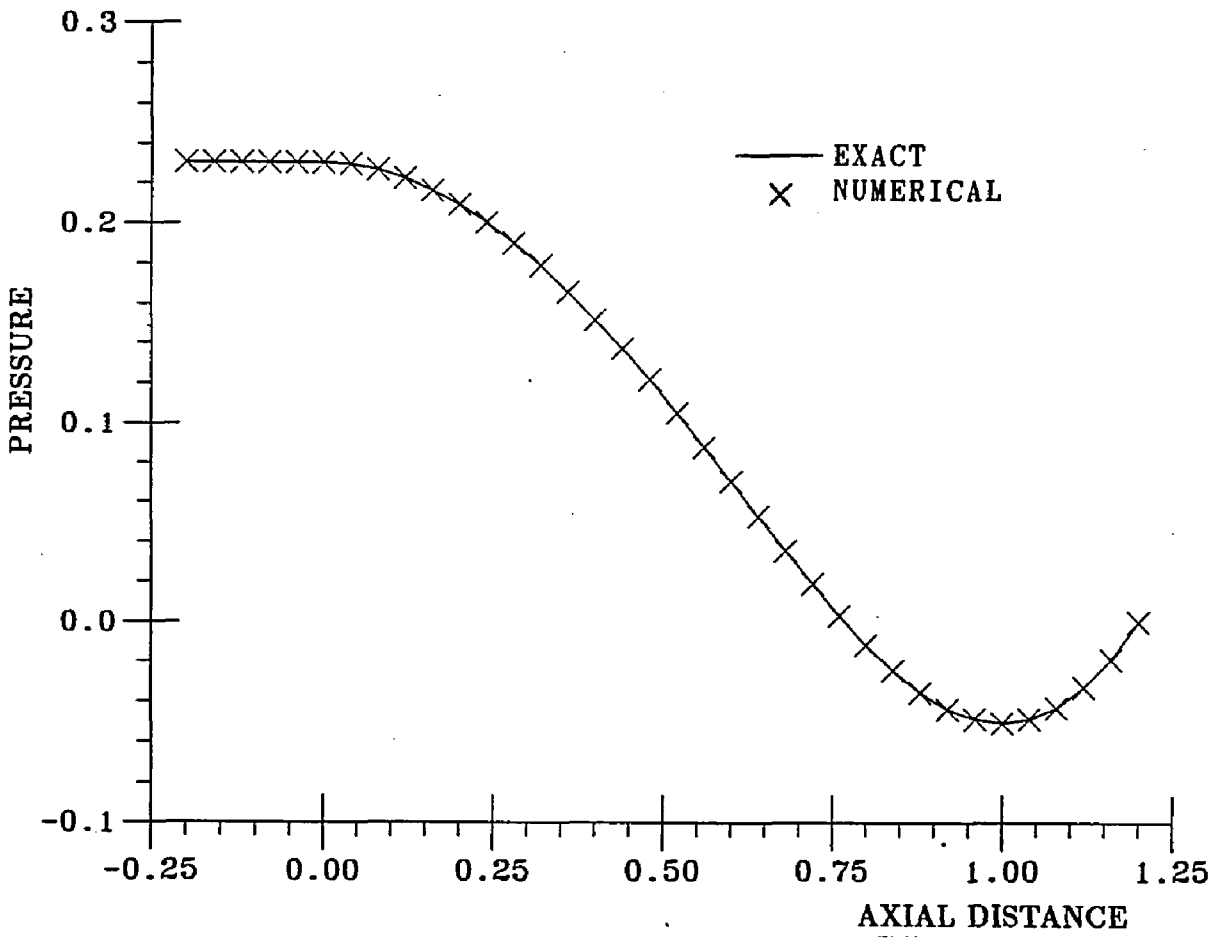

Figure 12: Pressure Distribution of an One-Dimensional Incompressible Nozzle Flow.

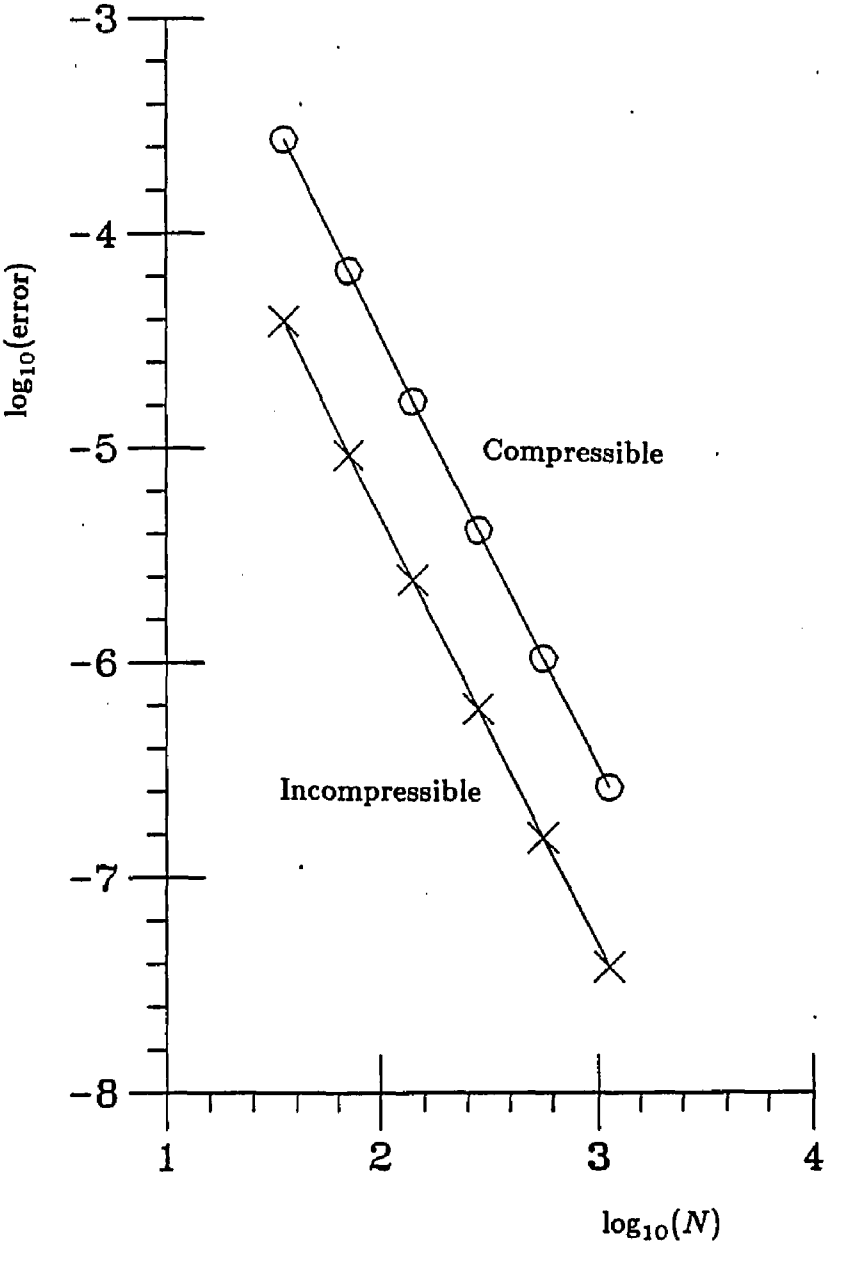

Figure 13: Convergence of Numerical Error of One-Dimensional Flows versus Number of Grid Points.

Regarding the convergence rates, implicit and explicit schemes are no contest for the one-dimensional Euler equations. Three convergence curves for a choked compressible flow are shown in Fig. 14. They are, in the sequence of ascending convergence speed, the explicit scheme without residual-smoothing, the explicit scheme with residual-smoothing, and the implicit scheme. Their respective CFL numbers are $2.8(\approx 2\sqrt{2})$, 6, and 100. As can be seen from these curves, the explicit scheme can be accelerated by a factor of greater than two if residual-smoothing is employed. Nevertheless, this effort does not make the explicit scheme comparable to the implicit scheme. The major reason for this is the time-step size limitation for the explicit scheme. Under this limited time-step size, the dissipation rate of the explicit scheme is quite low. On the other hand, the time-step sizes for the implicit scheme are limited only by the source term, and are not severe in this case. As shown in the stability analyses, the amplification-factor curves for the implicit scheme show a much stronger dissipation rate for a larger CFL number. A similar conclusion may be made for incompressible flows as well. However, the implicit scheme appears to be even more superior for incompressible flows because the source term of this system imposes no restriction on the CFL number of the implicit scheme, and the time-step limitation is completely removed.

The convergence rates sometimes can be affected by the amount of artificial viscosity added. From the amplification-factor curves it is not obvious how strong this effect can be. Accordingly, we study this based on some numerical experiments. We find that the fourth-order explicit viscosity does not affect the convergence rates of the implicit scheme or the explicit scheme without residual-smoothing significantly. (This is in agreement with our speculations of the low wavenumber region of the amplification-factor curves earlier.) The maximum convergence rate of the implicit scheme for a given CFL number occurs when the viscosity is added such that the

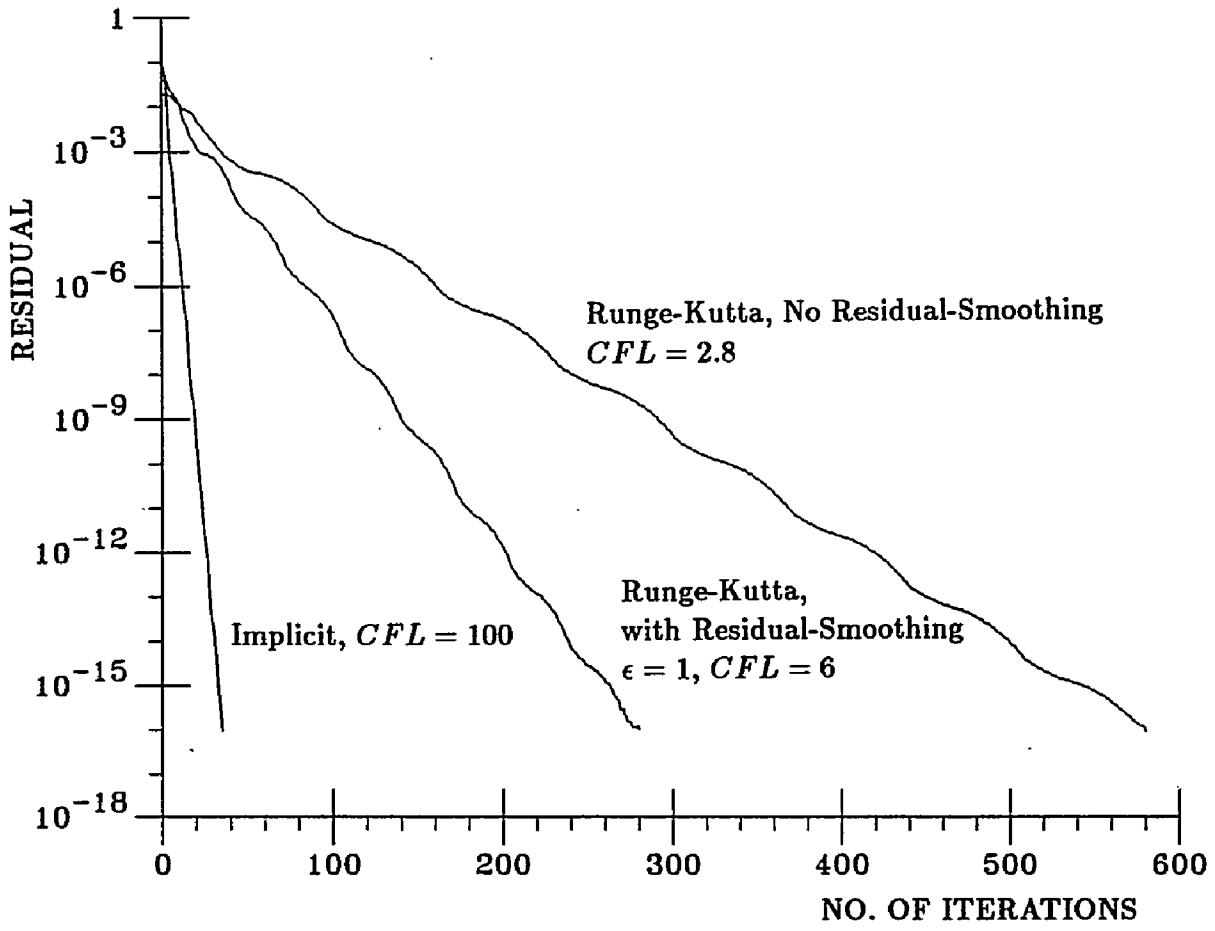

Figure 14: Convergence History of the Implicit and Explicit Schemes for an One-Dimensional Compressible Nozzle Flow.

highest-wavenumber has a zero $|g|$ (e. g., $\epsilon_i^{(2)} = 0, \epsilon_e^{(4)} = 0.5$). Addition of more explicit viscosity by means of increasing the implicit second-order viscosity does not enhance the convergence rate significantly. For the four-stage Runge-Kutta scheme without residual-smoothing, the convergence rate remains virtually unchanged for any amount of viscosity within a reasonably stable range (e. g., 0.1–1.4; refer to the stability analysis for the Runge-Kutta scheme). However, when residual-smoothing is applied to the Runge-Kutta scheme, the allowable viscosity increases, and the effect of viscosity on the convergence rate starts to show. In fact, the convergence curve shown in Fig. 14 for the Runge-Kutta scheme with residual-smoothing utilizes an $\epsilon_e^{(4)}$ of 6, which is much larger than the other two curves. Several convergence curves obtained by applying different amount of viscosity for $\epsilon = 1$ are shown in Fig. 15. They reveal that within a stable region, the Runge-Kutta scheme can be accelerated by a stronger fourth-order viscosity allowed by the application of residual-smoothing.

One final aspect should be mentioned regarding the influence of boundary conditions on solution quality. The outgoing information on the boundaries is extracted by taking one-sided differences. The order of the one-sided differencing can affect the amplitude of fluctuations. Figure 16 shows the mass flux fluctuation of the steady state solution for an incompressible flow. (Note that Fig. 16 uses an enlarged scale, and does not represent a poor mass conservation.) The fluctuation is caused by the so-called odd-even decoupling of the central-difference formulation. The steady state finite-difference equation enforces flux conservation only on every other grid point. The zigzag type of flux is thus allowed as a solution. However, this fluctuation is reduced by employing second-order one-sided differencing on the boundaries. The relative advantage of the second order boundary condition is clearly seen in Fig. 16.

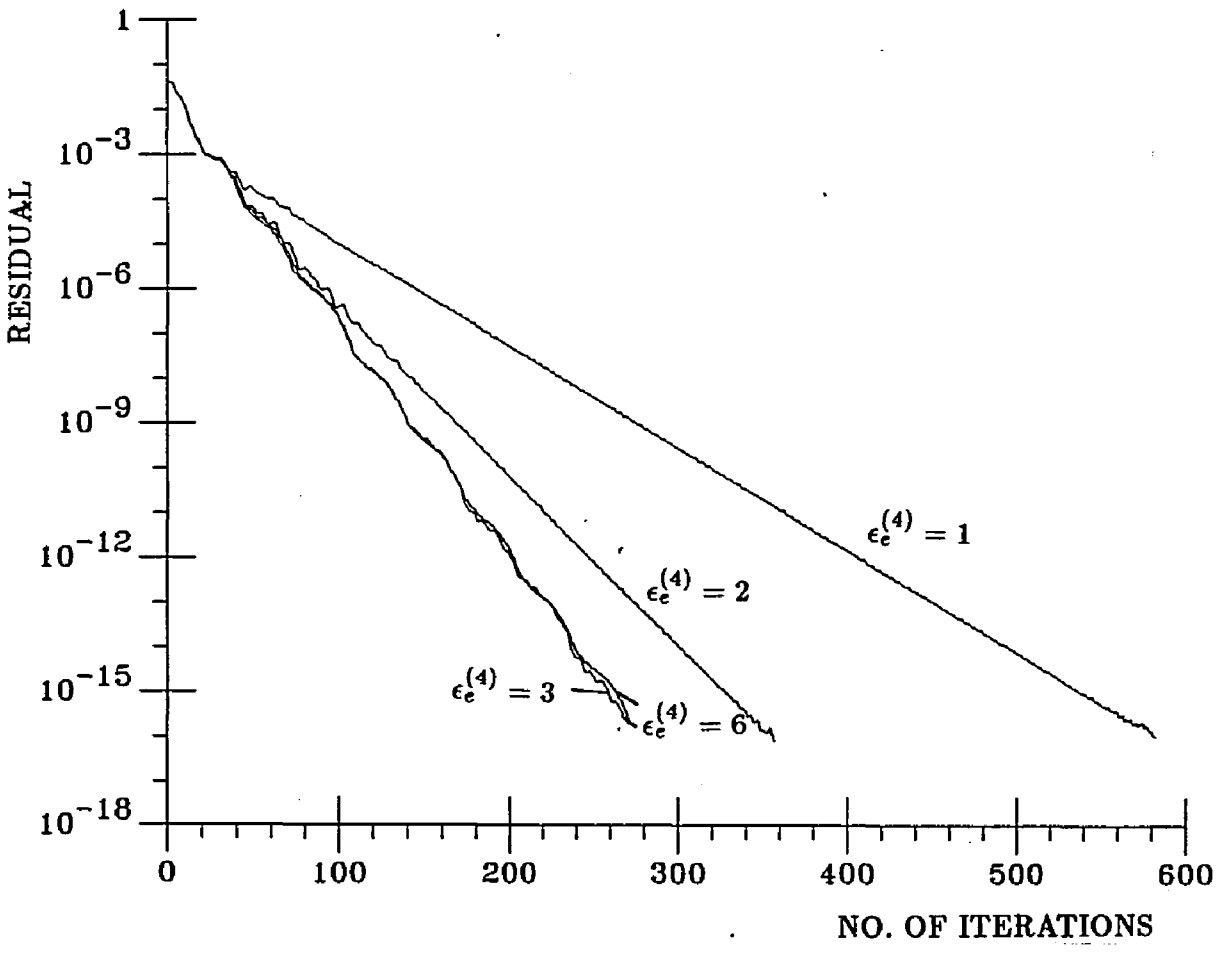

Figure 15: Convergence History of the Runge-Kutta Scheme with Different Artificial Viscosity Coefficients for an One-Dimensional Compressible Nozzle Flow. $\epsilon = 1$, $CFL = 6$

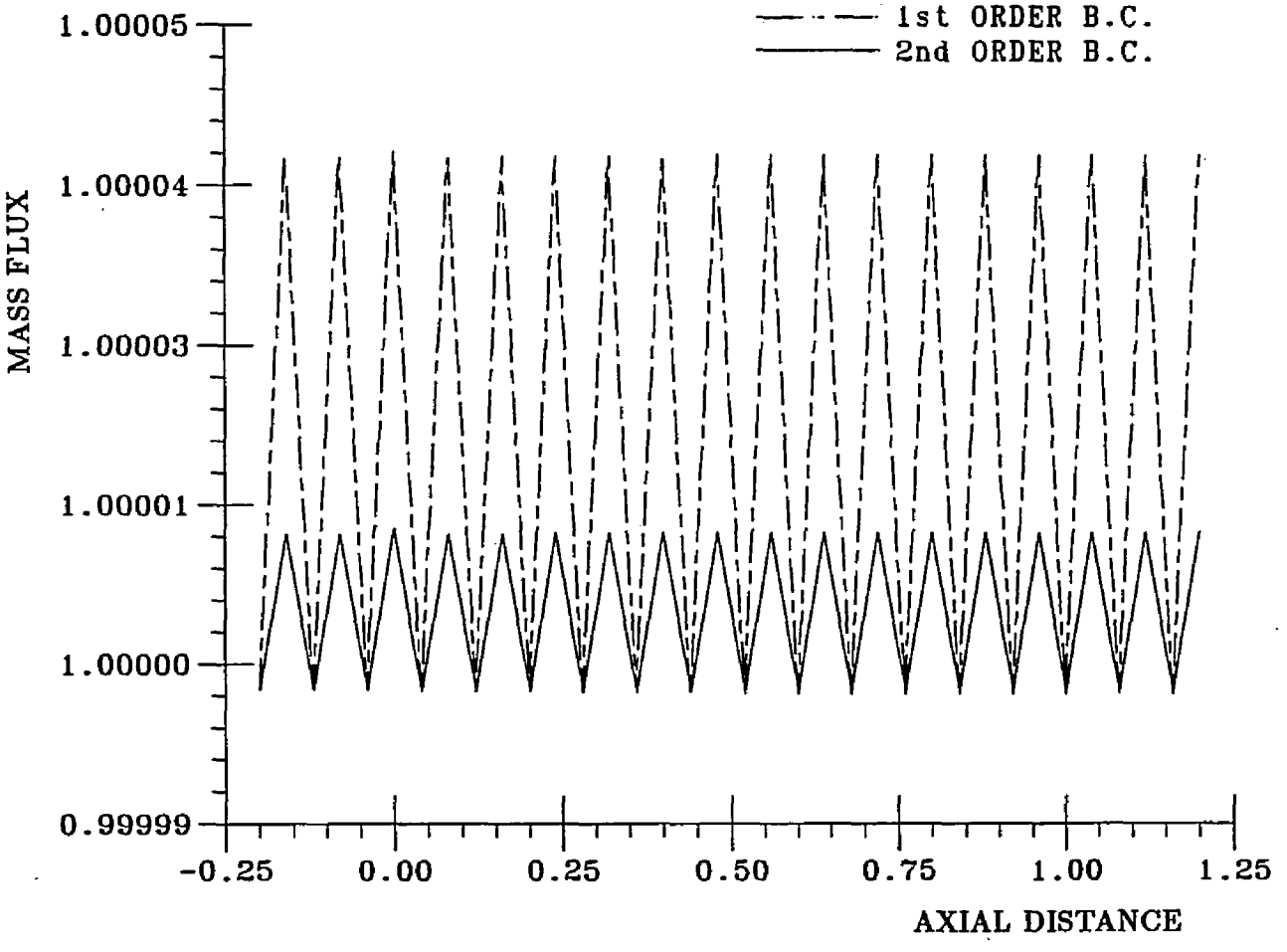

Figure 16: Mass Flux Oscillation of a Central-Difference Scheme.

CHAPTER 4
TWO-DIMENSIONAL PROBLEMS

4.1 Governing Equations

The equations governing a two-dimensional inviscid flow can be expressed in the following conservative form [35].

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{E}(\mathbf{Q})}{\partial x} + \frac{\partial \mathbf{F}(\mathbf{Q})}{\partial y} = \mathbf{H} \tag{4.1}$$

Note that this is the same as the one-dimensional governing equation (see Eq. (3.1)) except for the addition of $\partial \mathbf{F}/\partial y$. With the flux Jacobians $\mathbf{A}$ and $\mathbf{B}$ being defined as

$$\mathbf{A} \equiv \frac{\partial \mathbf{E}}{\partial \mathbf{Q}}$$

$$\mathbf{B} \equiv \frac{\partial \mathbf{F}}{\partial \mathbf{Q}}$$

Eq. (4.1) can be rewritten in the non-conservative form as

$$\frac{\partial \mathbf{Q}}{\partial t} + \mathbf{A}\frac{\partial \mathbf{Q}}{\partial x} + \mathbf{B}\frac{\partial \mathbf{Q}}{\partial y} = \mathbf{H} \tag{4.2}$$

Equation (4.1) is written on Cartesian coordinates. For computing flows around irregularly shaped bodies, it is inconvenient and inaccurate to apply boundary conditions if a rectangular grid on the physical domain is used, because in general the grid points do not fall on the body surfaces. It is therefore extremely helpful to invoke body-fitted, generalized grid systems. To do so, the governing equations must be transformed from (x, y) onto a body-fitted coordinate system (ξ, η) which is called the computational domain. For convenience, the grid system on the computational domain can be equally spaced. Grid points on the physical domain can be either moving or fixed (with respect to time). If the grid points are moving, ξ

and η are functions of x, y, and t; if the grid points are fixed, ξ and η are functions of x and y only.

Consider a mapping of a fixed grid from the Cartesian coordinates (x, y) to the general coordinates (ξ, η). Under this mapping, the conservative form of the governing equation is transformed by using the chain rule.

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{E}}{\partial \xi}\xi_x + \frac{\partial \mathbf{E}}{\partial \eta}\eta_x + \frac{\partial \mathbf{F}}{\partial \xi}\xi_y + \frac{\partial \mathbf{F}}{\partial \eta}\eta_y = \mathbf{H}$$

Dividing through by the Jacobian of transformation, J.

$$\frac{\partial (\mathbf{Q}/J)}{\partial t} + \frac{\partial \mathbf{E}}{\partial \xi}\frac{\xi_x}{J} + \frac{\partial \mathbf{E}}{\partial \eta}\frac{\eta_x}{J} + \frac{\partial \mathbf{F}}{\partial \xi}\frac{\xi_y}{J} + \frac{\partial \mathbf{F}}{\partial \eta}\frac{\eta_y}{J} = \frac{\mathbf{H}}{J}$$

and rearranging, we get

$$\begin{aligned}\frac{\partial (\mathbf{Q}/J)}{\partial t} &+ \frac{\partial}{\partial \xi}\left[\frac{1}{J}\left(\mathbf{E}\xi_x + \mathbf{F}\xi_y\right)\right] + \frac{\partial}{\partial \eta}\left[\frac{1}{J}\left(\mathbf{E}\eta_x + \mathbf{F}\eta_y\right)\right] \\ &-\mathbf{E}\left[\frac{\partial}{\partial \xi}\left(\frac{\xi_x}{J}\right) + \frac{\partial}{\partial \eta}\left(\frac{\eta_x}{J}\right)\right] - \mathbf{F}\left[\frac{\partial}{\partial \xi}\left(\frac{\xi_y}{J}\right) + \frac{\partial}{\partial \eta}\left(\frac{\eta_y}{J}\right)\right] = \frac{\mathbf{H}}{J}\end{aligned}$$

Since it can be shown (see Appendix) that

$$\frac{\partial}{\partial \xi}\left(\frac{\xi_x}{J}\right) + \frac{\partial}{\partial \eta}\left(\frac{\eta_x}{J}\right) = 0$$

and

$$\frac{\partial}{\partial \xi}\left(\frac{\xi_y}{J}\right) + \frac{\partial}{\partial \eta}\left(\frac{\eta_y}{J}\right) = 0$$

we can rewrite the above equation in the fully conservative form,

$$\frac{\partial (\mathbf{Q}/J)}{\partial t} + \frac{\partial}{\partial \xi}\left[\frac{1}{J}\left(\mathbf{E}\xi_x + \mathbf{F}\xi_y\right)\right] + \frac{\partial}{\partial \eta}\left[\frac{1}{J}\left(\mathbf{E}\eta_x + \mathbf{F}\eta_y\right)\right] = \frac{\mathbf{H}}{J}$$

Then by defining the following:

$$\hat{\mathbf{Q}} \equiv \frac{1}{J}\mathbf{Q}$$

$$\hat{\mathbf{E}} \equiv \frac{1}{J}\left(\mathbf{E}\xi_x + \mathbf{F}\xi_y\right)$$

$$\hat{\mathbf{F}} \equiv \frac{1}{J}\left(\mathbf{E}\eta_x + \mathbf{F}\eta_y\right)$$

$$\hat{\mathbf{H}} \equiv \frac{1}{J}\mathbf{H}$$

The strong-conservation form of the governing equation in the transformed coordinate system is obtained [37].

$$\frac{\partial \hat{\mathbf{Q}}}{\partial t} + \frac{\partial \hat{\mathbf{E}}}{\partial \xi} + \frac{\partial \hat{\mathbf{F}}}{\partial \eta} = \hat{\mathbf{H}} \tag{4.3}$$

Its non-conservative form can be written as

$$\frac{\partial \hat{\mathbf{Q}}}{\partial t} + \hat{\mathbf{A}}\frac{\partial \hat{\mathbf{Q}}}{\partial \xi} + \hat{\mathbf{B}}\frac{\partial \hat{\mathbf{Q}}}{\partial \eta} = \hat{\mathbf{H}} \tag{4.4}$$

The Jacobians, $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$, in the (ξ, η) coordinates, can be related to their counterparts $\mathbf{A}$ and $\mathbf{B}$ in Cartesian coordinates by the following equations.

$$\begin{aligned}
\hat{\mathbf{A}} &= \partial \hat{\mathbf{E}} / \partial \hat{\mathbf{Q}} \\
&= \partial \left(\mathbf{E}\xi_x + \mathbf{F}\xi_y \right) / \partial \mathbf{Q} \\
&= \mathbf{A}\xi_x + \mathbf{B}\xi_y \\
\hat{\mathbf{B}} &= \mathbf{A}\eta_x + \mathbf{B}\eta_y
\end{aligned}$$

The above derivation which results in the strongly conservative equation, Eq. (4.3), is suitable for both incompressible and compressible Euler equations. Note that we have not yet mentioned the elements in the vectors $\mathbf{Q}$, $\mathbf{E}$, $\mathbf{F}$, and $\mathbf{H}$. The system of equations that describe incompressible and compressible inviscid flows are now discussed.

For the incompressible case, the elements in the vectors $\mathbf{Q}_i$, $\mathbf{E}_i$, $\mathbf{F}_i$, and $\mathbf{H}_i$ are given as

$$\mathbf{Q}_i = \begin{pmatrix} p \\ u \\ v \end{pmatrix}, \quad \mathbf{E}_i = \begin{pmatrix} \beta u \\ u^2 + p \\ uv \end{pmatrix}, \quad \mathbf{F}_i = \begin{pmatrix} \beta v \\ uv \\ v^2 + p \end{pmatrix}, \quad \mathbf{H}_i = \mathbf{0} \tag{4.5}$$

where p is the pressure divided by the density, and β is the pseudocompressibility coefficient. Once again, the subscript i denotes 'incompressible'. As mentioned in the previous chapter, the first element of the primary dependent variable, p, is

artificial. The temporal term in the continuity equation, $\partial p/\partial t$, is introduced to make the system hyperbolic in time. When the system reaches steady state, all time derivative terms drop out, and the continuity equation is recovered.

The incompressible flux Jacobians $\mathbf{A}_i$ and $\mathbf{B}_i$ can be derived from their definition. They are given as

$$\mathbf{A}_i = \begin{pmatrix} 0 & \beta & 0 \\ 1 & 2u & 0 \\ 0 & v & u \end{pmatrix}, \quad \mathbf{B}_i = \begin{pmatrix} 0 & 0 & \beta \\ 0 & v & u \\ 1 & 0 & 2v \end{pmatrix} \tag{4.6}$$

The three eigenvalues of $\mathbf{A}_i$ are the real numbers — u, $u + c_x$, and $u - c_x$. The pseudoacoustic speed in the x-direction, c_x, is defined as

$$c_x \equiv \sqrt{u^2 + \beta}$$

Similarly, $\mathbf{B}_i$ also has three real eigenvalues — v, $v + c_y$, and $v - c_y$, where

$$c_y \equiv \sqrt{v^2 + \beta}$$

If β is positive, the wave propagation speeds in both directions are greater than the flow velocities in the corresponding directions. This situation also appears in one-dimensional incompressible flow, and is analogous to a subsonic compressible flow.

The source Jacobian, $\mathbf{D}_i$, which is defined as

$$\mathbf{D}_i = \frac{\partial \mathbf{H}_i}{\partial \mathbf{Q}_i}$$

is needed in the implicit scheme when the source vector is non-zero. For the two-dimensional incompressible inviscid system written in Cartesian coordinates, no source term occurs, and $\mathbf{D}_i$ is a 3×3 zero matrix.

In the general coordinate system, the vectors and matrices for the incompressible system are given as

$$\hat{\mathbf{Q}}_i = \frac{1}{J}\begin{pmatrix} p \\ u \\ v \end{pmatrix}, \quad \hat{\mathbf{E}}_i = \frac{1}{J}\begin{pmatrix} \beta U \\ uU + \xi_x p \\ vU + \xi_y p \end{pmatrix}, \quad \hat{\mathbf{F}}_i = \frac{1}{J}\begin{pmatrix} \beta V \\ uV + \eta_x p \\ vV + \eta_y p \end{pmatrix}, \quad \hat{\mathbf{H}}_i = 0$$

and

$$\hat{\mathbf{A}}_i = \begin{pmatrix} 0 & \beta\xi_x & \beta\xi_y \\ \xi_x & U + \xi_x u & \xi_y u \\ \xi_y & \xi_x v & U + \xi_y v \end{pmatrix}, \quad \hat{\mathbf{B}}_i = \begin{pmatrix} 0 & \beta\eta_x & \beta\eta_y \\ \eta_x & V + \eta_x u & \eta_y u \\ \eta_y & \eta_x v & V + \eta_y v \end{pmatrix}$$

where U and V are the contravariant velocity components in (ξ, η) coordinates. They are defined as

$$U \equiv u\xi_x + v\xi_y \tag{4.7}$$

$$V \equiv u\eta_x + v\eta_y \tag{4.8}$$

Notice that $\hat{\mathbf{F}}_i$ and $\hat{\mathbf{B}}_i$ can be obtained by replacing ξ in $\hat{\mathbf{E}}_i$ and $\hat{\mathbf{A}}_i$ by η.

The eigenvalues of $\hat{\mathbf{A}}_i$ and $\hat{\mathbf{B}}_i$ are given as

$$\lambda_{1,2,3}(\hat{\mathbf{A}}_i) = U, U + C_\xi, U - C_\xi \tag{4.9a}$$

and

$$\lambda_{1,2,3}(\hat{\mathbf{B}}_i) = V, V + C_\eta, V - C_\eta \tag{4.9b}$$

where C_ξ and C_η are the pseudoacoustic speeds in the ξ and η directions as defined by

$$C_\xi \equiv \sqrt{U^2 + \beta(\xi_x^2 + \xi_y^2)}$$

and

$$C_\eta \equiv \sqrt{V^2 + \beta(\eta_x^2 + \eta_y^2)}$$

As before, if β is positive, $U + C_\xi$ and $V + C_\eta$ are positive, and $U - C_\xi$ and $V - C_\eta$ are negative.

For compressible flows, the energy equation must be included. If the ideal gas relationship is assumed, the vectors and matrices in Eq. (4.1) and Eq. (4.2) are as listed below:

$$\mathbf{Q}_c = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ e \end{pmatrix}, \quad \mathbf{E}_c = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (e+p)u \end{pmatrix}, \quad \mathbf{F}_c = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (e+p)v \end{pmatrix}, \quad \mathbf{H}_c = \mathbf{0}$$

$$\mathbf{A}_c = \begin{pmatrix} 0 & 1 & 0 & 0 \\ \frac{\gamma-3}{2}u^2 + \frac{\gamma-1}{2}v^2 & (3-\gamma)u & (1-\gamma)v & (\gamma-1) \\ -uv & v & u & 0 \\ [-\frac{\gamma e}{\rho} + (\gamma-1)(u^2+v^2)]u & \frac{\gamma e}{\rho} - \frac{\gamma-1}{2}(3u^2+v^2) & (1-\gamma)uv & \gamma u \end{pmatrix}$$

$$\mathbf{B}_c = \begin{pmatrix} 0 & 0 & 1 & 0 \\ -uv & v & u & 0 \\ \frac{\gamma-1}{2}u^2 + \frac{\gamma-3}{2}v^2 & (1-\gamma)u & (3-\gamma)v & (\gamma-1) \\ [-\frac{\gamma e}{\rho} + (\gamma-1)(u^2+v^2)]v & (1-\gamma)uv & \frac{\gamma e}{\rho} - \frac{\gamma-1}{2}(u^2+3v^2) & \gamma v \end{pmatrix}$$

where ρ, p, and e are density, pressure, and total energy, respectively. The total energy is defined by

$$e = \frac{p}{\gamma-1} + \frac{\rho}{2}(u^2+v^2)$$

in which γ is the ratio of the specific heats. The eigenvalues of $\mathbf{A}_c$ and $\mathbf{B}_c$ are

$$\lambda_{1,2,3,4}(\mathbf{A}_c) = u, u, u+c, u-c$$

and

$$\lambda_{1,2,3,4}(\mathbf{B}_c) = v, v, v+c, v-c$$

where c is the sound speed.

In general coordinates, the vectors and matrices for the compressible system are given as

$$\hat{\mathbf{Q}}_c = \frac{1}{J}\begin{pmatrix} \rho \\ \rho u \\ \rho v \\ e \end{pmatrix}, \quad \hat{\mathbf{E}}_c = \frac{1}{J}\begin{pmatrix} \rho U \\ \rho u U + \xi_x p \\ \rho v U + \xi_y p \\ (e+p)U \end{pmatrix}, \quad \hat{\mathbf{H}}_c = \mathbf{0}$$

and

$$\hat{\mathbf{A}}_c =$$

$$\begin{pmatrix} 0 & \xi_x & \xi_y & 0 \\ -uU + \xi_x(\gamma-1)\alpha & U - (\gamma-2)\xi_x u & \xi_y u - (\gamma-1)\xi_x v & (\gamma-1)\xi_x \\ -vU + \xi_y(\gamma-1)\alpha & \xi_x v - (\gamma-1)\xi_y u & U - (\gamma-2)\xi_y v & (\gamma-1)\xi_y \\ [(\gamma-2)\alpha - \frac{c^2}{\gamma-1}]U & [\frac{c^2}{\gamma-1} + \alpha]\xi_x - (\gamma-1)uU & [\frac{c^2}{\gamma-1} + \alpha]\xi_y - (\gamma-1)vU & \gamma U \end{pmatrix}$$

where α is defined as

$$\alpha = \frac{1}{2}(u^2+v^2)$$

As before, $\hat{\mathbf{F}}_c$ and $\hat{\mathbf{B}}_c$ can be obtained by replacing ξ in $\hat{\mathbf{E}}_c$ and $\hat{\mathbf{A}}_c$ by η. The eigenvalues of $\hat{\mathbf{A}}_c$ and $\hat{\mathbf{B}}_c$ are,

$$\lambda_{1,2,3,4}(\hat{\mathbf{A}}_c) = U, U, U + C_\xi, U - C_\xi \tag{4.10a}$$

and

$$\lambda_{1,2,3,4}(\hat{\mathbf{B}}_c) = V, V, V + C_\eta, V - C_\eta \tag{4.10b}$$

where the contravariant velocity components, U and V, are defined in Eq. (4.7) and Eq. (4.8). C_ξ and C_η are the acoustic wave speeds in the ξ and η directions, respectively. They are given as

$$C_\xi = c\sqrt{\xi_x^2 + \xi_y^2}$$

and

$$C_\eta = c\sqrt{\eta_x^2 + \eta_y^2}$$

4.2 Implicit Scheme

The one-dimensional governing equation, Eq. (3.1), was discretized in delta-form as Eq. (3.6) in the previous chapter. The analogous discretized form of the two-dimensional equation (Eq. (4.1)) is

$$(\mathbf{I} - \Delta t\mathbf{D} + \Delta t\delta_\xi \mathbf{A}\cdot + \Delta t\delta_\eta \mathbf{B}\cdot)\,\Delta\mathbf{Q} = RHS \tag{4.11}$$

where RHS stands for the finite-difference form of the steady-state part of the governing equation, that is,

$$RHS = \Delta t\left(\mathbf{H}_{i,j}^n - \delta_\xi \mathbf{E}_{i,j}^n - \delta_\eta \mathbf{F}_{i,j}^n\right) \tag{4.12}$$

and δ_ξ and δ_η are central-difference operators in ξ- and η-directions, respectively. In Eq. (4.11) and Eq. (4.12), the 'hats' of $\hat{\mathbf{Q}}$, $\hat{\mathbf{E}}$, $\hat{\mathbf{F}}$, and $\hat{\mathbf{H}}$ are dropped for convenience.

The most direct analogy of a one-dimensional implicit scheme is to invert the left-hand-side matrix in Eq. (4.11) directly. In most practical cases, however, the left-hand-side matrix is too large to be inverted directly for two-dimensional problems. Direct inversion requires tremendous storage space as well as a huge number of operations. Therefore, it is more economical to replace the LHS by approximate-factorization [17,38] such as

$$(I - \Delta t\mathbf{D} + \Delta t\delta_\xi \mathbf{A})\,(I - \Delta t\mathbf{D})^{-1}\,(I - \Delta t\mathbf{D} + \Delta t\delta_\eta \mathbf{B})\,\Delta\mathbf{Q} = RHS \qquad (4.13)$$

This factorization allows us to solve a two-dimensional problem by solving one-dimensional operators.

Equation (4.13) has three operators on the LHS. The first and the third operators are block-tridiagonal inversions. The second is simply a matrix-vector multiplication. For cases without the source term, i.e., $\mathbf{H} = 0$, the second operator is an identity matrix and may be dropped out. The ξ- and η-sweep are implemented in a line-by-line pattern. Since the inversion of each line is isolated in one time step, all the ξ- or η-lines can be parallel-processed. This makes approximate-factorization vectorizable on a vector machine.

4.2.1 Artificial Dissipation and Variable Δt

In order to damp out high-wavenumbers, fourth-order explicit and second-order implicit dissipation terms are added to Eq. (4.13), as suggested by Steger [39].

$$\left(I - \Delta t\mathbf{D} + \Delta t\delta_\xi \mathbf{A}\cdot - \frac{\epsilon_i}{8}\frac{1}{J}\delta_{\xi\xi} J\cdot\right)(I - \Delta t\mathbf{D})^{-1}$$

$$\cdot\left(I - \Delta t\mathbf{D} + \Delta t\delta_\eta \mathbf{B}\cdot - \frac{\epsilon_i}{8}\frac{1}{J}\delta_{\eta\eta} J\cdot\right)\Delta\mathbf{Q}$$

$$= RHS - \frac{\epsilon_e}{8}\frac{1}{J}(\delta_{\xi\xi\xi\xi} + \delta_{\eta\eta\eta\eta})J\mathbf{Q}^n \qquad (4.14)$$

where

$$\delta_{\xi\xi} J\Delta \mathbf{Q} = (J\mathbf{Q})_{i+1,j} - 2(J\mathbf{Q})_{i,j} + (J\mathbf{Q})_{i-1,j} \tag{4.15}$$

$$\delta_{\xi\xi\xi\xi} J\mathbf{Q} = (J\mathbf{Q})_{i+2,j} - 4(J\mathbf{Q})_{i+1,j} + 6(J\mathbf{Q})_{i,j} - 4(J\mathbf{Q})_{i-1,j} + (J\mathbf{Q})_{i-2,j} \tag{4.16}$$

The second-order implicit dissipation terms do not break down the block-tridiagonal structure because they involve only three points in each direction. They have no effect on the converged steady-state solution as they are added to the left-hand-side. The fourth-order explicit dissipation, on the other hand, does modify the steady state solution. However, since it is of fourth-order, it affects the governing equation by only $O(\Delta x^3, \Delta y^3)$, assuming that the CFL numbers are kept 'constant' as will be shown later on in this section (see also Section 2.2). The values of ϵ_i and ϵ_e can be determined by the stability analysis that follows. The results will suggest the following relations for maximum dissipation of the highest-wavenumber waves:

$$\epsilon_e = \left(1 + \frac{\epsilon_i}{2}\right)^2 /4 \qquad \text{for 2-D} \tag{4.17}$$

Equation (4.17) indicates that the implicit viscosity plays the role of relieving the restriction on the fourth order explicit viscosity. For some severe cases, this can be one way to coerce this scheme to converge. However, it is usually unfavorable to apply large values of explicit viscosity, since it introduces errors.

In Chapter 3, the strategy of using constant CFL for one-dimensional problems have been shown to be effective and stable. In two dimensions, we wish to use the same concept to push the Δt of each grid point to approximately the same convergence rate and accelerate the convergence. Consequently, we estimate the 'effective' CFL number as

$$CFL = \Delta t \sqrt{\lambda_{\xi_{max}}^2 + \lambda_{\eta_{max}}^2} \tag{4.18}$$

where $\lambda_{\xi_{max}}$ and $\lambda_{\eta_{max}}$ are the maximum eigenvalues in ξ- and η-directions, respectively. The time-step size is then determined by using Eq. (4.18) while keeping

CFL constant. The eigenvalues for both incompressible and compressible systems have been shown in Eq. (4.9) and Eq. (4.10). They are functions of the flow variables, and must be updated for each iteration.

4.2.2 Implicit Boundary Conditions

In this section, the MOC boundary conditions that were used in one dimension are extended to two dimensions. For a one-dimensional problem, a matrix $\mathbf{M}^{-1}$ is premultiplied to the hyperbolic system under consideration such that the system is implicitly decoupled into several scalar hyperbolic equations. A selection matrix $\mathbf{L}$ is then premultiplied to the resulting uncoupled system to extract the outgoing information. As for the incoming information, specified boundary conditions are imposed. A procedure that is analogous to that for a one-dimensional problem is applied to a two-dimensional problem.

Consider a source-free two-dimensional system of equations. If one is allowed to use the full implicit scheme, the MOC boundary formulation on a constant-ξ boundary line should be as follows in order to extract the appropriate information from the computational domain:

$$\mathbf{L}\mathbf{M}_{\xi}^{-1}\left[\left(\mathbf{I}+\Delta t\delta_{\xi}\mathbf{A}\cdot+\Delta t\delta_{\eta}\mathbf{B}\cdot\right)\Delta\mathbf{Q}=RHS\right] \tag{4.19}$$

where $\mathbf{L}$, as explained in Section 3.2.2, is the selection matrix. Equation (4.19) represents the finite-difference form of the outgoing characteristics. Specified boundary conditions written in a delta-form (refer to Eq. (3.12)) are then added to Eq. (4.19) to form the complete equations for boundary points as

$$\left(\mathbf{S}+\Delta t\mathbf{L}\mathbf{M}_{\xi}^{-1}\delta_{\xi}\mathbf{A}\cdot+\Delta t\mathbf{L}\mathbf{M}_{\xi}^{-1}\delta_{\eta}\mathbf{B}\cdot\right)\Delta\mathbf{Q}=\mathbf{L}\mathbf{M}_{\xi}^{-1}RHS \tag{4.20}$$

where

$$\mathbf{S}=\mathbf{L}\mathbf{M}_{\xi}^{-1}+\frac{\partial\mathbf{\Omega}}{\partial\mathbf{Q}} \tag{4.21}$$

Equation (4.20) is the same as the one-dimensional boundary formulation except for the η-operators. The left side of Eq. (4.20) cannot be inverted directly for the same reason as the interior grid points. One could find several possible ways to approximately factorize Eq. (4.20). Chakravarthy [29] premultiplied Eq. (4.20) by $\mathbf{S}^{-1}$ and approximately factorized the left-hand-side of the resulting equation to get

$$(\mathbf{I}+\Delta t\mathbf{S}^{-1}\mathbf{L}\mathbf{M}_{\xi}^{-1}\frac{\partial \mathbf{A}\cdot}{\partial \xi})(\mathbf{I}+\Delta t\mathbf{S}^{-1}\mathbf{L}\mathbf{M}_{\xi}^{-1}\frac{\partial \mathbf{B}\cdot}{\partial \eta})\Delta\mathbf{Q}=\mathbf{S}^{-1}\mathbf{L}\mathbf{M}_{\xi}^{-1}RHS$$

or

$$(\mathbf{S}+\Delta t\mathbf{L}\mathbf{M}_{\xi}^{-1}\frac{\partial \mathbf{A}\cdot}{\partial \xi})\mathbf{S}^{-1}(\mathbf{S}+\Delta t\mathbf{L}\mathbf{M}_{\xi}^{-1}\frac{\partial \mathbf{B}\cdot}{\partial \eta})\Delta\mathbf{Q}=\mathbf{L}\mathbf{M}_{\xi}^{-1}RHS \qquad (4.22)$$

Equation (4.22) requires the calculation of modified inversion operators in both the ξ- and η-directions at the boundaries, and is less convenient than what Rai and Chaussee [28] proposed. They suggested a procedure which carries out the boundary-line direction inversion first, and then premultiplies the resulting equation by $\mathbf{L}\mathbf{M}_{\xi}^{-1}$:

$$(\mathbf{S}+\Delta t\mathbf{L}\mathbf{M}_{\xi}^{-1}\frac{\partial \mathbf{A}\cdot}{\partial \xi})\Delta\mathbf{Q}=\mathbf{L}\mathbf{M}_{\xi}^{-1}(\mathbf{I}+\Delta t\frac{\partial \mathbf{B}\cdot}{\partial \eta})^{-1}RHS \qquad (4.23)$$

The procedure thus formed is more convenient to program and to vectorize for it avoids the need to modify the inversion in the η-direction when a constant-ξ boundary line is considered, and is more economical because it saves one matrix-vector multiplication as appears in Eq. (4.22). Hence, the present work adopts Eq. (4.23) as the boundary procedure.

Equation (4.23) is not strictly characteristic in nature in the sense of Eq. (4.19), because the multiplication by $\mathbf{L}\mathbf{M}_{\xi}^{-1}$ is carried out after the first inversion. The error that is introduced by using Eq. (4.23) instead of Eq. (4.19) is shown by Rai and Chaussee to be of the same order as the error introduced by approximate-factorization [28]. This indicates that this boundary procedure should not contaminate the transient accuracy or the stability inherent to the ADI scheme.

To implement the MOC boundary conditions, inverses of the eigenmatrices, $\mathbf{M}_{\xi}^{-1}$ and $\mathbf{M}_{\eta}^{-1}$, must be derived. The complexity of compressible flow flux Jacobians makes the derivation of its eigenmatrices a laborious task. The derivation can be simplified by transforming the conservative system into a non-conservative form first, and then deriving the eigenmatrices for the non-conservative system. The algebra of deriving the eigenmatrices of the non-conservative system is noticeably easier. The eigenmatrices for the conservative system can thus be obtained via an inverse transformation. The details have been addressed by Beam and Warming [38].

For two-dimensional compressible flow, the inverse of the eigenmatrix, $\mathbf{M}_{\xi c}^{-1}$, is

$$\mathbf{M}_{\xi c}^{-1} = \begin{pmatrix} 1-\frac{\gamma-1}{2}M^2 & (\gamma-1)\frac{u}{c^2} & (\gamma-1)\frac{v}{c^2} & -(\gamma-1)\frac{1}{c^2} \\ \xi_x v - \xi_y u & \xi_y & -\xi_x & 0 \\ \frac{\gamma-1}{2}M^2 - \frac{U}{C_\xi} & \frac{\xi_x}{C_\xi} - (\gamma-1)\frac{u}{c^2} & \frac{\xi_y}{C_\xi} - (\gamma-1)\frac{v}{c^2} & \frac{\gamma-1}{c^2} \\ \frac{\gamma-1}{2}M^2 + \frac{U}{C_\xi} & -\frac{\xi_x}{C_\xi} - (\gamma-1)\frac{u}{c^2} & -\frac{\xi_y}{C_\xi} - (\gamma-1)\frac{v}{c^2} & \frac{\gamma-1}{c^2} \end{pmatrix} \tag{4.24}$$

where M is the Mach number, U is the contravariant velocity in ξ direction, as defined in Eq. (4.7), and C_ξ is defined as

$$C_\xi \equiv c\sqrt{\xi_x^2 + \xi_y^2}$$

which is the compressible-flow acoustic speed in ξ-direction of the computational domain. A similarity transformation by $\mathbf{M}_{\xi c}^{-1}$ and $\mathbf{M}_{\xi c}$ transforms the flux Jacobian $\mathbf{A}_c$ into a diagonal matrix:

$$\mathbf{M}_{\xi c}^{-1}\mathbf{A}_c\mathbf{M}_{\xi c} = \begin{pmatrix} U & 0 & 0 & 0 \\ 0 & U & 0 & 0 \\ 0 & 0 & U+C_\xi & 0 \\ 0 & 0 & 0 & U-C_\xi \end{pmatrix}$$

As before, $\mathbf{M}_{\eta c}^{-1}$ can obtained by changing ξ in Eq. (4.24) to η.

The specification of boundary conditions in compressible flow calculations depends upon whether the flow is subsonic or supersonic, inflow or outflow. Stagnation

pressure, stagnation temperature, and flow angle are usually specified at an subsonic inflow boundary, while the back pressure can be specified for subsonic outflow. For supersonic inflow, Mach number can be imposed in addition to the three conditions mentioned above. For a supersonic outflow, no condition is specified. On a non-penetrable wall surface, only the tangency condition needs to be imposed.

For an incompressible flow, $\mathbf{M}_{\xi i}^{-1}$ is given as

$$\mathbf{M}_{\xi i}^{-1} = \begin{pmatrix} v\xi_x - u\xi_y & vU + \xi_y\beta & -uU - \xi_y\beta \\ -m_\xi & -\xi_x(U + C_\xi) & -\xi_y(U + C_\xi) \\ m_\xi & \xi_x(U - C_\xi) & \xi_y(U + C_\xi) \end{pmatrix} \tag{4.25}$$

where

$$m_\xi = \xi_x^2 + \xi_y^2$$

and

$$C_\xi = \sqrt{U^2 + \beta(\xi_x^2 + \xi_y^2)}$$

This matrix defines the following similarity transformation:

$$\mathbf{M}_{\xi i}^{-1}\mathbf{A}_i\mathbf{M}_{\xi i} = \begin{pmatrix} U & 0 & 0 \\ 0 & U + C_\xi & 0 \\ 0 & 0 & U - C_\xi \end{pmatrix}$$

As mentioned earlier, this incompressible hyperbolic system is always subsonic. The number of boundary conditions at an inflow boundary is two. Stagnation pressure and flow angle are suitable for inflow conditions, and pressure can be used as the outflow condition. At a solid wall surface, the tangency condition is applied.

4.2.3 Explicit Boundary Conditions

The implicit treatment of boundary conditions based on the MOC as addressed previously is stable and transiently consistent with the field points. This method is used for most cases in this work. However, cases are encountered when it is more convenient to treat boundary conditions separately from the field points. This

procedure is generally called the explicit boundary condition procedure [25]. For instance, in the diagonal algorithm [25,40] which will be presented later on, blocks of the left-hand-side matrices must be diagonal. Unfortunately, MOC boundary conditions can destroy the diagonal structure. On such occasion explicit boundary conditions can be invoked.

By definition, the explicit procedure is applied by updating field points and boundary points separately and in sequence. More specifically, ΔQ is set to be zero for all boundary points during the time-integration process, and then the properties at the boundary points are updated after the field points. In contrast, the MOC boundary-condition procedure couples field points and boundary points such that simultaneous updating results.

Several advantages and disadvantages are observed in using the explicit boundary procedure. It has the unavoidable disadvantage of lagging the properties of the boundary points during the time-evolution. As a result, residual waves may reflect on the boundaries. Moreover, the lagging contaminates the time-accuracy of a scheme, and may not be suitable for transient calculations. Nevertheless, it does have the advantage of making programming easier because all boundary conditions can be isolated in one subroutine, and can be modified without knowledge of the time-marching procedures being used. It also makes the left-hand-side blocks at the boundaries of an implicit scheme diagonal. As stated earlier, this makes it easier to be used for a diagonal algorithm.

The explicit boundary procedure has been employed by many widely used programs. Examples are ARC2D (uses ADI scheme) by Steger and Pulliam [25] and FLO52 (uses Runge-Kutta explicit scheme) by Jameson [34]. As an example of the explicit boundary procedures, the set of compressible-flow boundary conditions used by Steger and Pulliam [25] in ARC2D is described below. Jameson applies similar

boundary conditions to the Runge-Kutta scheme. Analogous boundary conditions for incompressible flows are also given. Note that the following sets of boundary conditions are merely examples. There are other plausible sets of explicit boundary conditions.

4.2.3.1 Boundary Conditions for Compressible Flows

As before, at a rigid body surface tangency must be satisfied for inviscid flows. In other words, the velocity normal to solid surfaces must be zero. If a body-fitted grid system which maps a body surface to η = constant is used, the contravariant velocity component, $V = u\eta_x + v\eta_y$, is proportional to the normal velocity. Thus

$$u\eta_x + v\eta_y = 0 \tag{4.26}$$

is a required boundary condition. In order to separate the boundary conditions from the field points calculation, Eq. (4.26) cannot be used as it was in MOC procedure. Hence we must identify substitutional boundary conditions. A second condition that is used by Pulliam [25] is extrapolating the tangential contravariant velocity component from the field points as

$$(u\xi_x + v\xi_y)_1 = U_1 = 2U_2 - U_3 \tag{4.27}$$

where the subscripts indicate node numbers in the η-direction, and where '1' is on the wall. Equations (4.26) and (4.27) form a 2×2 linear system for u and v. Inverting the system gives

$$\begin{pmatrix} u \\ v \end{pmatrix} = \frac{1}{\xi_x\eta_y - \xi_y\eta_x} \begin{pmatrix} \eta_y U_1 \\ -\eta_x U_1 \end{pmatrix} \tag{4.28}$$

Steger and Pulliam obtained the pressure on the body surface from the normal momentum equation

$$(\xi_x\eta_x + \xi_y\eta_y)p_\xi + (\eta_x^2 + \eta_y^2)p_\eta + \rho U(\eta_x u_\xi + \eta_y v_\xi) = 0 \tag{4.29}$$

Equation (4.29) can be derived as follows. Starting with the non-conservative form of the momentum equations,

$$\rho\frac{\partial u}{\partial t}+\rho u\frac{\partial u}{\partial x}+\rho v\frac{\partial u}{\partial y}=-\frac{\partial p}{\partial x} \tag{4.30a}$$

$$\rho\frac{\partial v}{\partial t}+\rho u\frac{\partial v}{\partial x}+\rho v\frac{\partial v}{\partial y}=-\frac{\partial p}{\partial y} \tag{4.30b}$$

Multiply Eq. (4.30a) and Eq. (4.30b) by η_x and η_y respectively, apply the chain rule to obtain derivatives of ξ and η, and then add the two equations to get

$$\rho\frac{\partial V}{\partial t}+\rho U(\eta_x u_\xi+\eta_y v_\xi)+\rho V(\eta_x u_\eta+\eta_y v_\eta)+(\xi_x\eta_x+\xi_y\eta_y)p_\xi+(\eta_x^2+\eta_y^2)p_\eta=0 \tag{4.31}$$

On a solid surface, V vanishes and Eq. (4.29) is obtained.

Steger and Pulliam solved Eq. (4.29) at the surface by using central differencing in the ξ-direction and one-sided first- or second-order accurate differencing in the η-direction. For steady uniform incoming flow, they apply the free-stream stagnation enthalpy $H_0=(e+p)/\rho$ along the body in an inviscid flow. Using the equation for enthalpy, and the computed velocity components and pressure, a value of density can be obtained at the body surface.

At inflow and outflow boundaries, Steger and Pulliam decompose the velocity into normal and tangential components. The locally one-dimensional Riemann invariants are then given in terms of the normal velocity component, u_n, as

$$R_1=u_n-\frac{2c}{\gamma-1}$$

and

$$R_2=u_n+\frac{2c}{\gamma-1}$$

where the normal vector is pointed outward from the domain. The Riemann invariants R_1 and R_2 are associated with the two characteristic velocity $\lambda_1=u_n-c$ and $\lambda_2=u_n+c$ respectively. When $|u_n|<c$, λ_1 and λ_2 have opposite signs. The

Riemann invariant corresponding to the incoming characteristics is taken from the exterior of the domain, and the other Riemann invariant is extrapolated from the field points. Once the Riemann invariants of boundary points are known, u_n and c are determined. Two other quantities are needed for the remaining two characteristics associated with $\lambda_{3,4} = u_n, u_n$. They choose the tangential velocity component u_t and entropy $s = \ln(p/\rho^\gamma)$. When u_n is positive, the flow is outward, and u_t and s are extrapolated from the interior points. When u_n is negative, properties outside the computational domain are applied. If $|u_n| > c$, the flow is supersonic, and all four quantities are either extrapolated or taken from flow outside the domain, depending on the sign of u_n.

4.2.3.2 Boundary Conditions for Incompressible Flows

The boundary conditions for incompressible flows can be applied in a fashion analogous to compressible flows. At body surfaces, Eq. (4.26), Eq. (4.27), Eq. (4.28), and Eq. (4.29) are still applicable, and the procedure of applying them on compressible flows holds for incompressible flows. Since the energy equation is absent for the incompressible Euler equations, the density calculation on the body surfaces is not required for incompressible flows.

On the inflow and outflow boundaries, we can make the locally one-dimensional time-evolution assumption just as Steger and Pulliam did for compressible flows. However, the invariants for incompressible flows are different from those for compressible flows. We now derive them. Consider the non-conservative equations for a transient one-dimensional incompressible flow, which is allowed to be 'compressible' by the concept of pseudocompressibility.

$$\frac{\partial p}{\partial t} + \beta \frac{\partial u}{\partial x} = 0 \tag{4.32a}$$

$$\frac{\partial u}{\partial t} + 2u\frac{\partial u}{\partial x} + \frac{\partial p}{\partial x} = 0 \tag{4.32b}$$

Multiplying Eq. (4.32b) by $u+c$, where $c=\sqrt{u^2+\beta}$ is the pseudoaccoustic speed (see also Sec. 3.1), and adding it to Eq. (4.32a) give

$$\left[\frac{\partial p}{\partial t}+(u+c)\frac{\partial p}{\partial x}\right]+(u+c)\left[\frac{\partial u}{\partial t}+(u+c)\frac{\partial u}{\partial x}\right]=0 \qquad (4.33a)$$

Similarly, multiplying Eq. (4.32b) by $u-c$ and adding it to Eq. (4.32a) give

$$\left[\frac{\partial p}{\partial t}+(u-c)\frac{\partial p}{\partial x}\right]+(u-c)\left[\frac{\partial u}{\partial t}+(u-c)\frac{\partial u}{\partial x}\right]=0 \qquad (4.33b)$$

Equation (4.33b) is the characteristic equation along the path

$$dx=(u+c)dt$$

Thus we have

$$\frac{du}{dt}_{u+c}=\frac{\partial u}{\partial t}+(u+c)\frac{\partial u}{\partial x}$$

and

$$\frac{dp}{dt}_{u+c}=\frac{\partial p}{\partial t}+(u+c)\frac{\partial p}{\partial x}$$

Equation (4.33a) then becomes

$$dp+(u+c)du=0 \qquad (4.34)$$

By integrating Eq. (4.34), one gets the first invariant

$$p+\frac{1}{2}u(u+c)+\frac{\beta}{2}\ln(u+c)=\text{constant} \qquad \text{along} \qquad dx=(u+c)dt \qquad (4.35a)$$

The second invariant is derived from Eq. (4.33b) in a similar fashion. The result is

$$p+\frac{1}{2}u(u-c)+\frac{\beta}{2}\ln(c-u)=\text{constant} \qquad \text{along} \qquad dx=(u-c)dt \qquad (4.35b)$$

For incompressible flows with pseudocompressibility, the pseudoacoustic speed c is always greater than u. Hence one of the invariants must be extrapolated and the

other replaced by a boundary condition. For an inflow boundary, stagnation pressure and u_t from the exterior of the domain along with one invariant extrapolated from the interior, determine the boundary quantities. For an outflow boundary, one extrapolated invariant, extrapolated u_t, and the specified pressure are sufficient boundary conditions for the explicit procedure.

4.2.4 Stability Analysis

The two-dimensional linear stability analysis is based on the two-dimensional Fourier transformation. Similar to one-dimensional waves, two-dimensional waves can be considered as a combination of a finite number of wave modes:

$$Q^n_{j,k} = \sum_{-\pi/\Delta x}^{\pi/\Delta x} \sum_{-\pi/\Delta y}^{\pi/\Delta y} \bar{Q}^n(\omega_x, \omega_y) e^{i(\omega_x j \Delta x + \omega_y k \Delta y)} \tag{4.36}$$

where $\bar{Q}^n$ is the vector Fourier coefficient, ω_x and ω_y are the wavenumbers and $i = \sqrt{-1}$. The amplification matrix G is defined as

$$Q^{n+1} = G(\omega_x \Delta x, \omega_y \Delta y) Q^n$$

and is determined by inserting the Fourier expansion for Q^n and Q^{n+1} into the specific finite-difference equation. A necessary condition for stability is that all eigenvalues of matrix G have magnitudes less than one.

A vector stability analysis, rather than a scalar analysis, is the most precise and accurate method for predicting the stability of a given scheme. For the Euler equations, however, it is difficult to obtain a closed form of the amplification factor. Practically, the vector stability analysis is done numerically for particular cases. This makes it difficult to draw a clear-cut conclusion concerning the stability of a scheme. On the other hand, a scalar stability analysis is an easier problem to consider, and provides an explicit form of the amplification factor in most occasions. By studying the amplification factor of a scalar analysis, some preliminary

conclusions may be drawn concerning the effect of some parameters such as the CFL number and the viscosities.

Consider the following scalar equation in a two-dimensional space:

$$\frac{\partial u}{\partial t} + \lambda_\xi \frac{\partial u}{\partial \xi} + \lambda_\eta \frac{\partial u}{\partial \eta} = 0 \tag{4.37}$$

If Eq. (4.37) is to be solved using the full implicit scheme with artificial viscosity added, the finite-difference equation is written as

$$\begin{aligned} &\left(1 + \Delta t \left(\lambda_\xi \delta_\xi + \lambda_\eta \delta_\eta\right) - \frac{\epsilon_i}{8}\left(\delta_{\xi\xi} + \delta_{\eta\eta}\right)\right) \Delta u \\ &= -\Delta t \left(\lambda_\xi \delta_\xi + \lambda_\eta \delta_\eta\right) u^n - \frac{\epsilon_e}{8}\left(\delta_{\xi\xi\xi\xi} + \delta_{\eta\eta\eta\eta}\right) u^n \end{aligned} \tag{4.38}$$

When approximate-factorization is applied, the finite-difference equation becomes

$$\begin{aligned} &\left(1 + \Delta t \lambda_\xi \delta_\xi - \frac{\epsilon_i}{8}\delta_{\xi\xi}\right)\left(1 + \Delta t \lambda_\eta \delta_\eta - \frac{\epsilon_i}{8}\delta_{\eta\eta}\right) \Delta u \\ &= -\Delta t \left(\lambda_\xi \delta_\xi + \lambda_\eta \delta_\eta\right) u^n - \frac{\epsilon_e}{8}\left(\delta_{\xi\xi\xi\xi} + \delta_{\eta\eta\eta\eta}\right) u^n \end{aligned} \tag{4.39}$$

Applying Von Neumann stability analysis gives the amplification factor in the absence of approximate-factorization (Eq. (4.38)):

$$g_{\text{implicit}} = \frac{1 + \frac{\epsilon_i}{4}(2 - C_\xi - C_\eta) - \frac{\epsilon_e}{2}\left((1 - C_\xi)^2 + (1 - C_\eta)^2\right)}{1 + \frac{\epsilon_i}{4}(2 - C_\xi - C_\eta) + i(CFL_\xi S_\xi + CFL_\eta S_\eta)} \tag{4.40}$$

The amplification factor for the ADI scheme (Eq. (4.39)) is the same as that for the full implicit scheme (Eq. (4.40)) except for the addition of a term to the numerator and the denominator:

$$g_{\text{ADI}} = \frac{1 + \frac{\epsilon_i}{4}(2 - C_\xi + C_\eta) - \frac{\epsilon_e}{2}\left((1 - C_\xi)^2 + (1 - C_\eta)^2\right) + E}{1 + \frac{\epsilon_i}{4}(2 - C_\xi + C_\eta) + i(CFL_\xi S_\xi + CFL_\eta S_\eta) + E} \tag{4.41}$$

where as before, C_ξ, C_η, S_ξ, and S_η are abbreviations for trigonometric functions $\cos(\omega_\xi \Delta\xi)$, $\cos(\omega_\eta \Delta\eta)$, $\sin(\omega_\xi \Delta\xi)$, and $\sin(\omega_\eta \Delta\eta)$. The additional term, E, corresponds to the temporal error introduced by approximate-factorization,

$$E = \left(\frac{\epsilon_i}{4}(1 - C_\xi) + iCFL_\xi S_\xi\right)\left(\frac{\epsilon_i}{4}(1 - C_\eta) + iCFL_\eta S_\eta\right) \tag{4.42}$$

As an approximation, consider the symmetric analysis, namely,

$$\lambda_\xi = \lambda_\eta$$

According to Eq. (4.18), CFL_ξ and CFL_η can be expressed in terms of CFL as

$$CFL_xi = \Delta t \lambda_\xi^2 = CFL/\sqrt{2}$$

and

$$CFL_eta = \Delta t \lambda_\eta^2 = CFL/\sqrt{2}$$

To simplify the analysis, we furthur assume

$$\omega_\xi \Delta\xi = \omega_\eta \Delta\eta = \theta$$

This approximation simplifies the stability analysis significantly, and in many cases this includes the most unstable mode. In fact, when the CFL number and the artificial viscosities in one direction are set to be zeroes, one-dimensional analysis can be recovered from either full implicit or ADI schemes. The symmetric analysis is a case which is midway between two one-dimensional limits of the stability analysis, and is likely to result in a more restrictive, and therefore a safer, stability criterion.

First we consider the case without artificial viscous terms. The absolute values of the amplification factors are

$$|g_{\text{implicit}}| = \frac{1}{\sqrt{1 + 2CFL^2S^2}} \tag{4.43}$$

for the full implicit scheme and

$$|g_{\text{ADI}}| = \frac{|1 - CFL^2S^2/2|}{\sqrt{(1 - CFL^2S^2/2)^2 + 2CFL^2S^2}} \tag{4.44}$$

for the ADI scheme. Representative curves of Eq. (4.43) and Eq. (4.44) as functions of wave modes for various CFL are shown in Fig. 17 and Fig. 18.

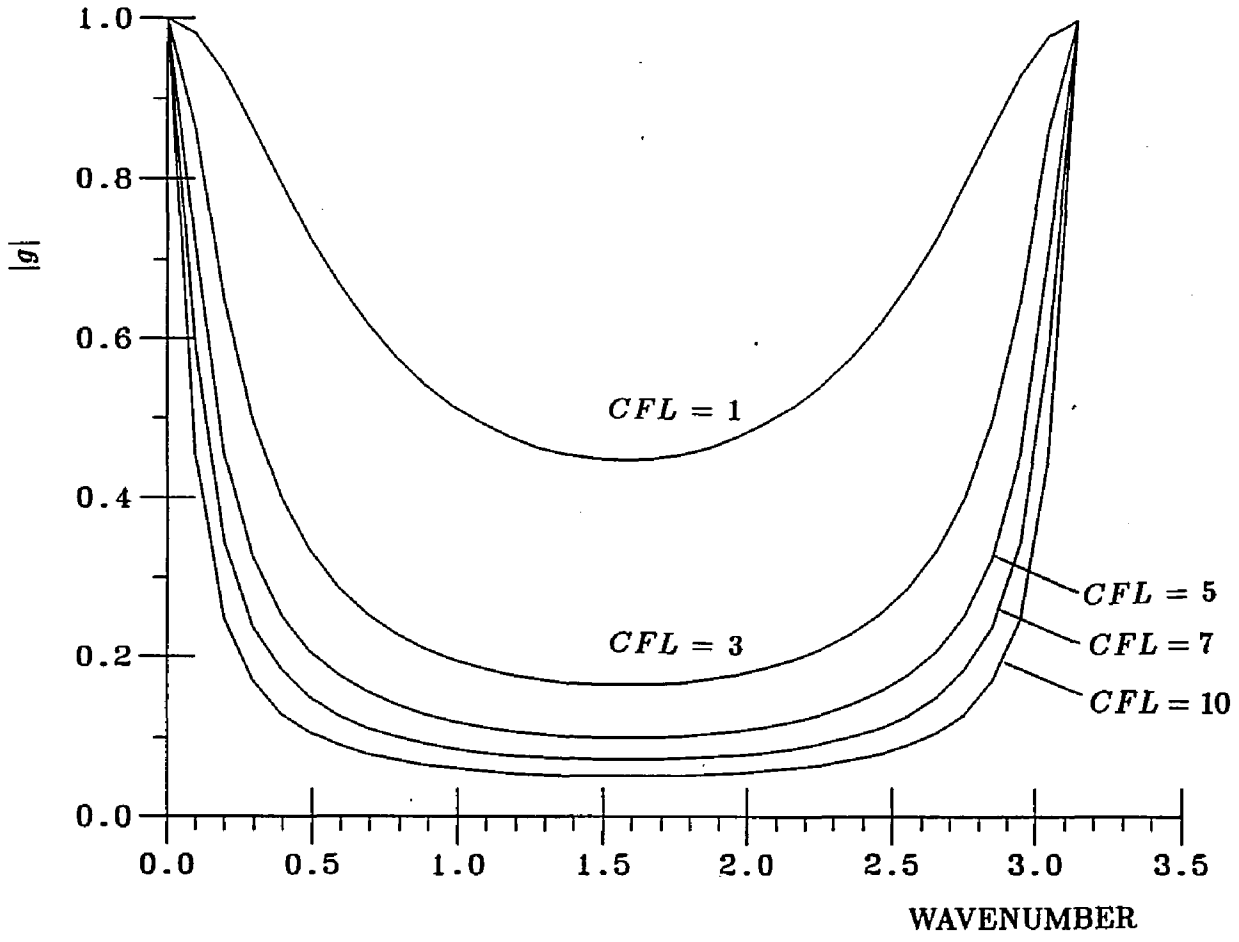

Figure 17: Amplification Factor of the Full Implicit Scheme Applied to a Two-Dimensional Scalar Hyperbolic Equation. No Viscosity Added.

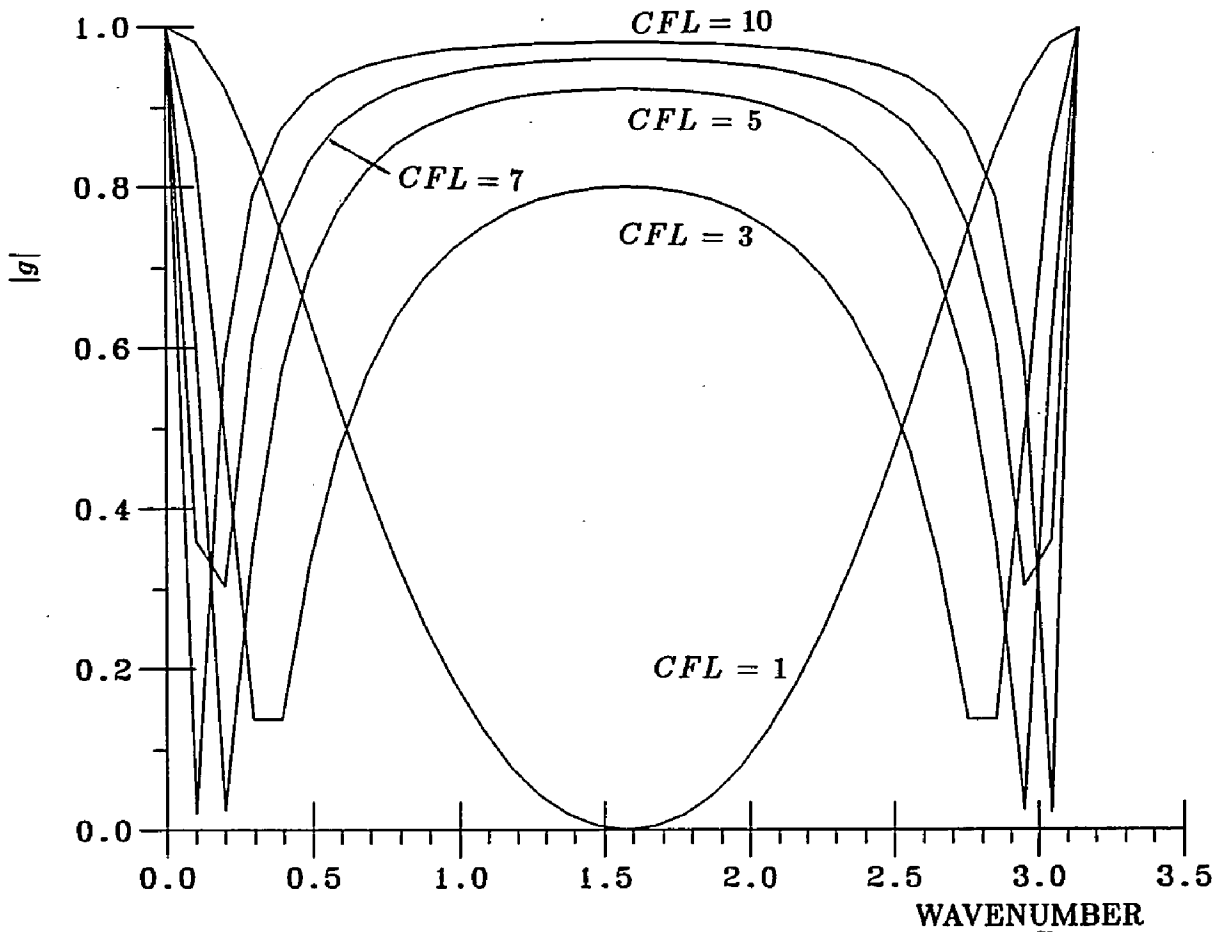

Figure 18: Amplification Factor of the Implicit ADI Scheme Applied to a Two-Dimensional Scalar Hyperbolic Equation. No Viscosity Added.

By observing Eq. (4.43) and Eq. (4.44), one finds that the implicit scheme is unconditionally stable, with or without approximate-factorization. However, Eq. (4.43) has the feature of increasing stability with increasing CFL, while Eq. (4.44) approaches neutral stability as CFL increases. This can be understood by considering the wave mode $(\frac{\pi}{2}, \frac{\pi}{2})$. At this mode, Eq. (4.43) and Eq. (4.44) become

$$|g_{\text{implicit}}| = \frac{1}{\sqrt{1+2CFL^2}} \tag{4.45}$$

and

$$|g_{\text{ADI}}| = \frac{|1-CFL^2/2|}{1+CFL^2/2} \tag{4.46}$$

The right-hand-side of Eq. (4.45) approaches zero asymptotically and monotonically with increasing CFL, while that of Eq. (4.46) reduces to zero as $CFL = \sqrt{2}$, then approaches unity asymptotically as CFL approaches infinity. This indicates that the convergence rate of the ADI scheme does not always increase with the CFL number it uses. In other words, there must be an optimum CFL number with which the ADI scheme enjoys its maximum convergence rate.

Next consider the effect of the artificial viscous terms. With the symmetric assumption, Eq. (4.40) and Eq. (4.41) become

$$g_{\text{implicit}} = \frac{1+\frac{\epsilon_i}{2}(1-C)-\epsilon_e(1-C)^2}{1+\frac{\epsilon_i}{2}(1-C)+i\sqrt{2}CFLS} \tag{4.47}$$

and

$$g_{\text{ADI}} = \frac{1+\frac{\epsilon_i}{2}(1-C)-\epsilon_e(1-C)^2+E}{1+\frac{\epsilon_i}{2}(1-C)+i\sqrt{2}CFLS+E} \tag{4.48}$$

where

$$E = \left(\frac{\epsilon_i}{4}(1-C)+iCFLS/2\right)^2 \tag{4.49}$$

The amplification factor of the full implicit scheme (Eq. (4.47)) is the same as the one-dimensional result except that CFL is multiplied by $\sqrt{2}$, and ϵ_i, and ϵ_e are multiplied by 2. As a contrast, g_{ADI} does not represent any one-dimensional

result because of the presence of the term, E, which is introduced by approximate-factorization.

The artificial viscosity terms are most effective on the highest-wavenumber disturbance, as addressed earlier. The dissipation of the highest-wavenumber mode is mostly controlled by the artificial viscosities. The amount of viscosity added should be determined by considering the highest-wavenumber. At the highest wavenumber, Eq. (4.47) and Eq. (4.48) reduce to

$$|g_{\text{implicit}}| = \frac{|1 + \epsilon_i - 4\epsilon_e|}{1 + \epsilon_i} \tag{4.50}$$

and

$$|g_{\text{ADI}}| = \frac{\left|\left(1 + \frac{\epsilon_i}{2}\right)^2 - 2\epsilon_e\right|}{\left(1 + \frac{\epsilon_i}{2}\right)^2} \tag{4.51}$$

Therefore, the strongest dissipation of the highest wavenumber can be obtained by adding the artificial viscosities according to

$$1 + \epsilon_i - 4\epsilon_e = 0 \qquad \text{full implicit scheme}$$

$$\left(1 + \frac{\epsilon_i}{2}\right)^2 - 2\epsilon_e = 0 \qquad \text{ADI scheme}$$

4.3 Explicit Scheme

We have just shown that the implicit schemes are unconditionally stable, with or without approximate-factorization. However, the second-order temporal error term introduced by approximate-factorization reduces the optimal CFL number as well as the convergence rate. Unfortunately, approximate-factorization is usually required because of the capability of current computers. For this reason, the explicit scheme may be of interest. Here we also consider the explicit scheme.

The Runge-Kutta explicit scheme can be extended from one dimension to two dimensions with no difficulty. The time-stepping procedures for one dimension (Eq. (3.22) and Eq. (3.25)) are also suitable for two dimensions. The only difference is

that two spatial operators must be evaluated in two dimensions. As noted earlier, a strict 'constant CFL' condition is difficult to impose for a two-dimensional implicit scheme. This difficulty also occurs in the explicit scheme.

For the explicit scheme, the boundary conditions may again be applied by considering characteristics. Compared to the two-dimensional implicit ADI scheme, the two-dimensional explicit scheme is relatively easy to adopt MOC boundary conditions. By premultiplying the governing equations with the inverse of the eigenmatrix and a selection matrix, one can select the characteristics outgoing from interior of the domain.

$$\mathbf{LM^{-1}}(\Delta\mathbf{Q} = RHS)$$

Proper boundary conditions as addressed in Section 4.2.2 are then added to result in

$$(\mathbf{S} + \mathbf{LM^{-1}})\Delta\mathbf{Q} = \mathbf{LM^{-1}}RHS \tag{4.52}$$

where the matrix S is defined previously by Eq. (4.21) for the implicit scheme. Note that we can also obtain Eq. (4.52) by dropping flux Jacobian terms in the boundary-condition formulations for the implicit schemes (Eq. (4.20) or Eq. (4.23)). In order to maintain stability, this boundary procedure for the explicit scheme is performed for each stage of the Runge-Kutta integration. The spatial operator associated with the direction normal to the boundary must again be one-sided differenced. Three-point differencing is taken to retain the second-order accuracy.

The two-dimensional scalar stability analysis of Runge-Kutta scheme can make use of one-dimensional results (given in Section 3.3.3). Once again, we consider the two-dimensional scalar model equation given by Eq. (4.37). With the artificial viscosity added, the amplification factor for each Runge-Kutta stage, ΔtP, is given by

$$Z = \frac{\epsilon_{ex}^{(4)}}{2}(1 - C_x)^2 + \frac{\epsilon_{ey}^{(4)}}{2}(1 - C_y)^2 + i(CFL_xS_x + CFL_yS_y) \tag{4.53}$$

If we only consider the diagonal line of the Fourier domain as was done in the implicit case, Eq. (4.53) becomes

$$Z = \frac{(\epsilon_{ex}^{(4)} + \epsilon_{ey}^{(4)})}{2}(1-C)^2 + i(CFL_x + CFL_y)S$$

Again defining

$$CFL \equiv CFL_x + CFL_y \tag{4.54}$$

and

$$\epsilon_e^{(4)} \equiv \epsilon_{ex}^{(4)} + \epsilon_{ey}^{(4)}$$

then the one-dimensional amplification-factor, Eq. (3.34), is recovered. The remaining analysis is identical to the one-dimensional derivation. Hence it is logical to calculate the time-step size for the two-dimensional Runge-Kutta scheme based on Eq. (4.54) as

$$\Delta t = \frac{CFL}{|\lambda_{\xi_{max}}| + |\lambda_{\eta_{max}}|}$$

where CFL is a constant for all grid points.

In the chapter 3, we have demonstrated that the residual-smoothing can effectively enhance the convergence rates for one-dimensional problems. We now also consider applying residual-smoothing in two dimensions. It seems logical to extend the one-dimensional residual-smoothing to two dimensions in the following fashion,

$$(1 - \epsilon_x\delta_{xx} - \epsilon_y\delta_{yy})\bar{P} = P \tag{4.55}$$

The stability analysis of Eq. (4.55) would show that the amplification factor for $\bar{P}$ is

$$\bar{Z} = \frac{Z}{1 + 2\epsilon_x(1-C_x) + 2\epsilon_y(1-C_y)}$$

where Z is the amplification factor of operator ΔtP. Once again, if the symmetric analysis is applied and the definition

$$\epsilon \equiv \epsilon_x + \epsilon_y$$

is made, then the remaining analysis exactly follows that of one dimension, and the gain of CFL number is

$$\text{gain} = \sqrt{1+4\epsilon}$$

Although Eq. (4.55) appear to be logical, there is no inexpensive way to invert the left-hand-side operator. Jameson [41] has suggested applying several steps of a Gauss-Sidel iteration (not necessarily until converged) to Eq. (4.55) and has motivated that this provides a significant increase for the allowable CFL number.

This alternative has not been tried here. Instead, Eq. (4.55) has been approximately factorized to get

$$(1-\epsilon_x\delta_{xx})(1-\epsilon_y\delta_{yy})\bar{P} = P \tag{4.56}$$

The amplification factor is then

$$\bar{Z} = \frac{Z}{(1+2\epsilon_x(1-C_x))\,(1+2\epsilon_y(1-C_y))}$$

The symmetric analysis is again utilized to estimate the CFL limit for stability. The amplification factor without artificial viscosity is

$$\bar{Z} = i\frac{CFL\,S}{(1+\epsilon(1-C))^2} \tag{4.57}$$

Detailed algebra shows that instability originates from the wave mode for which the trigonometric functions of its wavenumber are given by

$$C = \frac{-(1+\epsilon)+\sqrt{1+2\epsilon+9\epsilon^2}}{2\epsilon}$$

and

$$S = \frac{\sqrt{2(1+\epsilon)\sqrt{1+2\epsilon+9\epsilon^2}-2-4\epsilon-6\epsilon^2}}{2\epsilon}$$

regardless of the type of Runge-Kutta scheme (3-stage or 4-stage). Substituting C and S into Eq. (4.57) gives

$$\bar{Z} = i\frac{CFL}{\text{gain}}$$

where

$$\text{gain} = \frac{\epsilon\left(5+10\epsilon+9\epsilon^2-3(1+\epsilon)\sqrt{1+2\epsilon+9\epsilon^2}\right)}{\sqrt{2(1+\epsilon)\sqrt{1+2\epsilon+9\epsilon^2}-2-4\epsilon-6\epsilon^2}}$$

A comparison of the CFL number gain between residual-smoothing methods with and without approximate-factorization is shown in Fig. 19. No significant difference between two methods is observed, although approximate-factorization seems to give a larger gain. However, this does not indicate Eq. (4.56) should give faster convergence. The convergence rate is not determined by the CFL number, but rather controlled by the temporal damping rate, or more precisely, the amplification factor. Nevertheless, Eq. (4.56) gives an easier way to apply residual-smoothing. Albeit several iterations of Gauss-Sidel method can be use to relax Eq. (4.55) to a certain extent, the full effect of residual-smoothing may not be reached until Eq. (4.55) is completely relaxed. Therefore the present work utilizes direct inversions of the approximate-factorization formulation (Eq. (4.56)) throughout.

4.4 Comparison of Convergence Rates of Implicit and Explicit Schemes

This section makes comparisons between the implicit and the explicit schemes for two-dimensional incompressible and compressible flows. In two-dimensional cases, the convergence rates of the explicit and implicit schemes are much more competitive than those of one-dimensional cases. The major difference is not caused by an improvement in the explicit scheme but, rather, because the optimal CFL number in the implicit scheme is so much smaller in two dimensions when the usage of approximate-factorization is unavoidable. If the full implicit scheme (rather than ADI) is attempted, then the problem suffered by implicit scheme becomes the inversion of huge sparse matrices. Before any inexpensive algorithm is invented, this is not permissible for most cases of interest.

The problem chosen for the comparison purpose is a bicircular arc airfoil cas-

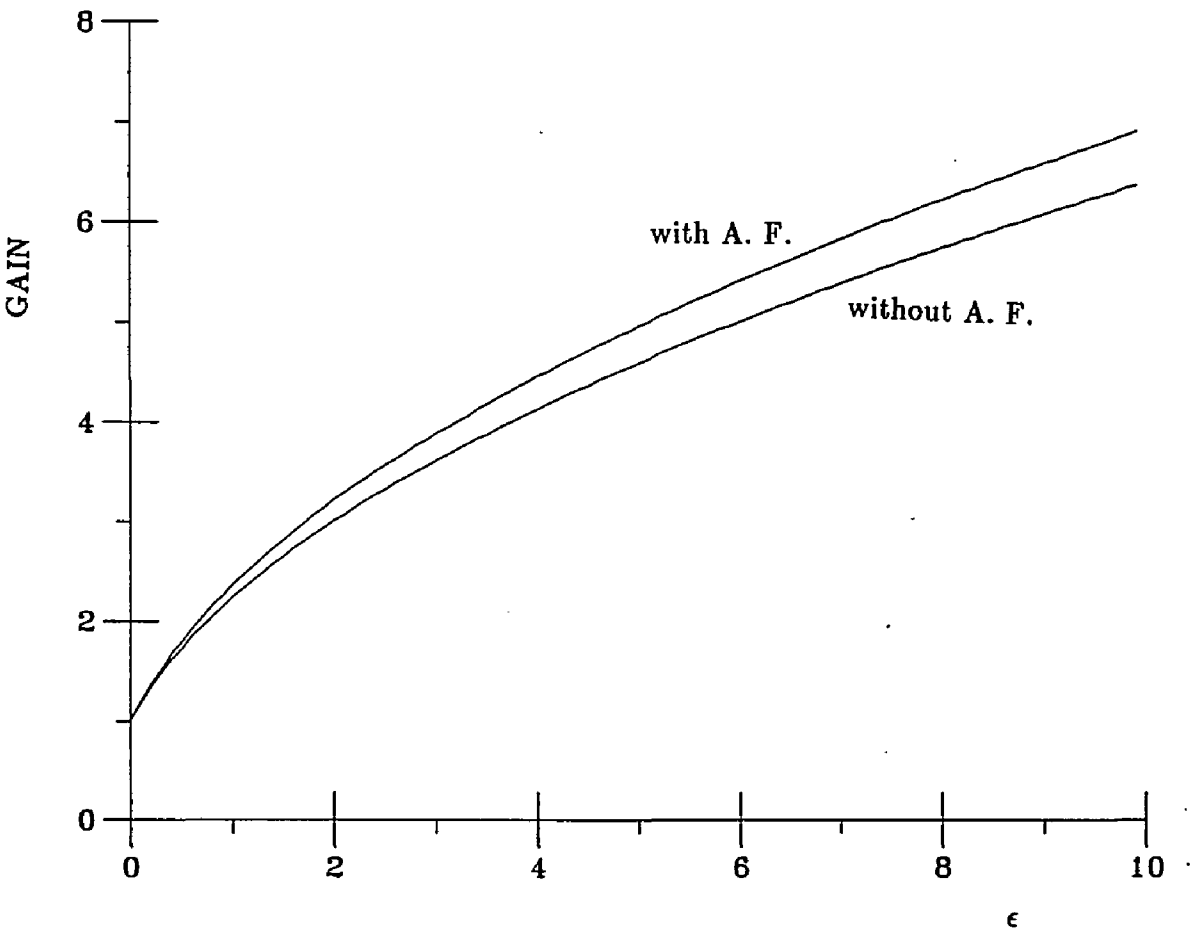

Figure 19: CFL Number Gain Due to Residual-Smoothing for a Runge-Kutta Scheme.

cade problem with 20% chord thickness airfoil. The grid system is a 65×17 H-mesh as shown in Fig. 20 which is identical to those used by Ni [42] and Chima and Johnson [43].

The convergence rates of an incompressible flow case are shown in Fig. 21. The curves in Fig. 21 represent averages of three residuals. Three convergence curves correspond to the ADI scheme, the explicit scheme without residual-smoothing, and the explicit scheme with residual-smoothing. The explicit-scheme solution is identical to those of the ADI formulation when the amount of artificial viscosity is the same, so the relative quality of the solutions is not an issue, only the rate of convergence. One representative solution corresponding to the convergence rate given in Fig. 21 is included as Fig. 22. The smooth pressure contour lines represent a fluctuation-free solution.

Figure 21 shows that the convergence rate of the ADI scheme is much faster than that of the explicit scheme on the basis of iterations, even though the CFL number of the ADI scheme is restricted to approximately the order of the explicit scheme. The curve for the ADI scheme is obtained by using $CFL = 14$. The residual of the ADI scheme is reduced by 15 orders of magnitude in 900 iterations, while it takes 3000 iterations for the explicit scheme to reach the same residual level.

The convergence rate of the explicit scheme can be enhanced significantly by applying residual-smoothing. As shown by stability analysis, residual-smoothing increases both the CFL limit and the allowable viscosity. As a result, the low-wavenumber amplification-factor is significantly reduced to give a faster convergence. As shown on Fig. 21, the one without residual-smoothing takes 12000 iterations (not shown completely on the figure) to converge to machine accuracy; the other with residual smoothing coefficient $\epsilon = 1.0$ converges 4 times faster. As indi-

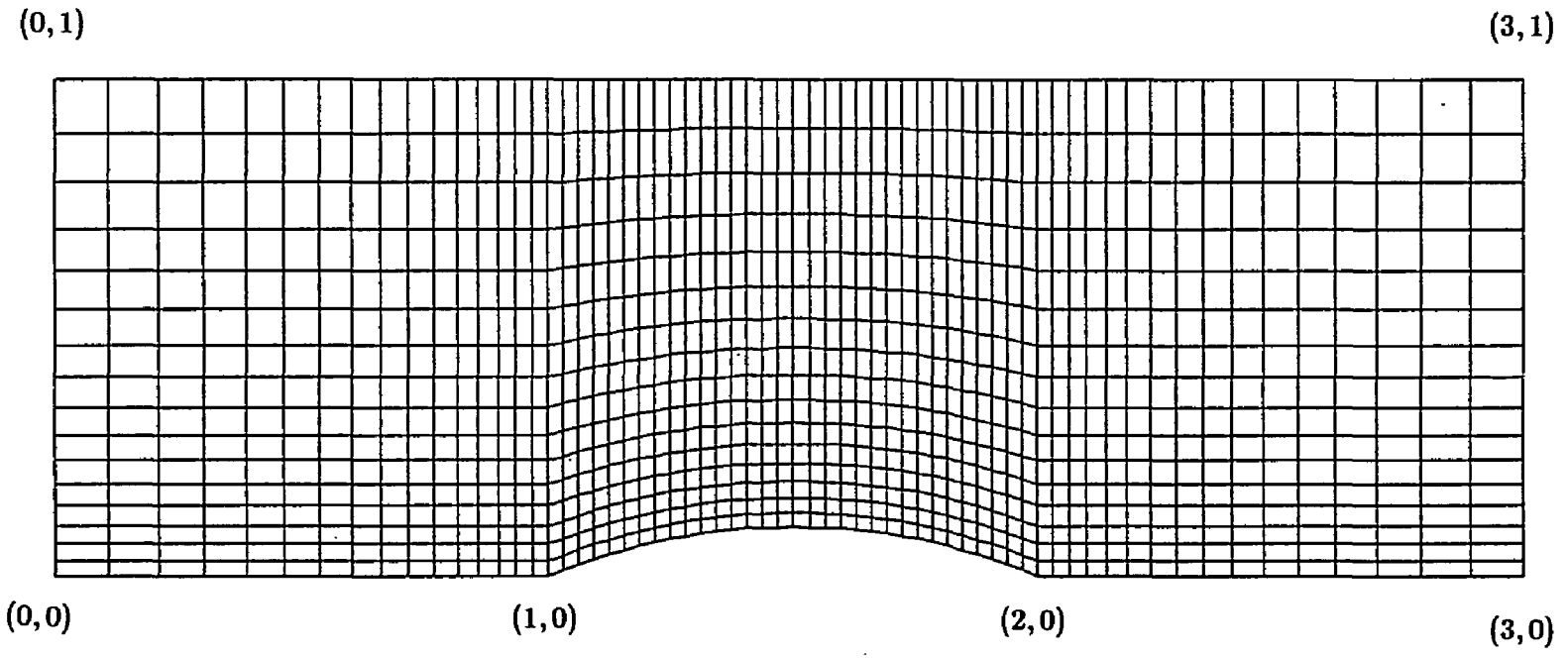

Figure 20: Test Case Geometry.

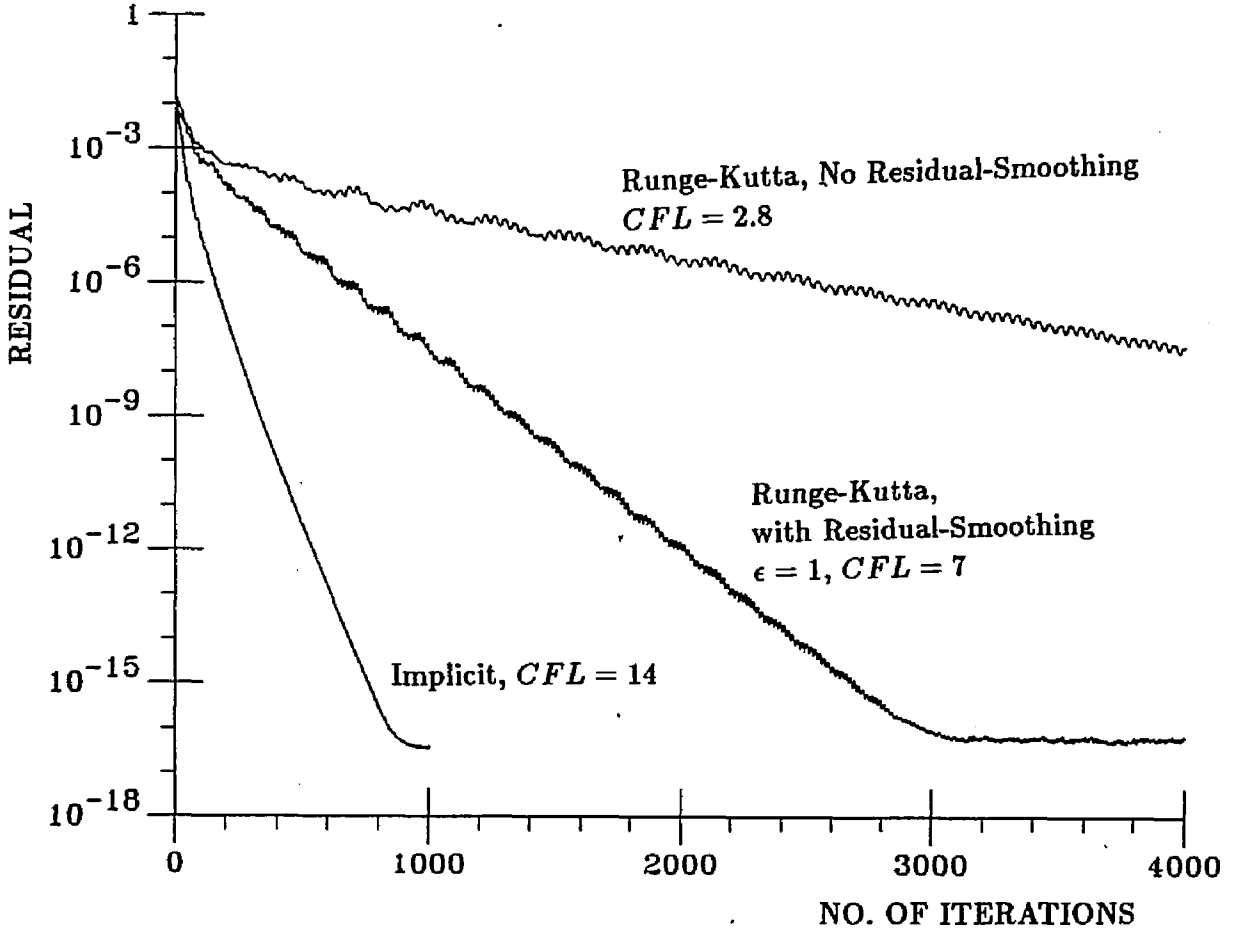

Figure 21: Convergence History of the Implicit and Explicit Schemes for a Two-Dimensional Incompressible Flow.

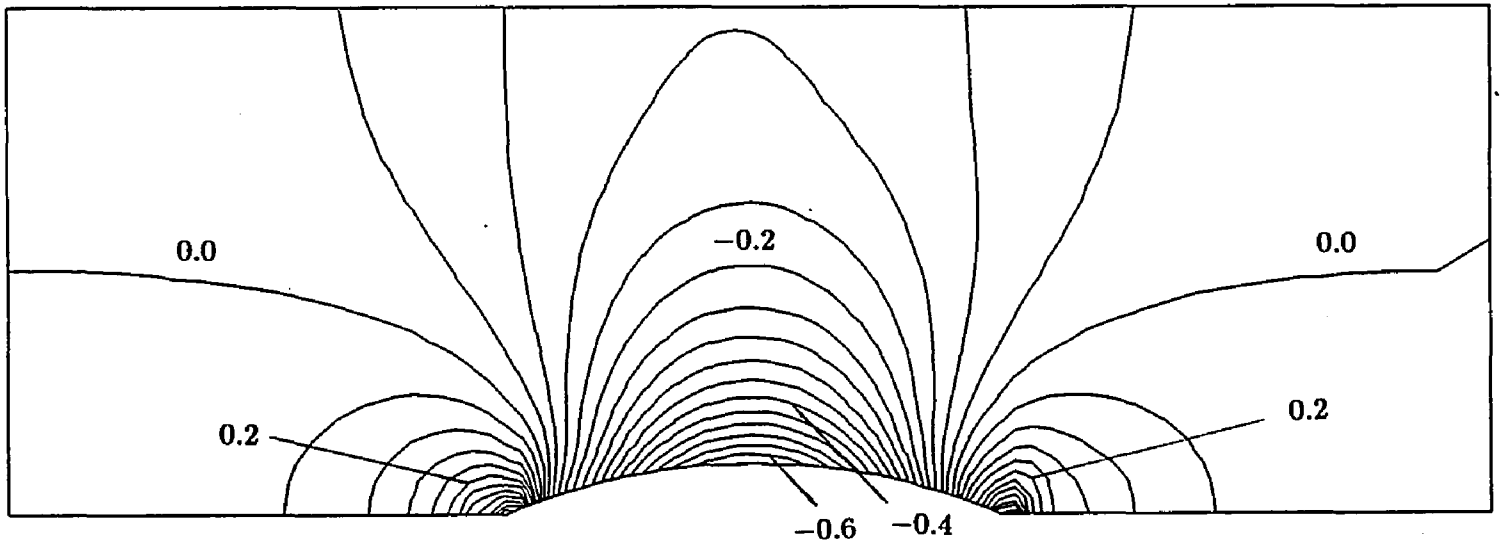

Figure 22: Pressure Contour of the Test Case.

cated for the one-dimensional problems, the amount of viscosity contributes to the convergence rate of the explicit scheme with residual-smoothing. In two dimensions, the same phenomenon has been observed. In fact, the case with residual-smoothing applies a much stronger artificial viscosity ($\epsilon_x = \epsilon_y = 3.5$) than the case without ($\epsilon_x = \epsilon_y = 0.3$).

A more important index of the efficiency of a scheme is the convergence rate based on computer CPU-time, rather than on the number of iterations. Although the ADI scheme converges in much fewer iterations than the explicit one, it consumes more CPU-time for each iteration than the explicit scheme does. For the test case under investigation, the ADI scheme, explicit scheme with residual-smoothing, and explicit scheme without residual-smoothing take 0.172, 0.116, and 0.0806 second per iteration on IBM-3090/180, respectively. IBM-3090/180 is a scalar machine. It performs arithmetic operations on a sequential basis. The CPU-time it consumes for a scheme is thus approximately a relative index of the number of operations. We show the convergence rate based on IBM-3090/180 CPU-time in Fig. 23. As shown in Fig. 23, the ADI scheme is the fastest on a scalar machine.

Evaluating CPU-time has been complicated by the advent of vector machines such as Cray-1, Cray-2, Cyber-205, ETA-10, or IBM-3090/400. A vector machine is capable of parallel-processing an array of data, thus increases its speed dramatically. The longer an array is, the more efficient a vector machine can be. Therefore, the performance of a scheme on vector machines depends strongly on its vectorizability. For example, on a vector machine the tridiagonal inversions of the two-dimensional ADI scheme should not be done in a line-by-line fashion, but should be parallel-inverted, or 'line-vectorized'. The residual-smoothing in the explicit scheme, which involves inversions of scalar tridiagonal matrices, can be vectorized with the same strategy. The right-hand-side calculation can even be

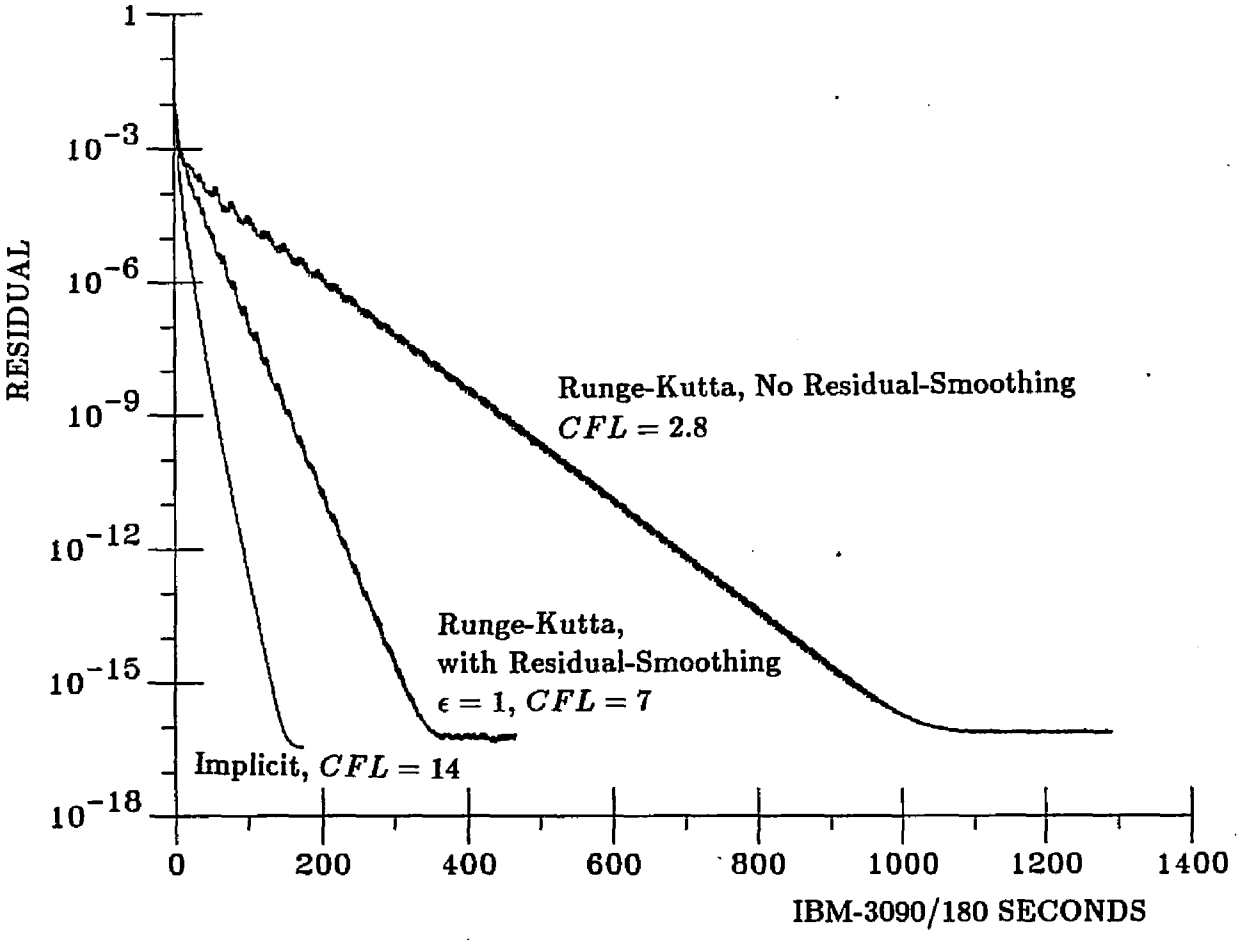

Figure 23: Scalar-Machine Cost Comparison of the Implicit and Explicit Schemes for a Two-Dimensional Incompressible Flow.

more efficiently 'plane-vectorized' in either case. However, since the explicit scheme without residual-smoothing requires only the right-hand-side calculations, its overall efficiency on a vector machine may be better than on a scalar machine. Figure 24 shows the comparison of both schemes based on Cyber-205 CPU-time. The CPU-time for each iteration is 0.0661, 0.0263, and 0.0109 for the ADI scheme, the explicit scheme with residual-smoothing, and the explicit scheme without residual-smoothing, respectively. As can be seen in Fig. 24, the ADI scheme is still the most economical one. However, the gap between the CPU-time cost for the ADI and the explicit schemes has been narrowed due to their vectorizabilities.

Another critical factor to the CPU-time is the block size, i.e., the number of governing equations. In the ADI scheme, most of the computation effort is on inversions of the left-hand-side block tridiagonal matrices. The number of operations in solving a block tridiagonal increases with cube of the block size, or more specifically, $\frac{14}{3}m^3 + \frac{9}{2}m^2 - \frac{1}{6}m$, where m is the dimension of the blocks [44]. The trend of required operation counts for left-hand-side inversions is shown in Fig. 25. One perceives the drastic escalation of computation effort for the ADI scheme as the block size increases. On the other hand, the number of operations required by the explicit scheme increases linearly with the number of equations. Therefore, for cases with more equations, e.g., two- and three-dimensional compressible flows or three-dimensional incompressible flows, the explicit scheme may be more favorable. Here we will present the convergence comparisons for the two-dimensional compressible flow. Three-dimensional cases will be discussed in the next chapter.

Same as for the incompressible flow, the compressible flow is solved by the ADI scheme and the explicit scheme, with and without residual-smoothing. The bicircular airfoil cascade geometry as shown in Fig. 20 is used once again for consistency. The back pressure is held to be 0.8 of the inlet stagnation pressure

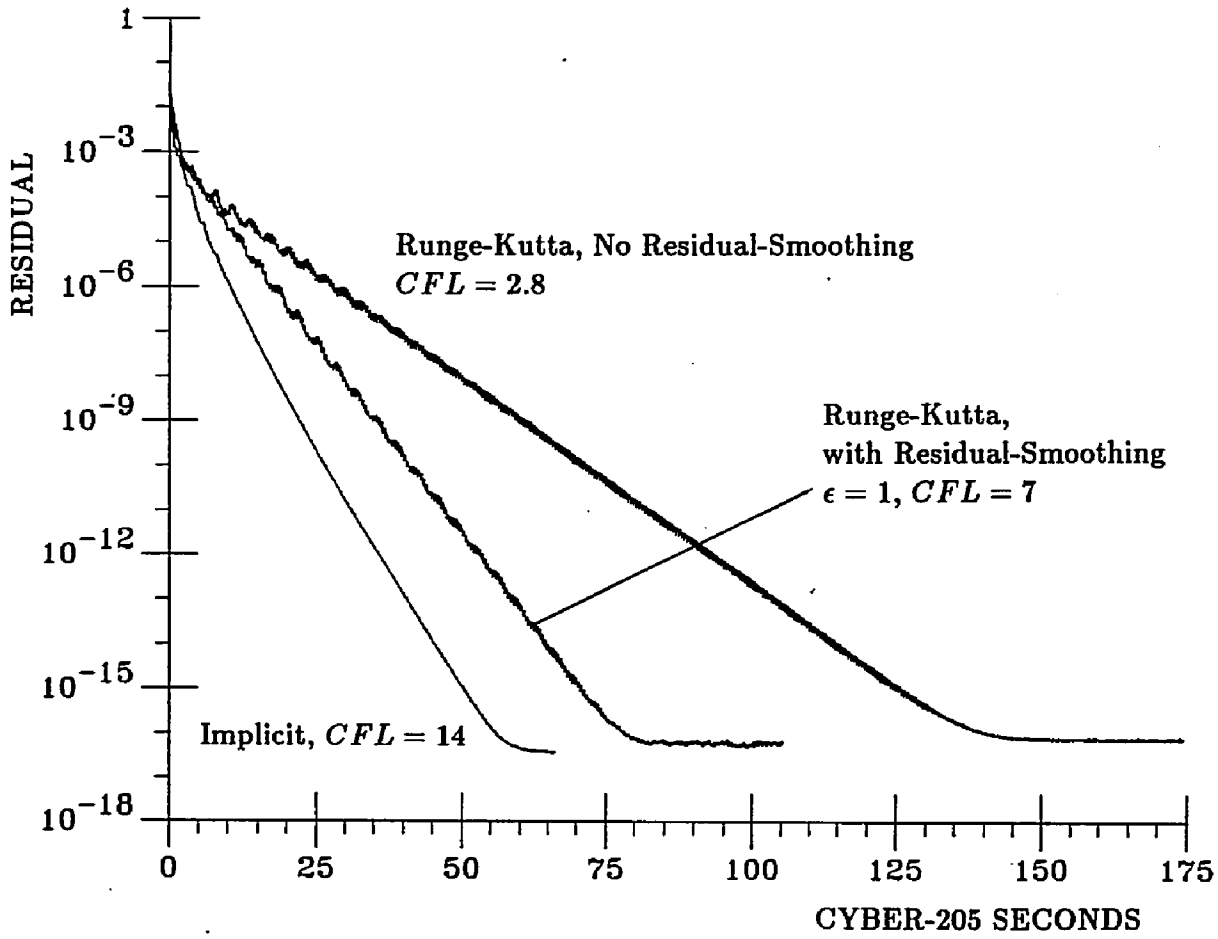

Figure 24: Vector-Machine Cost Comparison of Implicit and Explicit Schemes for a Two-Dimensional Incompressible Flow.

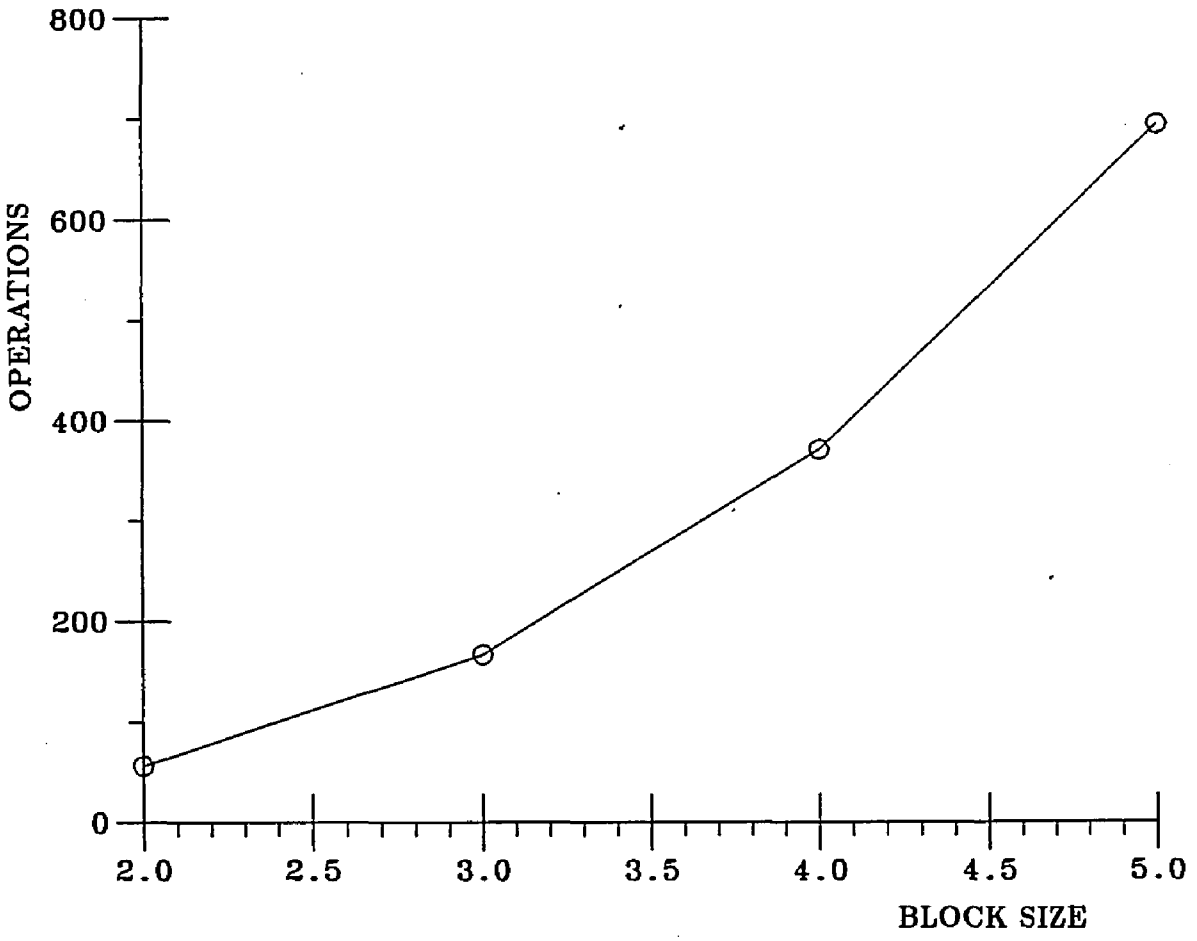

Figure 25: Arithmetic Operation Counts of a Block-Tridiagonal Solver.

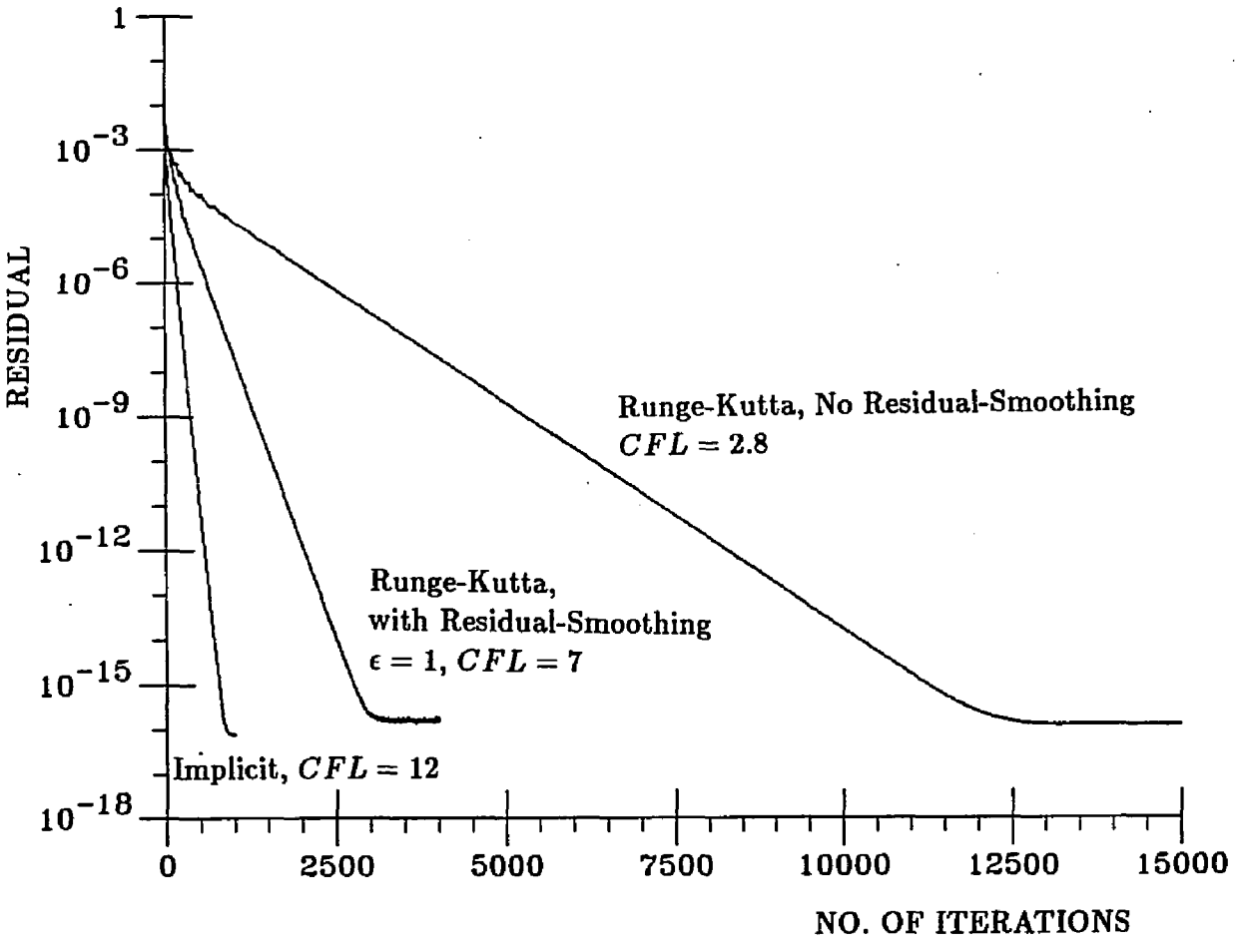

Figure 26: Convergence History of the Implicit and Explicit Schemes for a Two-Dimensional Compressible Flow.

such that the flow is subsonic and shock-free. The convergence curves based on the number of iterations are shown in Fig. 26. These convergence rates are very similar to those for the incompressible case. The optimal CFL number for the ADI scheme is 12, which is slightly smaller than 14 for the incompressible case. The CPU-time for each iteration of the schemes are, 0.360, 0.214, and 0.164 second on the IBM-3090/180, and 0.126, 0.0392, and 0.0186 second on the CYBER-205, in the sequence of the ADI scheme, the explicit scheme with residual-smoothing, and the explicit scheme without residual-smoothing. These data show that increasing the block size does give a preference to the explicit scheme. However, this preference seems to be less than we estimated previously. The reasons are that the RHS calculation does contribute a significant portion to the computation, and that the RHS of the compressible system happens to be much more time-consuming than the incompressible ones. The RHS for the compressible system involves not only more operations but also square-root calculations. Since the four-stage Runge-Kutta scheme has to compute the RHS four times per iteration, the CPU-time per iteration for both schemes is again brought closer. The convergence rates based on the CPU-time of the scalar machine and the vector machine are shown in Fig. 27 and Fig. 28. The ADI scheme is once again preferred on both types of machines. However, the speed of the explicit scheme on a vector machine is almost the same as the ADI scheme.

One important factor on the convergence rates yet to be discussed is the boundary-condition implementation. The convergence rate of a scheme is dictated by the boundary conditions as well as the dissipation rate of the field points. Efficient boundary conditions must be capable of driving the waves coming from the field points away through the boundaries. Boundary conditions incapable of doing this may cause slow convergence or even instability. We have presented both the

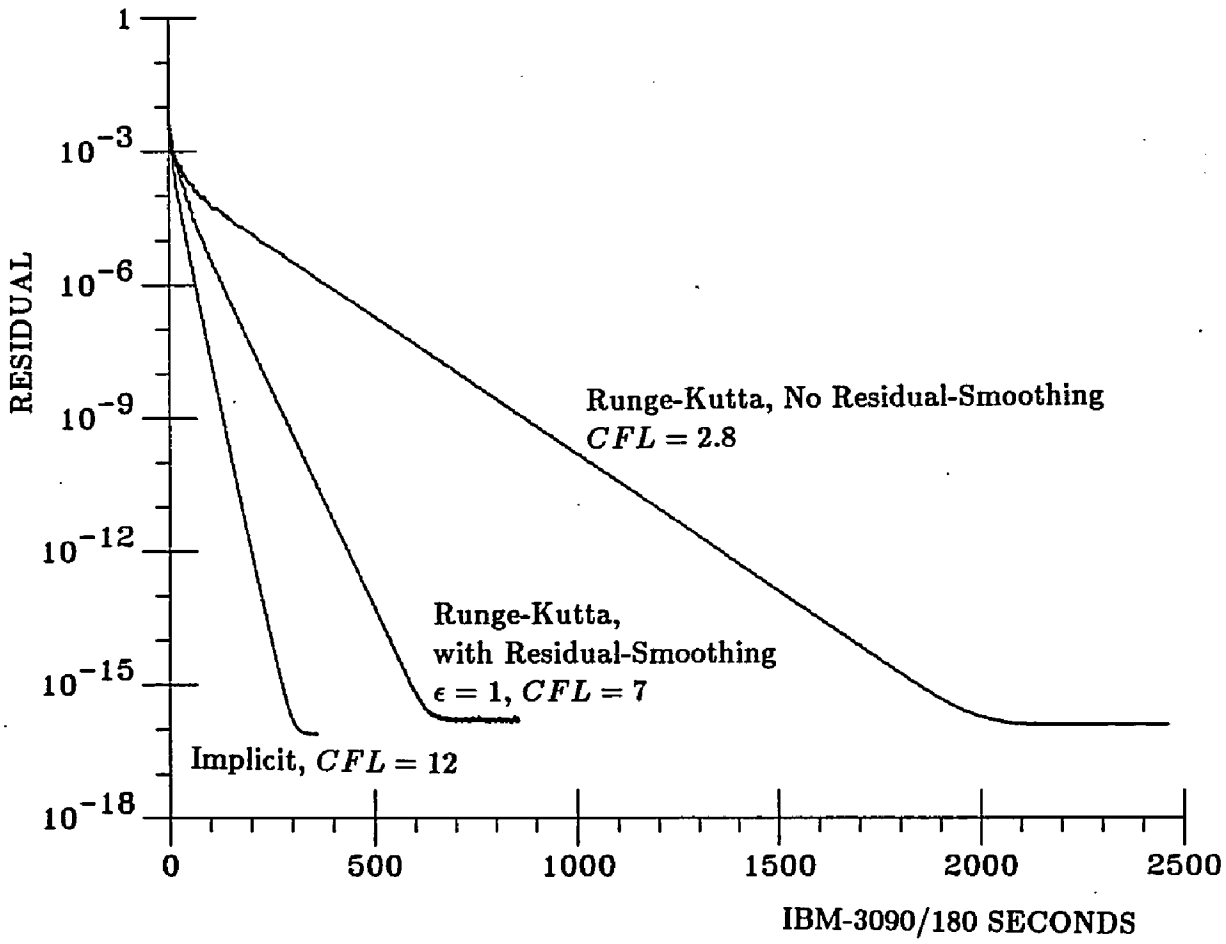

Figure 27: Scalar-Machine Cost Comparison of Implicit and Explicit Schemes for a Two-Dimensional Compressible Flow.

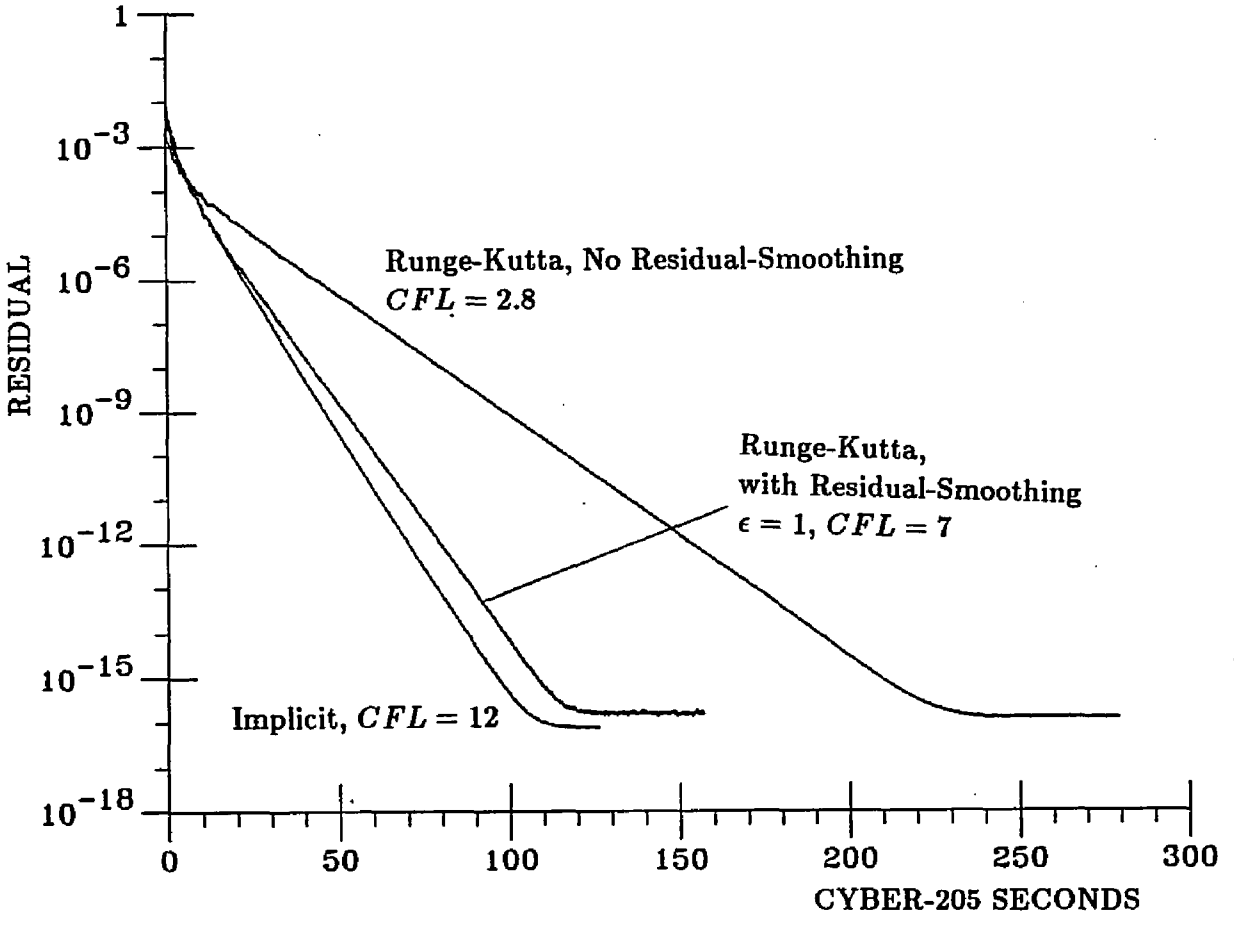

Figure 28: Vector-Machine Cost Comparison of Implicit and Explicit Schemes for a Two-Dimensional Compressible Flow.

MOC boundary conditions and the explicit boundary conditions earlier in this chapter. Both procedures can be applied to either the implicit or the explicit schemes. Here we will discuss their transient behaviors when applied to the implicit scheme.

The MOC boundary procedure usually offers a non-reflective wave propagation on the boundaries. This feature increases stability and convergence rate during the time-marching history. To visualize this, we apply the MOC and explicit boundary procedures to the ADI scheme for an incompressible flow passing the bicircular airfoil cascade shown in Fig. 20. The pressure distribution for the explicit boundary procedure after 5 iterations is shown in Fig. 29. Clearly, the explicit boundary procedure can not evict the disturbances out of the boundaries completely. Strong oscillation is observed on the wall. However, this phenomenon is absent when the MOC boundary procedure is used. Figure 30 shows the pressure contour after 5 iterations for MOC boundary procedure. Well behaved wave propagation can clearly be seen on the wall boundary. Consequently, we expect this implicit scheme to converge faster with MOC boundary conditions. Surprisingly, however, both procedures converge at nearly the same rate for $CFL = 14$. This indicates that the set of explicit boundary conditions used here can catch up with the dissipation rate of the field points for the ADI scheme at $CFL = 14$. What happens if the field points have a higher dissipation rate? To answer this question, we apply both boundary procedures to the full implicit scheme which offers a higher dissipation rate as the CFL number further increases. The convergence rates for the full implicit scheme with the explicit and MOC boundary procedures are shown in Fig. 31 and Fig. 32 for some CFL numbers. Note that the full implicit scheme converges faster at $CFL = 14$ than the ADI scheme at the same CFL number. Once again we see that the convergence rate is decided by the amplification factor instead of the CFL number. As shown in Fig. 31, the convergence with the explicit boundary procedure

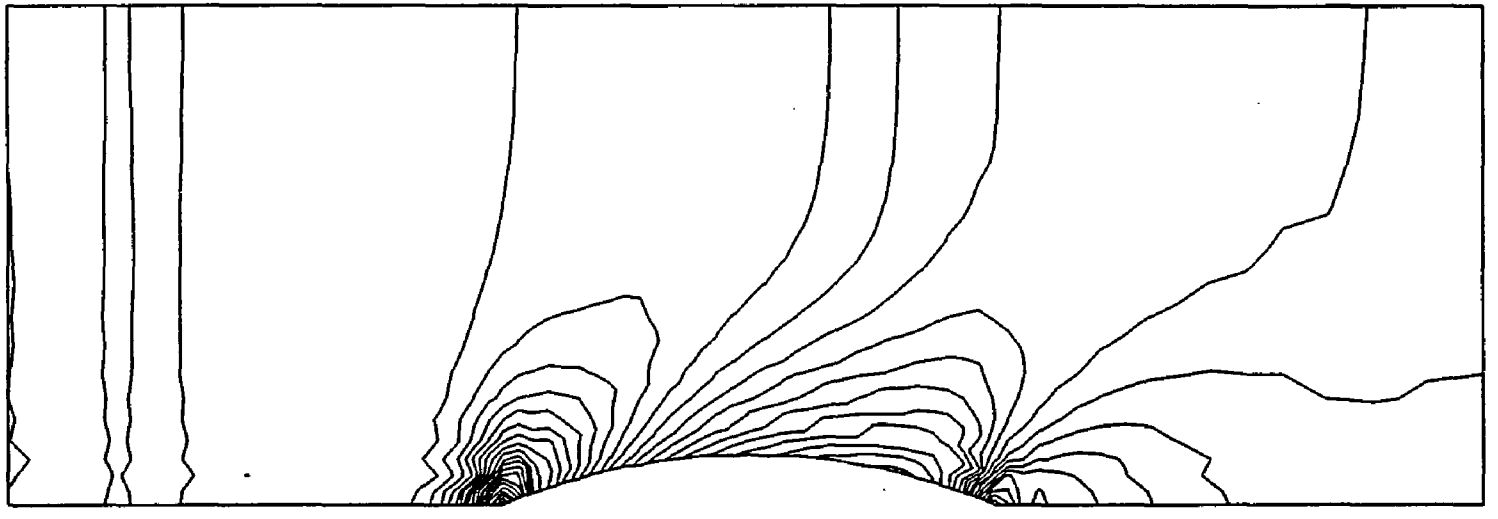

Figure 29: Pressure Contour by using the Explicit Boundary Conditions after 5 Iterations.

is enhanced as the CFL number increases from 14 to 20, but slowed down for further increase of the CFL number. On the other hand, the MOC boundary procedure can cope with further increase of the CFL number, and renders the full implicit scheme capable of converging 16 orders of magnitude in only 13 iterations with a CFL number of 10000.

A final word concerning comparisons between ADI and explicit schemes is in order. The CFL number used with Runge-Kutta scheme is approximately equal to the optimal value for ADI scheme. It is tempting on this basis to assume that equal CFL numbers will imply equal convergence rates. This is not the case. The convergence rate is dictated by the amplification factor and its variation with wavenumber, not by the CFL number. As a matter of fact, even if $CFL = 7$ is used for the ADI scheme, its convergence rate is still faster than the Runge-Kutta scheme with $CFL = 7$. Moreover, the ADI scheme can actually use a much higher CFL number (say, 100 or 1000) and still be stable. However, the convergence rate will be much slower as compared to $CFL = 14$. Stated in another way, it is not the time-step size that determines convergence, but rather the temporal dissipation rate. These are most directly deduced from the amplification factors.

4.5 Accuracy Verification for Steady State Solutions

In this section, several steady state solutions are presented to show the accuracy of the central-difference schemes under investigation. We selected both compressible and incompressible cases whose analytical solutions are available to make comparisons. For the compressible case, a flow entering a channel with a five-degree wedge is calculated. As representative incompressible cases, we calculated flows over a cylinder and a NACA0012 airfoil.

As stated earlier, the converged solutions of the implicit and explicit schemes are, for all intents and purposes, identical. Although the solutions in this section

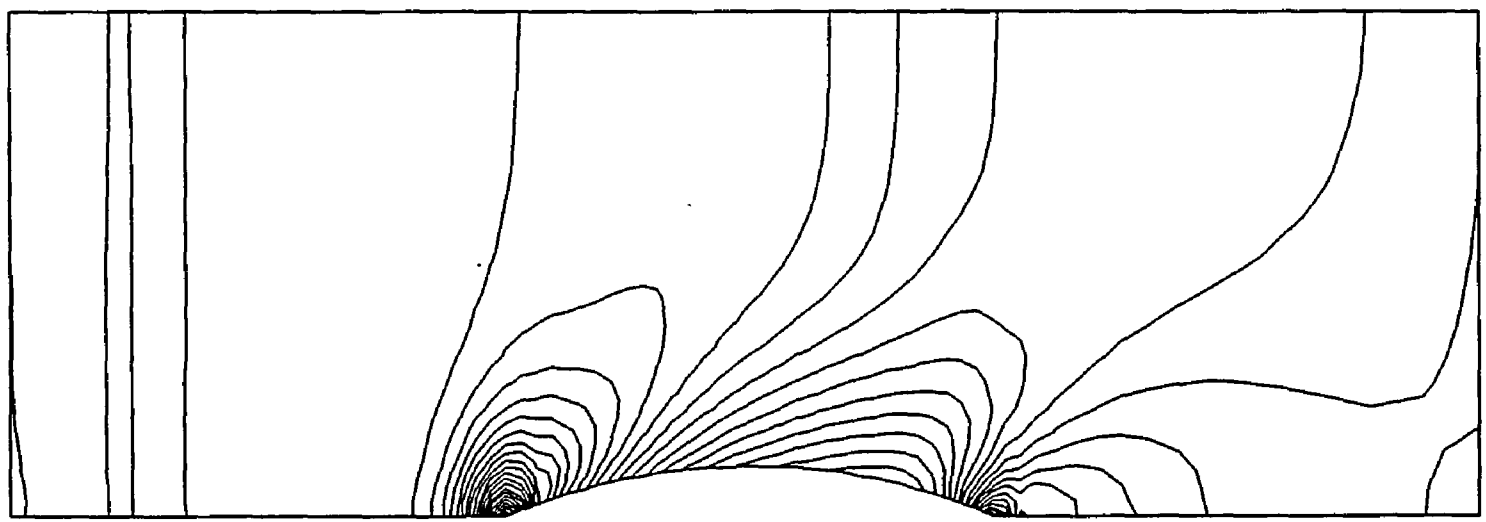

Figure 30: Pressure Contour by using the MOC Boundary Conditions after 5 Iterations.

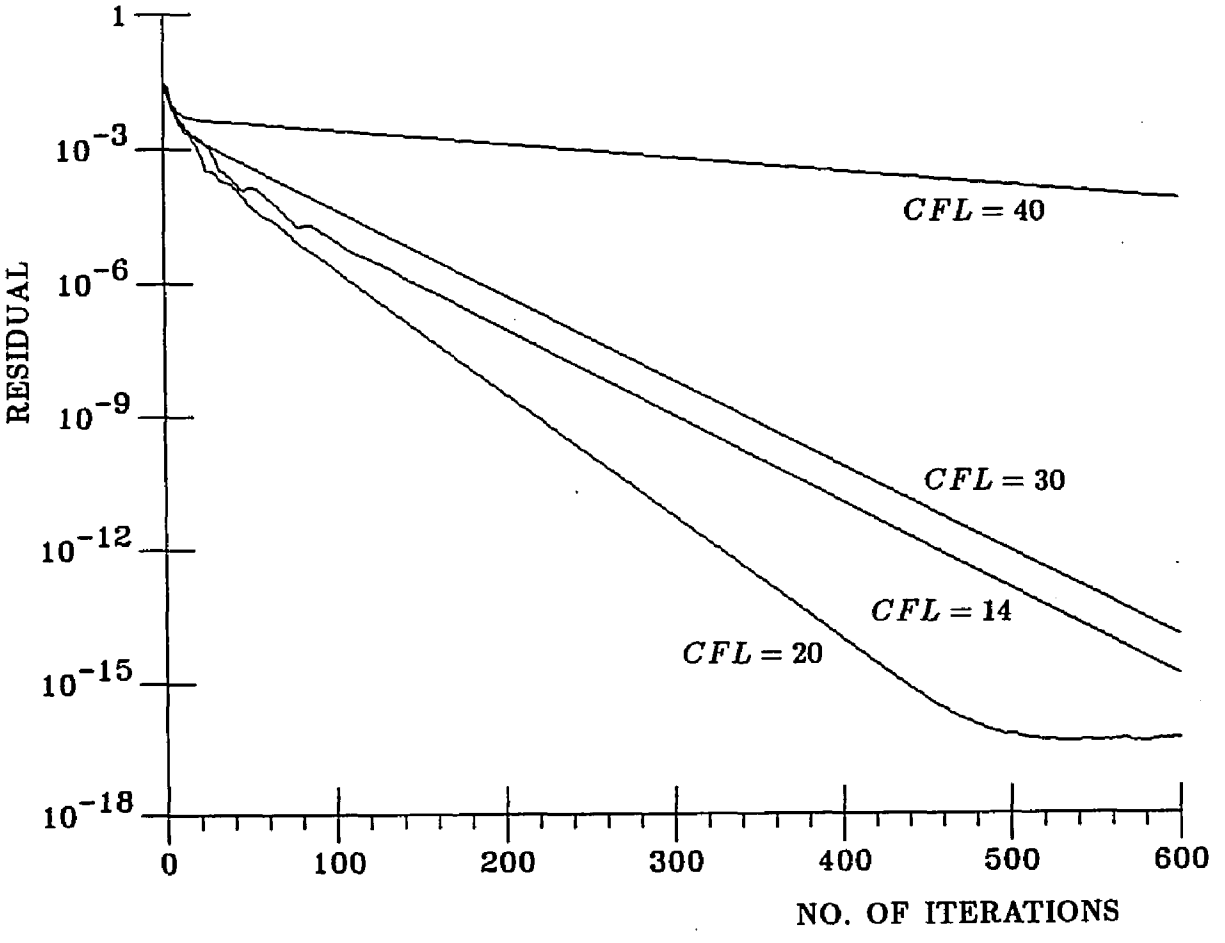

Figure 31: Convergence History of the Full Implicit Scheme with the Explicit Boundary Conditions.

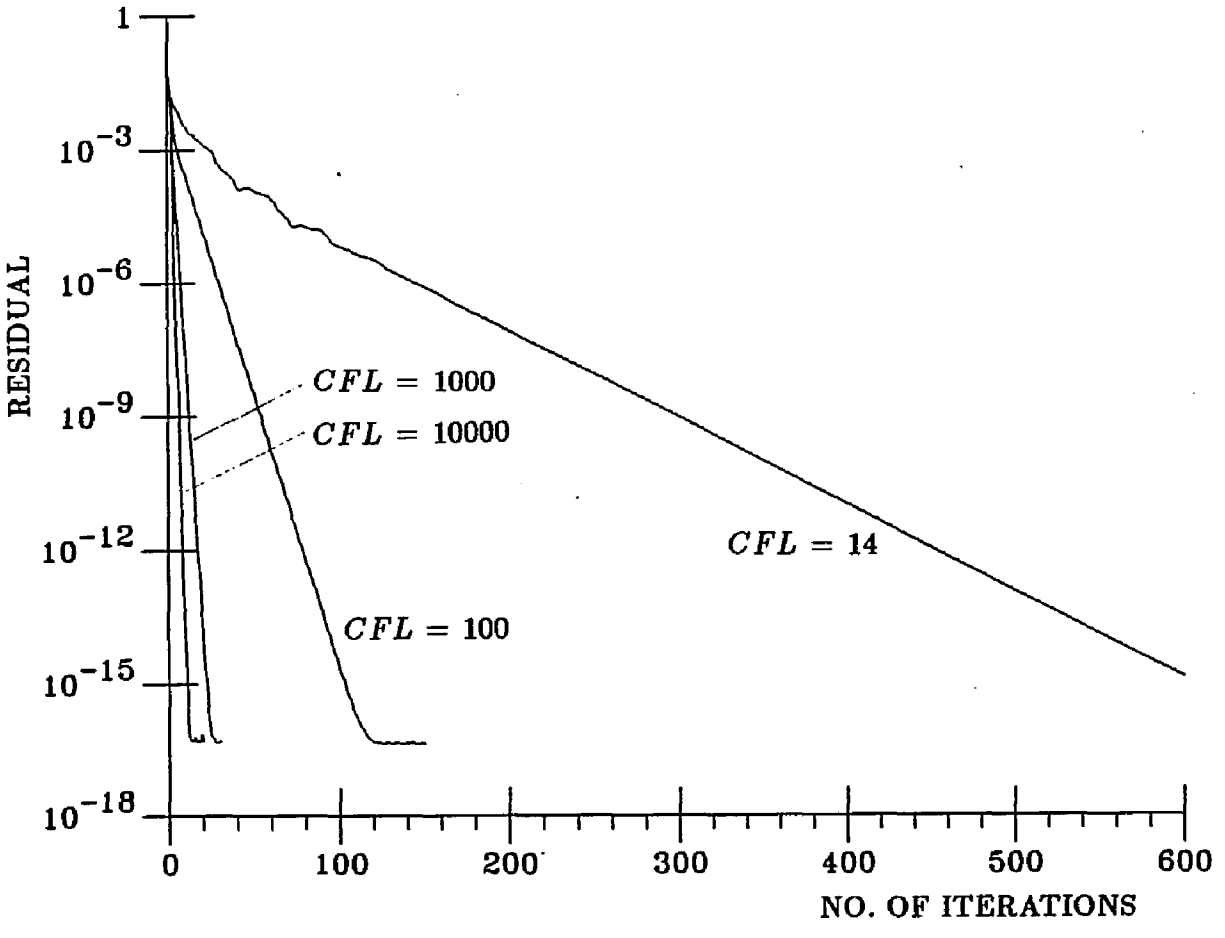

Figure 32: Convergence History of the Full Implicit Scheme by using the MOC Boundary Conditions.

are obtained by utilizing the implicit scheme, they also represent solutions of the explicit scheme. These solutions correspond to results as double-precision machine accuracy is reached. However, three orders of residual reduction is sufficient for practical purposes.

Figure 33 shows the grid system, the numerical result, and the analytical solution for a compressible internal flow. A Mach 3 flow enters a two-dimensional channel with a 5-degree wedge at the lower wall. The top figure shows the 81×33 grid system being used. The grid points are equally spaced in both x- and y-directions. Sharp turns of the lower solid wall create an oblique shock wave and a Prandl-Meyer expansion wave. The oblique shock reflects on the upper wall and merges into the expansion wave. The middle figure shows the Mach number contour results from the numerical computation. Such a case can be calculated analytically. The analytical result is shown in the lower figure. The computed wave-angles agree very well with the analytical calculation, and the Mach numbers are within 1% accuracy.

As a second example, an incompressible flow around a cylinder is computed and compared against analytical solution. The grid system (90×41) utilized to compute the flow field is shown in Fig. 34. The radius of the computational domain is approximately 10 times of the cylinder radius. Figure 35 shows the pressure contour plot. The boundary conditions applied are the total pressure and the zero flow angle at the front side, and a constant pressure at the back side. These non-symmetric boundary conditions results in a non-symmetric solution. However, a symmetric solution should be approached if the domain is expanded to infinity. For the domain size (10 cylinder radii) used here, the pressure distribution is already very close to symmetric. A comparison of the pressure coefficient on the cylinder surface is shown in Fig. 36. The solid line represents the analytical solution and

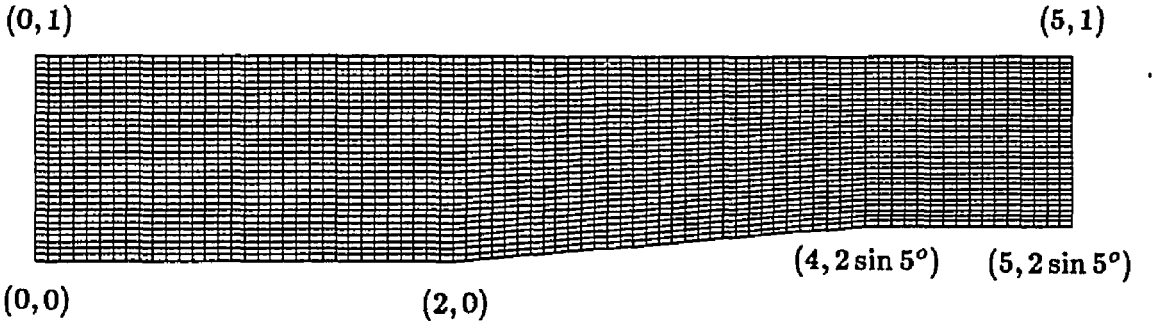

Figure 33a: Geometry of a Supersonic Inlet with a 5-Degree Wedge.

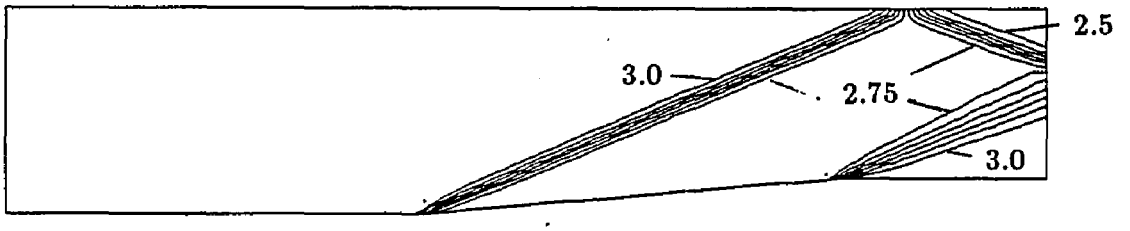

Figure 33b: Mach Number Contour of the Numerical Solution.

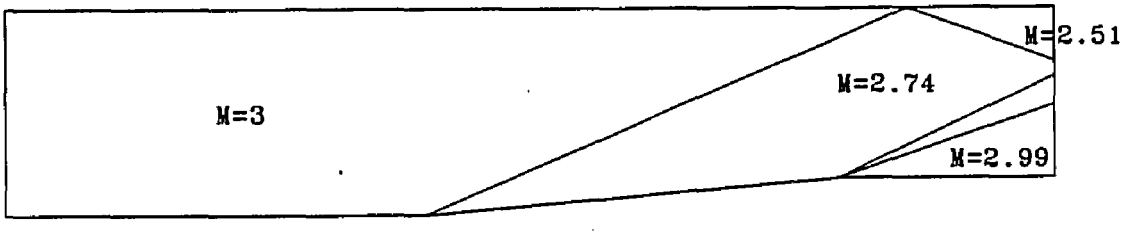

Figure 33c: Schematic of the Analytical Solution.

the circles stand for the numerical solution. Small discrepancies are observed at the top and bottom of the cylinder (minimum pressure region). This is also caused by moving the far field conditions to a finite distance from the body. This error is kept within 1% of the maximum pressure difference in the flow field by restricting the far field conditions 10 cylinder radii away from the body. Expanding the domain size can further reduces this error. To show how many orders of residual reduction are required, the pressure distribution after reducing the residual by three orders of magnitude is also presented in Fig. 36 as crosses. The difference made by further residual reduction is visually not detectable.

A third comparison regards a uniform flow passing through a two-dimensional cascade of NACA0012 airfoils. The calculated result is compared against Giesing's panel method program [26]. The panel method solves the velocity potential (Laplace) equation for an arbitrary boundary. In the case presented, 400 panels are used to represent the airfoil, so that the solution can be considered as analytical. As shown in Fig. 37, the pressure coefficients on the airfoil surface agree with each other accurately.

We conclude this section by a brief discussion on the effect of the boundary conditions on the solution accuracy. Boundary conditions play a decisive role in solution quality on many occasions. We have presented the MOC boundary conditions and examples of the explicit boundary conditions in the earlier sections. It is quite clear that the MOC boundary conditions should give a better solution since the equations for boundaries come directly from the governing equations without revoking any extra assumptions as those made for the explicit boundary conditions presented.

As an example, a subsonic compressible flow around a NACA0012 airfoil at zero angle of attack is computed using both the MOC and the explicit boundary

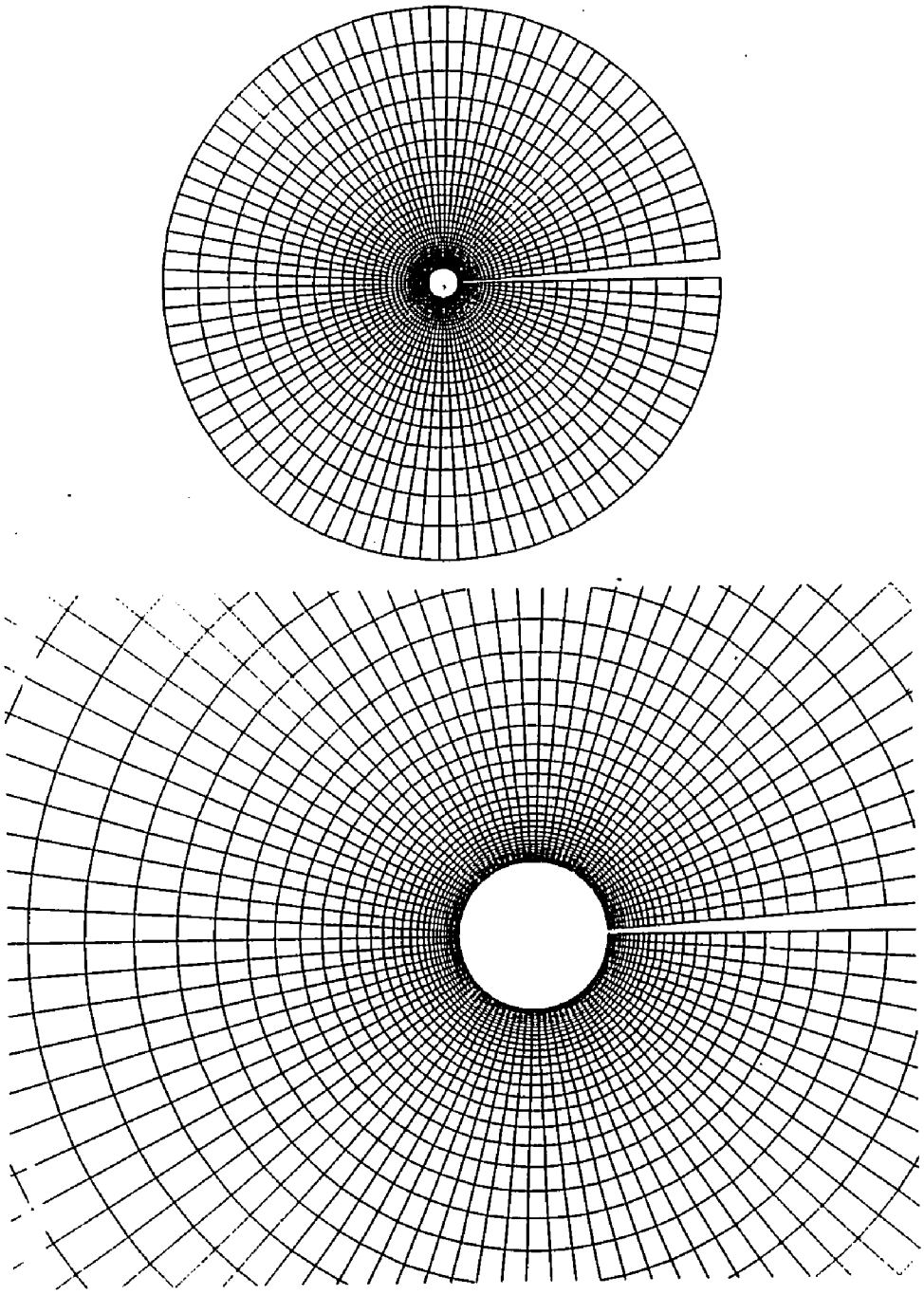

Figure 34: 90×41 Grid Around a Cylinder.

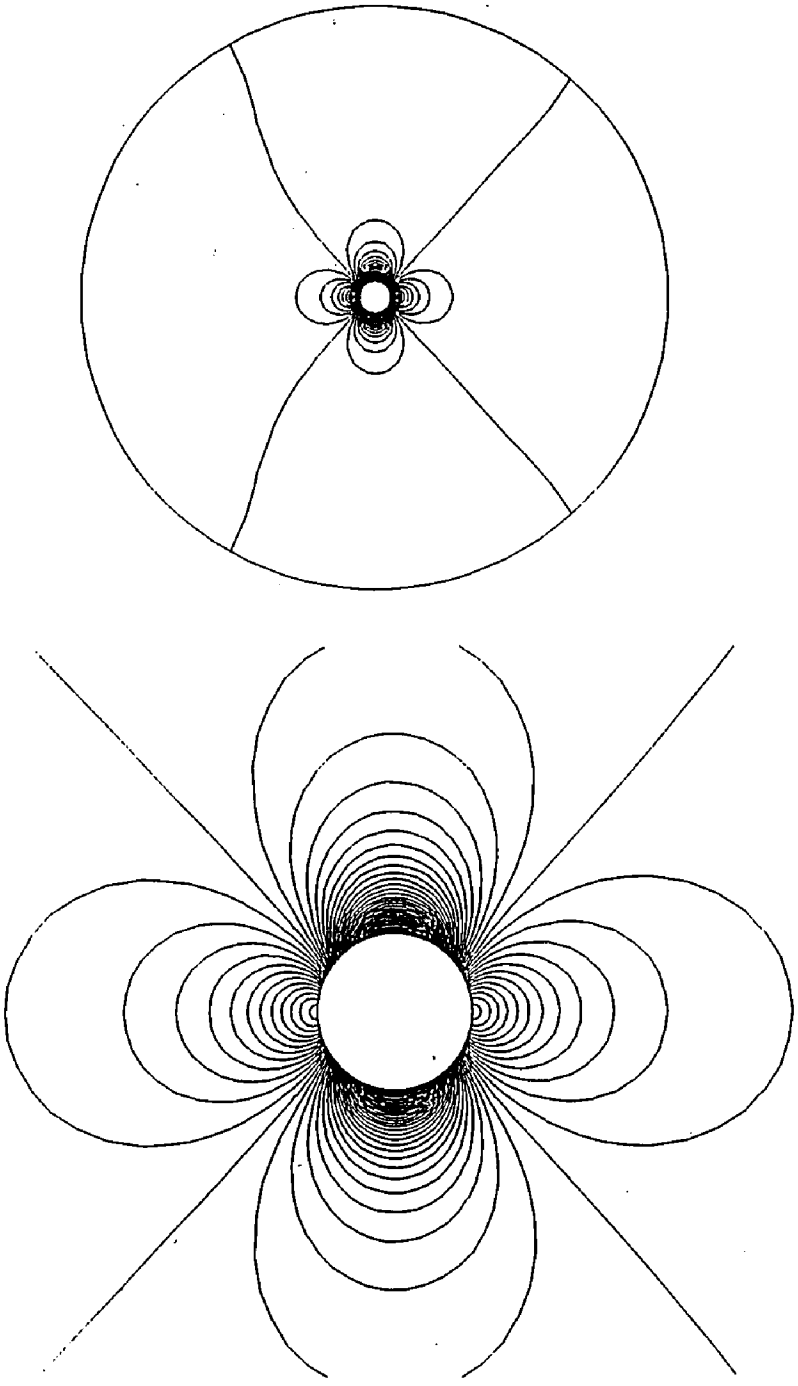

Figure 35: Calculated Pressure Contour Around a Cylinder.

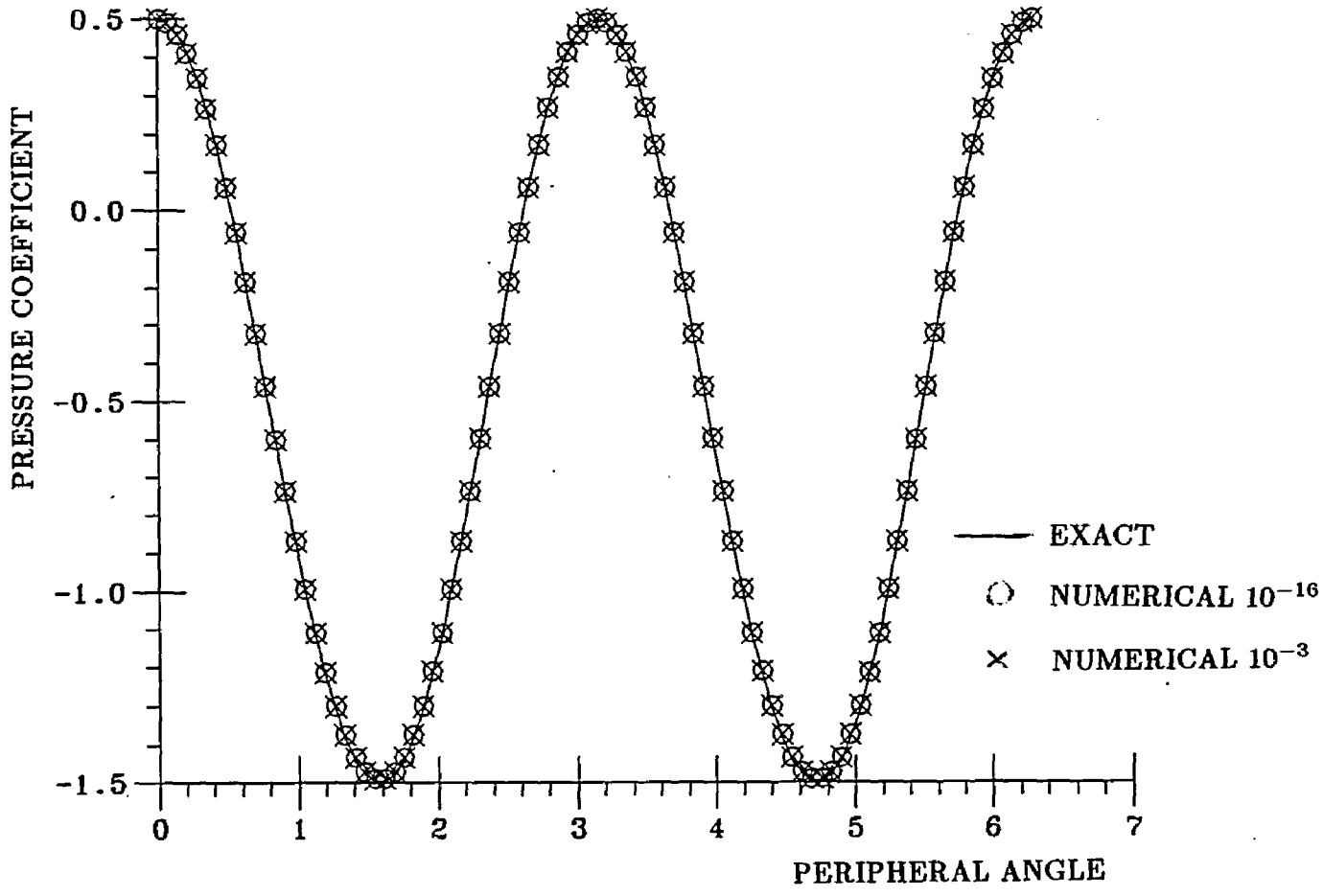

Figure 36: Pressure Coefficient Distribution on a Cylinder Wall.

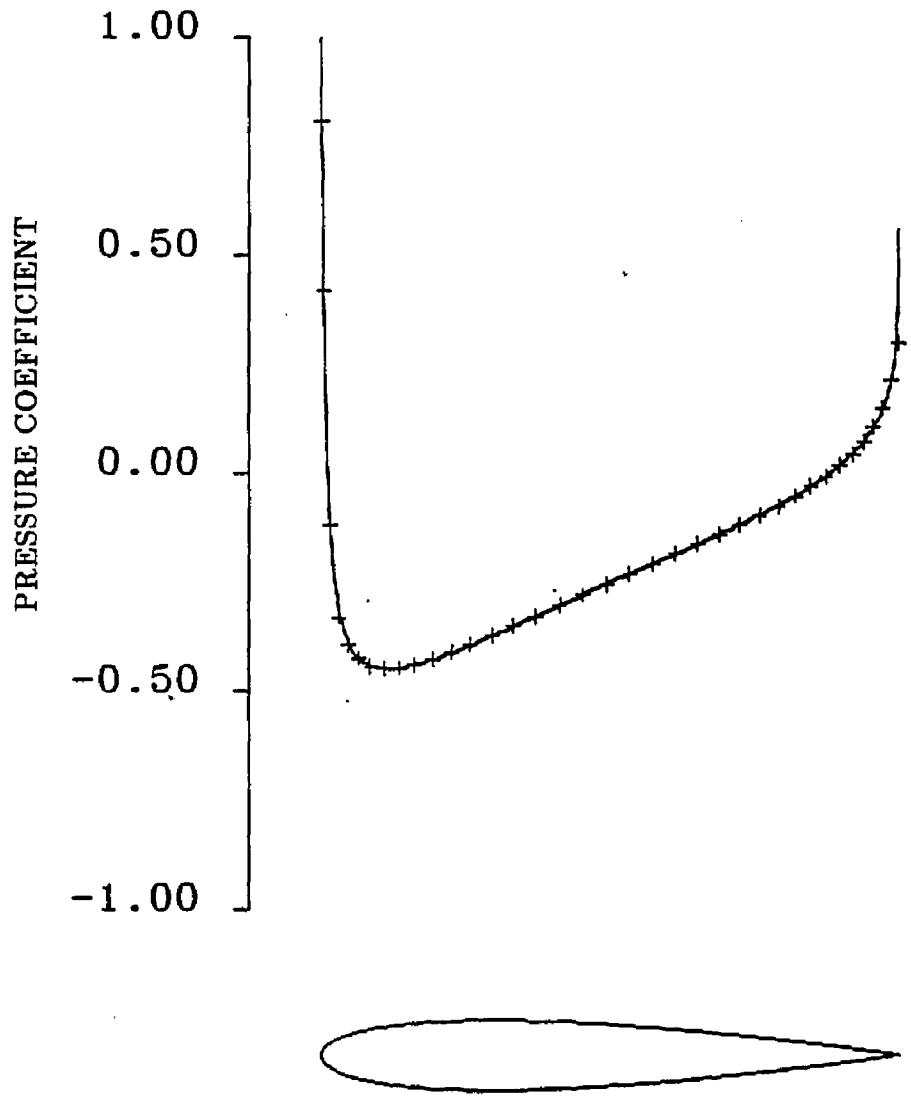

Figure 37: Pressure Coefficient Distribution on a NACA0012 Airfoil.

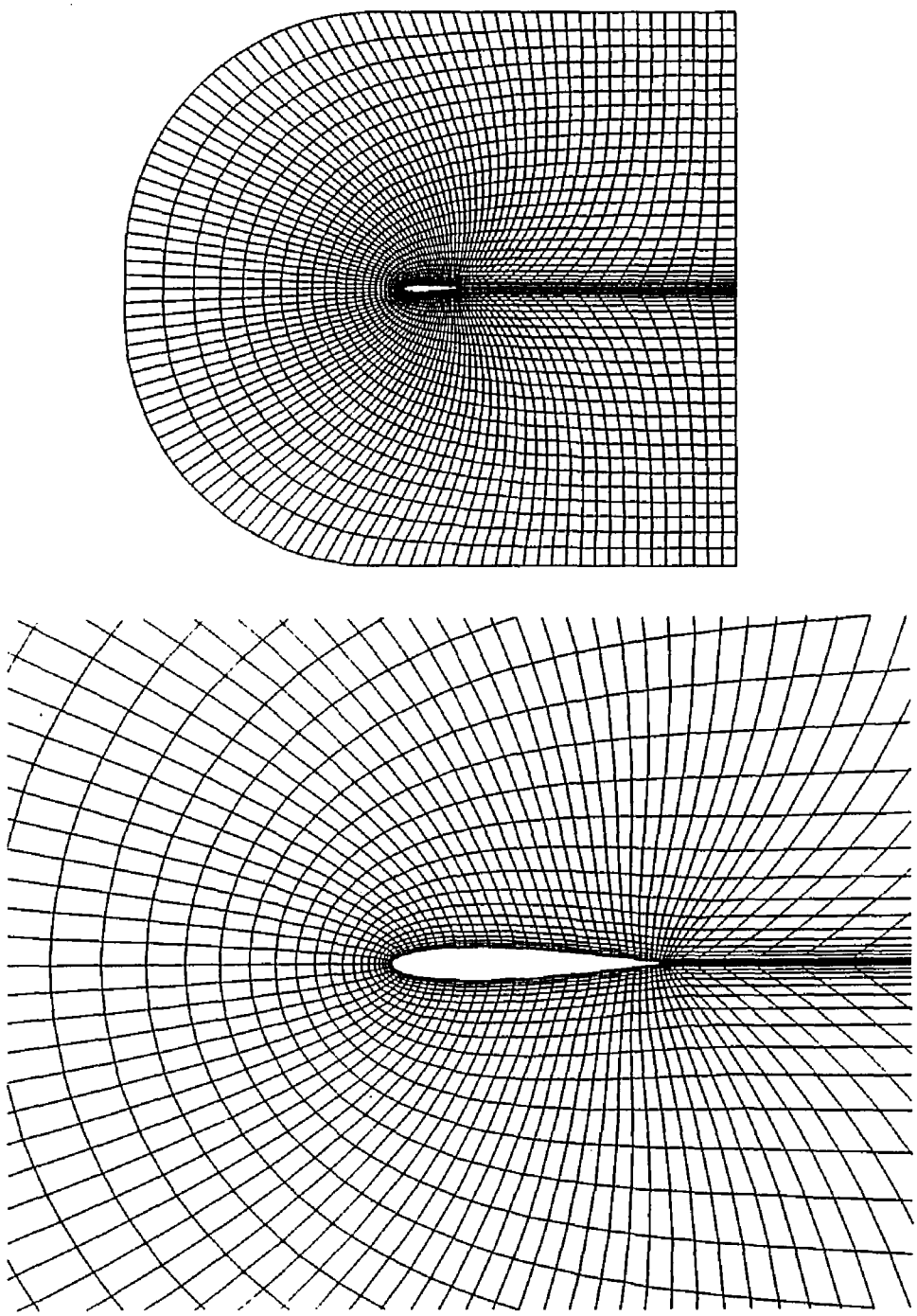

Figure 38: 111 × 31 Grid System Around a NACA0012 Airfoil.

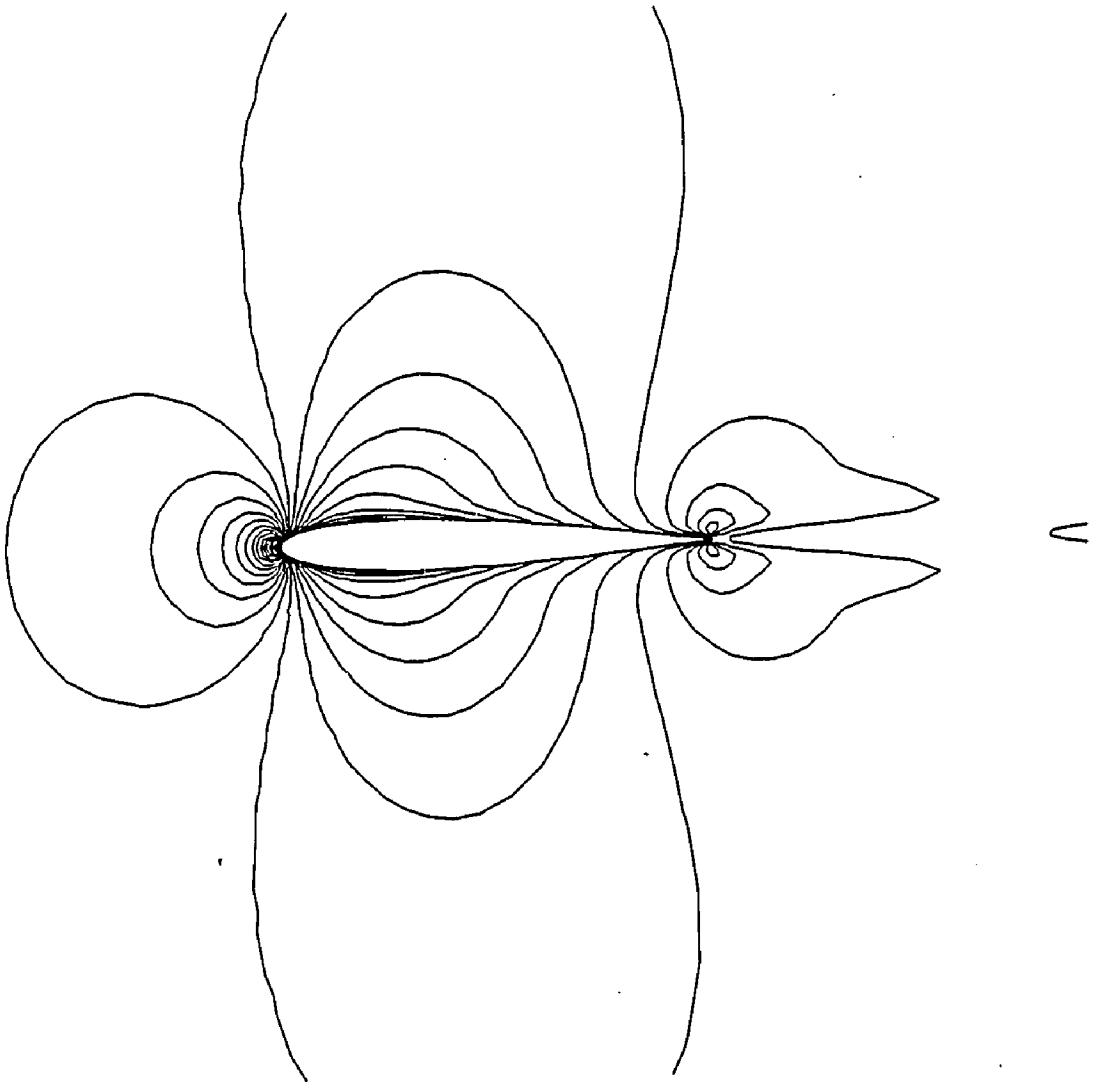

Figure 39: Mach Number Contour by Using the Explicit Boundary Conditions.

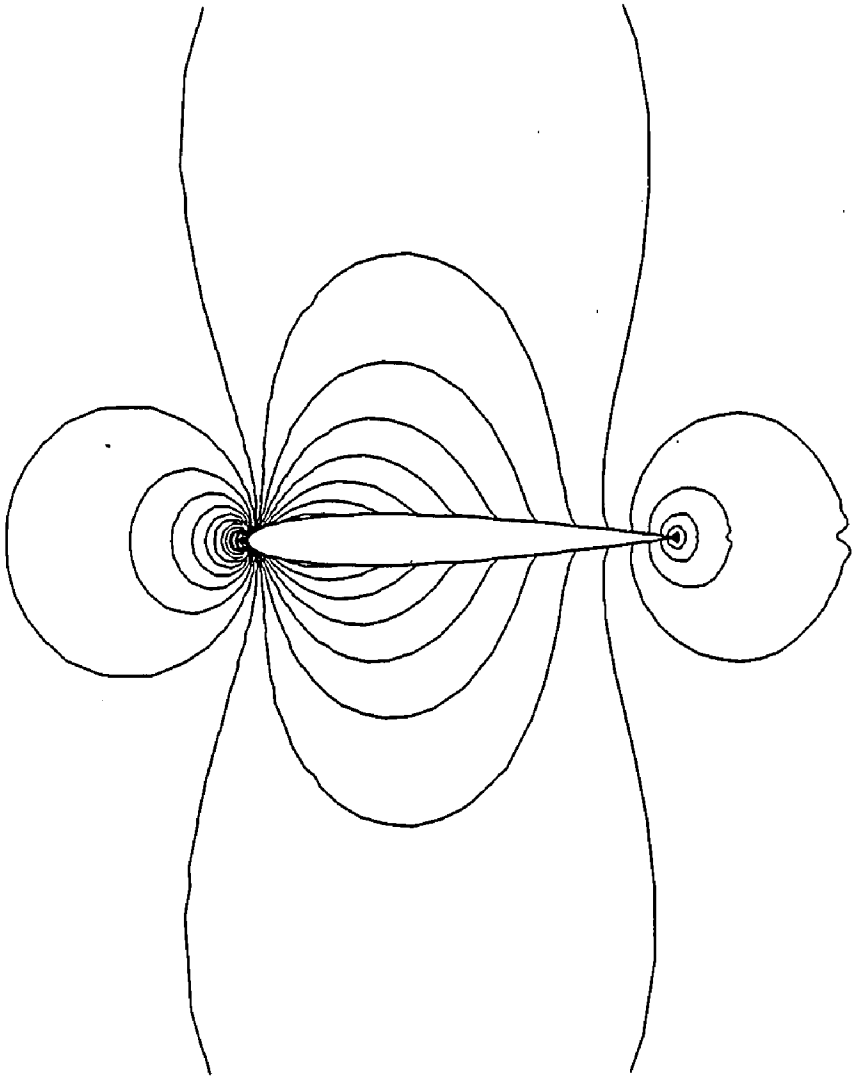

Figure 40: Mach Number Contour by Using the MOC Boundary Conditions.

conditions shown earlier. The 111 × 31 C-mesh grid system used is shown in Fig. 38. This grid is generated by the GRAPE code [45-46] which solves a system of elliptic equations as proposed by J. Thompson and developed by Sorenson. The thickness of cells on the airfoil is 0.2% of the chord length. Figure 39 and Figure 40 show the Mach contours as computed with explicit and MOC boundary conditions, respectively. The solution with explicit boundary conditions generates a boundary-layer-like Mach contour attached to the airfoil. Refining the grids near the airfoil can reduce the false boundary-layer region gradually. However, using MOC boundary conditions is the most effective way in this case, as demonstrates in Fig. 40 which shows the Mach contour by the MOC boundary conditions. Note that here we are merely referring the explicit boundary conditions to the set presented earlier. Actually, it is also possible to apply the MOC conditions explicitly and obtain exactly the same result as the implicit MOC conditions. However, the unfavorable time-lagging between the field points and the boundary points may still cause a convergence problem as discussed earlier.

4.6 Variational Implicit Schemes

The implicit scheme was shown to be effective in the previous section. However, as indicated earlier, the cost of inverting the left-hand-side matrices increases drastically as the block size expands. Accordingly, Algorithms with simplified inversion are of interest for systems involving more equations. In this section we shall discuss several approximate-factorization procedures other than the standard ADI.

4.6.1 Diagonal ADI Scheme

For steady state applications, or when first-order time accuracy is adequate, one can replace the left-hand-side operators by simplified matrices that maintain the stability properties of the original matrices. One approach used by Pulliam and

Chaussee [40] utilizes a similarity transformation to convert the block tridiagonal matrices into scalar tridiagonals. This idea has been applied to incompressible flows by Rogers, Chang and Kwak [47].

Starting with the finite-difference equations for a source-free hyperbolic system without artificial viscous terms, one has

$$(I + \Delta t \delta_\xi \mathbf{A}\cdot)\,(I + \Delta t \delta_\eta \mathbf{B}\cdot)\,\Delta\mathbf{Q} = RHS \tag{4.58}$$

The flux Jacobian matrices $\mathbf{A}$ and $\mathbf{B}$ each have real eigenvalues and a complete set of eigenvectors. Thus, they can be diagonalized by similarity transformation, (see Chapter 2)

$$\mathbf{\Lambda}_\xi = \mathbf{M}_\xi^{-1}\mathbf{A}\mathbf{M}_\xi, \qquad \mathbf{\Lambda}_\eta = \mathbf{M}_\eta^{-1}\mathbf{B}\mathbf{M}_\eta$$

or inversely,

$$\mathbf{A} = \mathbf{M}_\xi\mathbf{\Lambda}_\xi\mathbf{M}_\xi^{-1}, \qquad \mathbf{B} = \mathbf{M}_\eta\mathbf{\Lambda}_\eta\mathbf{M}_\eta^{-1}$$

By replacing the flux Jacobians $\mathbf{A}$ and $\mathbf{B}$ in Eq. (4.58) one gets

$$\left(\mathbf{M}_\xi\mathbf{M}_\xi^{-1} + \Delta t\delta_\xi\left(\mathbf{M}_\xi\mathbf{\Lambda}_\xi\mathbf{M}_\xi^{-1}\right)\cdot\right)\left(\mathbf{M}_\eta\mathbf{M}_\eta^{-1} + \Delta t\delta_\eta\left(\mathbf{M}_\eta\mathbf{\Lambda}_\eta\mathbf{M}_\eta^{-1}\right)\cdot\right)\Delta\mathbf{Q} = RHS \tag{4.59}$$

At this point Eq. (4.58) and Eq. (4.59) are equivalent. An useful approximation can be made by moving $\mathbf{M}_\xi$, $\mathbf{M}_\xi^{-1}$, $\mathbf{M}_\eta$ and $\mathbf{M}_\eta^{-1}$ outside the spatial derivative terms δ_ξ and δ_η. This modification reduces the time accuracy to at most first-order in time. The resulting equation is

$$\mathbf{M}_\xi\left(\mathbf{I} + \Delta t\delta_\xi\mathbf{\Lambda}_\xi\right)\mathbf{M}_\xi^{-1}\mathbf{M}_\eta\left(\mathbf{I} + \Delta t\delta_\eta\mathbf{\Lambda}_\eta\right)\mathbf{M}_\eta^{-1}\Delta\mathbf{Q} = RHS \tag{4.60}$$

The above modification is done to the left-hand-side of Eq. (4.58). The steady-state solution (the right-hand-side) remains unchanged. Thus, if the transient solution is not of interest, the diagonal algorithm gives exactly the same result as

the original scheme so long as it converges. Because the linear stability analysis assumes constant flux Jacobians and pulls them outside the spatial derivatives, it suggests that the diagonal algorithm gives exactly the same unconditional stability and convergence as the original algorithm.

The left-hand-side operator of Eq. (4.60) consists of three matrix-vector multiplications and inversions of m scalar tridiagonal systems. The total number of operations is thus $6m^2 + 6m$. Specifically for the incompressible system, the original scheme and the diagonal scheme need 164 and 72 operations, respectively, to invert the left-hand-side. The saving is even more significant for 4×4 systems.

Besides reduced computational time, there are other advantages to the diagonal algorithm. For example, it requires less temporary memory storage and simplifies coding for computer vectorization. Pulliam claims that the convergence rate can be enhanced with implicit 4th order viscosity added [25]. This may not be economical to be used with the original algorithm since the cost of block pentadiagonal solver could be higher than saving of iterations.

One drawback of diagonal algorithm is the difficulty encountered in applying the implicit MOC boundary conditions. As addressed earlier, incoming and outgoing information is treated differently for the MOC boundary procedure. The original algorithm extracts the outgoing information by taking one-sided differences of the flux-Jacobian derivatives on the left-hand-side. This procedure translates into one-sided differencing of the appropriate eigenvalues plus multiplications of $\mathbf{M}$ and $\mathbf{M}^{-1}$ for the diagonal algorithm. Hence the scalar tridiagonal structure is not destroyed by the outgoing characteristics. However, the incoming information contains specified boundary conditions, and some elements in the matrix $\partial\mathbf{\Omega}/\partial\mathbf{Q}$ may fall outside the diagonal line. This breaks down the scalar tridiagonal structure. Thus the diagonal algorithm is usually associated with the explicit boundary conditions, for it

retains the diagonal structure.

Comparisons of convergence rates between the original algorithm and diagonal algorithm is shown in Fig. 41 and Fig. 42. The geometry is again the bicircular-arc airfoil cascade case shown in Fig. 20. The steady state result is not presented since it is precisely identical for both algorithms. Explicit boundary condition is used in both cases. The optimal CFL number for the diagonal scheme is 9, which is smaller than the original algorithm. This is obviously caused by moving the flux Jacobians outside the differential operators. The convergence rates of two algorithms shown in Fig. 41 are based on the same CFL number. They converge at essentially the same rate. However, the CPU-time for each iteration of the diagonal algorithm is approximately 66% of the original algorithm on IBM-3090/180. A comparison of the optimal convergence rate based on the CPU-time of IBM-3090/180 (scalar machine) is shown in Fig. 42. The diagonal scheme is slightly faster in this case. It should become more favorable as the block size increases.

4.6.2 Lower-Upper Schemes

Other than the diagonal ADI scheme, one group of implicit scheme referred to as LU schemes are available as alternatives to the ADI scheme. These LU schemes are constructed by particular approximate-factorization procedures which result in implicit operators composed of only lower (L) and upper (U) block matrices. The procedure of inverting an LU operator resembles the Symmetric Successive Over Relaxation (SSOR) method [48] originally designed for elliptic equations. Such operators sometimes can be solved more efficiently than the block tridiagonal matrices because the LU decomposition is not required.

To assess the value of LU schemes, several interesting LU schemes are investigated and their efficiencies are compared with the ADI scheme. Central-difference is still used for the RHS operator, so that the final converged solutions are identical.

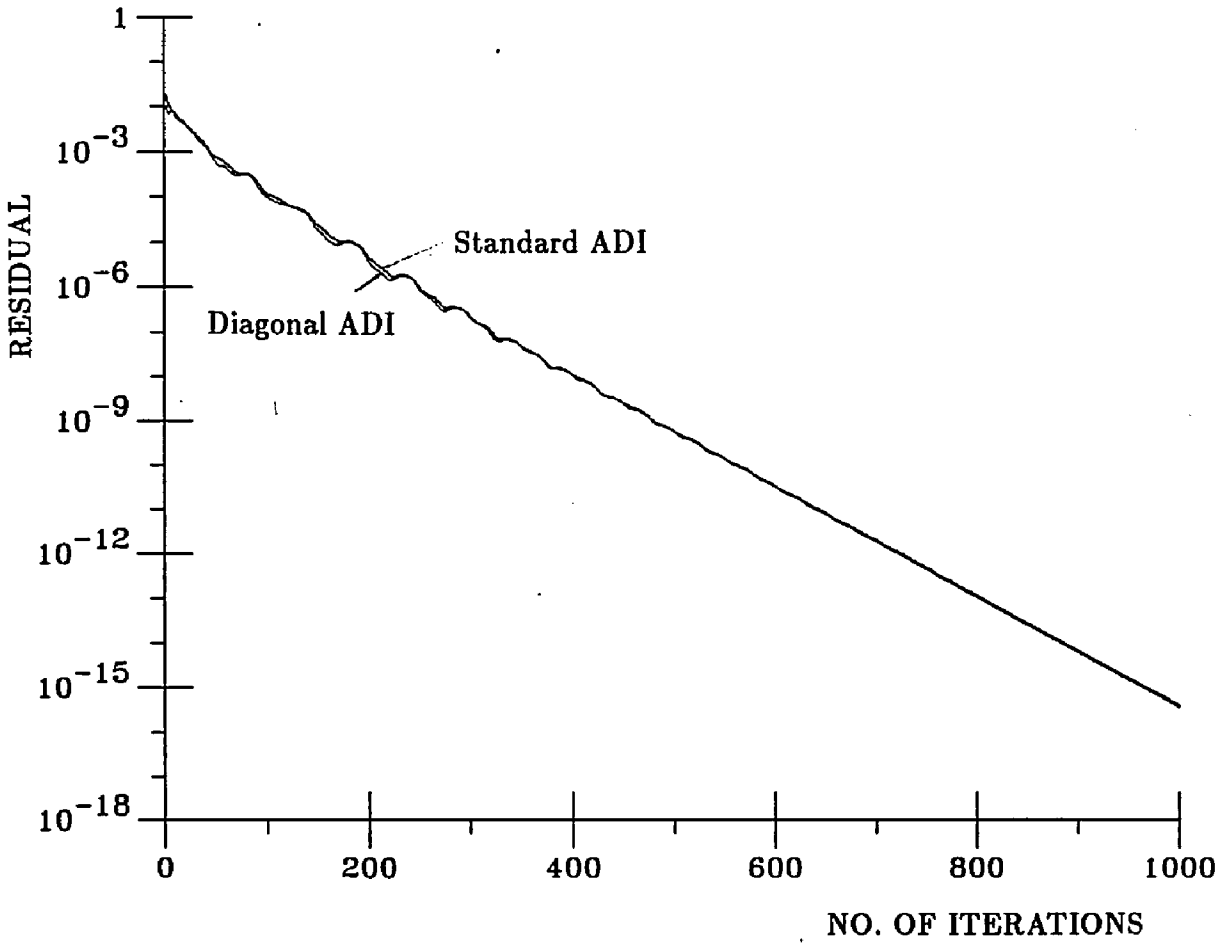

Figure 41: Convergence History of the Standard ADI and the Diagonal ADI Schemes. $CFL = 9$

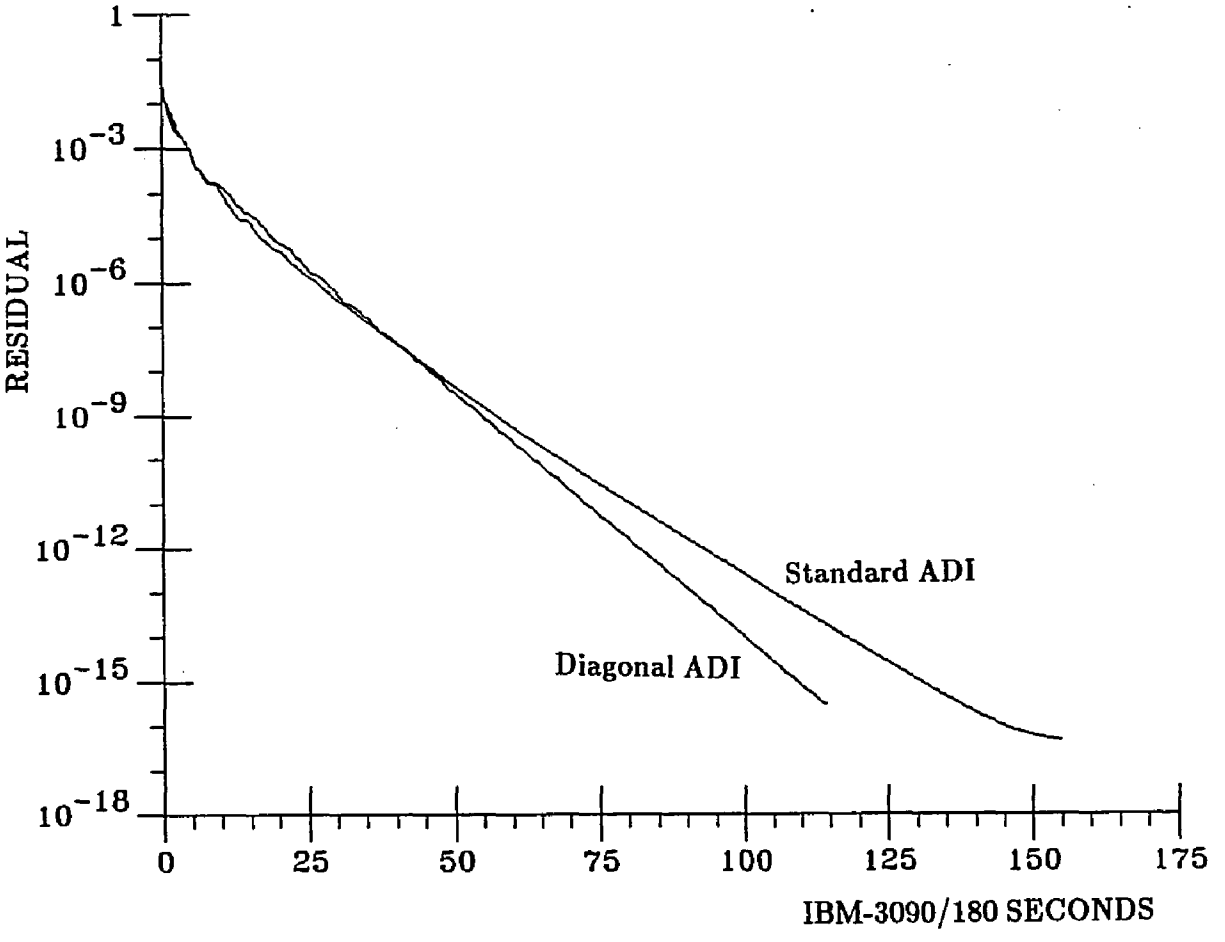

Figure 42: Scalar-Machine Cost Comparison of the Standard ADI and the Diagonal ADI Schemes. $CFL = 9$

The implicit operators of LU schemes are usually constructed on the basis of upwind difference. In the present study, the concept of flux-vector splitting originally proposed by Steger and Warming [49] is utilized to construct the upwind difference for the LHS operators. The concept of flux-vector splitting is described briefly.

A flux Jacobian $\mathbf{A}$ can be considered as a similarity transformation of a diagonal matrix, $\mathbf{\Lambda}$ (see Sec. 2.1),

$$\mathbf{A} = \mathbf{M}\mathbf{\Lambda}\mathbf{M}^{-1}$$

This matrix can be split into two matrices,

$$\mathbf{A} = \mathbf{M}(\mathbf{\Lambda}^+ + \mathbf{\Lambda}^-)\mathbf{M}^{-1} = \mathbf{M}\mathbf{\Lambda}^+\mathbf{M}^{-1} + \mathbf{M}\mathbf{\Lambda}^-\mathbf{M}^{-1} = \mathbf{A}^+ + \mathbf{A}^- \tag{4.61}$$

in which $\mathbf{\Lambda}^+$ contains only non-negative eigenvalues, and $\mathbf{\Lambda}^-$ contains only non-positive eigenvalues. The eigenvalues of $\mathbf{A}^+$ and $\mathbf{A}^-$ are the same as those of $\mathbf{\Lambda}^+$ and $\mathbf{\Lambda}^-$. Once the flux Jacobian $\mathbf{A}$ are split into $\mathbf{A}^+$ and $\mathbf{A}^-$, upwind difference can be utilized to formulate the LHS of an LU scheme.

In this section, three LU schemes are studied. They can be derived from the formulation for the fully implicit scheme. By splitting the flux Jacobians according to Eq. (4.61), the fully implicit scheme can be expressed as

$$(\mathbf{I} + \Delta t\delta_\xi \mathbf{A}^+ \cdot + \Delta t\delta_\xi \mathbf{A}^- \cdot + \Delta t\delta_\eta \mathbf{B}^+ \cdot + \Delta t\delta_\eta \mathbf{B}^- \cdot)\Delta\mathbf{Q} = RHS \tag{4.62}$$

In order to construct a stable LU scheme, upwind differences are used. In other words, backward differences are applied to $\mathbf{A}^+$ and $\mathbf{B}^+$, and forward differences are applied to $\mathbf{A}^-$ and $\mathbf{B}^-$ to give

$$(\mathbf{I} + \Delta t\delta_\xi^- \mathbf{A}^+ \cdot + \Delta t\delta_\xi^+ \mathbf{A}^- \cdot + \Delta t\delta_\eta^- \mathbf{B}^+ \cdot + \Delta t\delta_\eta^+ \mathbf{B}^- \cdot)\Delta\mathbf{Q} = RHS \tag{4.63}$$

where δ^- and δ^+ denote backward and forward differences, respectively. First-order differences are utilized throughout this section to simplify the LHS.

One LU scheme can be constructed directly from Eq. (4.63) by approximately factorizing the LHS into four factors to give

$$(\mathbf{I}+\Delta t\delta_{\xi}^{-}\mathbf{A}^{+}\cdot)(\mathbf{I}+\Delta t\delta_{\xi}^{+}\mathbf{A}^{-}\cdot)(\mathbf{I}+\Delta t\delta_{\eta}^{-}\mathbf{B}^{+}\cdot)(\mathbf{I}+\Delta t\delta_{\eta}^{+}\mathbf{B}^{-}\cdot)\Delta\mathbf{Q}=RHS \quad (4.64)$$

which may tentatively be called the four-factor LU scheme. This scheme has been used by Obayashi and Kuwahara [50] in solving the two-dimensional Navier-Stoke equations, and subsequently extended to three dimensions by Obayashi and Fujii [51]. Equation (4.64) can also be considered as splitting each of the two ADI operators into L and U factors, which are bidiagonal matrices if the first-order differences are taken. The price we pay for solving simpler operators is doubling the number of the factors in the LHS. The approximate-factorization error of the four-factor LU is of $O(\Delta t^4)$ as opposed to $O(\Delta t^2)$ of the ADI scheme. This may imply a more restrictive limit for the time-step size. Even less favorable stability criteria are expected in three dimensions where the four-factor LU scheme becomes a six-factor scheme.

In order to improve the stability characteristics, one could seek for other LU schemes which give lower order temporal error. Refer back to Eq. (4.63). A second LU scheme can be formulated by grouping the backward-difference operators into one factor and the forward-difference operators into another as

$$\left(\mathbf{I}+\Delta t\delta_{\xi}^{-}\mathbf{A}^{+}\cdot+\Delta t\delta_{\eta}^{-}\mathbf{B}^{+}\cdot\right)\left(\mathbf{I}+\Delta t\delta_{\xi}^{+}\mathbf{A}^{-}\cdot+\Delta t\delta_{\eta}^{+}\mathbf{B}^{-}\cdot\right)\Delta\mathbf{Q}=RHS \quad (4.65)$$

Analogous to the four-factor LU, this scheme is tentatively called the two-factor LU scheme. The temporal error of the two-factor LU scheme is of $O(\Delta t^2)$ and should allow a larger time-step size. Notice that in three dimensions, this scheme remains two-factored, and the temporal error is still of second order, while the approximate-factorization error of the ADI scheme increases to the third order.

The third LU scheme is formulated by an approximate-factorization procedure which, in an attempt to increase the weight of the diagonal blocks, results in a more stable scheme than the two-factor LU scheme. Starting from Eq. (5.63), one can perform the first-order differencing to obtain

$$\mathbf{I}+\Delta t\left(\mathbf{A}^{+}_{i,j}-\mathbf{A}^{-}_{i,j}+\mathbf{B}^{+}_{i,j}-\mathbf{B}^{-}_{i,j}\right)\Delta\mathbf{Q}_{i,j}$$

$$+\Delta t\left(-(\mathbf{A}^{+}\Delta\mathbf{Q})_{i-1,j}+(\mathbf{A}^{-}\Delta\mathbf{Q})_{i+1,j}-(\mathbf{B}^{+}\Delta\mathbf{Q})_{i,j-1}+(\mathbf{B}^{-}\Delta\mathbf{Q})_{i,j+1}\right)=RHS \tag{4.66}$$

in which the grid spacings, $\Delta\xi$ and $\Delta\eta$, are set to one. Equation (4.66) is approximately factorized to give

$$\left(\mathbf{I}+\Delta t\left(-\mathbf{A}^{-}-\mathbf{B}^{-}+\delta^{-}_{\xi}\mathbf{A}^{+}\cdot+\delta^{-}_{\eta}\mathbf{B}^{+}\cdot\right)\right)\cdot$$

$$\left(\mathbf{I}+\Delta t\left(\mathbf{A}^{-}+\mathbf{B}^{-}+\mathbf{A}^{+}+\mathbf{B}^{+}\right)\right)^{-1}\cdot$$

$$\left(\mathbf{I}+\Delta t\left(\mathbf{A}^{+}+\mathbf{B}^{+}+\delta^{+}_{\xi}\mathbf{A}^{-}\cdot+\delta^{+}_{\eta}\mathbf{B}^{-}\right)\right)\Delta Q=RHS \tag{4.67}$$

The LU scheme represented by Eq. (4.67) is usually referred to as the Diagonally Dominant ADI (DDADI) scheme [52]. Only two sweeps are involved in the DDADI, and the temporal error is of $O(\Delta t^2)$, even in three dimensions.

The three LU schemes are utilized to solve the incompressible Euler equations for the geometry shown in Fig. 20 in order to test their relative efficiencies and compare to the standard and diagonal ADI schemes. As before, we compare their convergence rates on two basis, the number of iterations and the CPU time. Comparisons of the convergence rates of the five schemes based on number of iterations is shown in Fig. 43. Corresponding results in terms of CPU time of CYBER-205 are shown in Fig. 44. The results shown are for the optimum CFL numbers of these schemes.

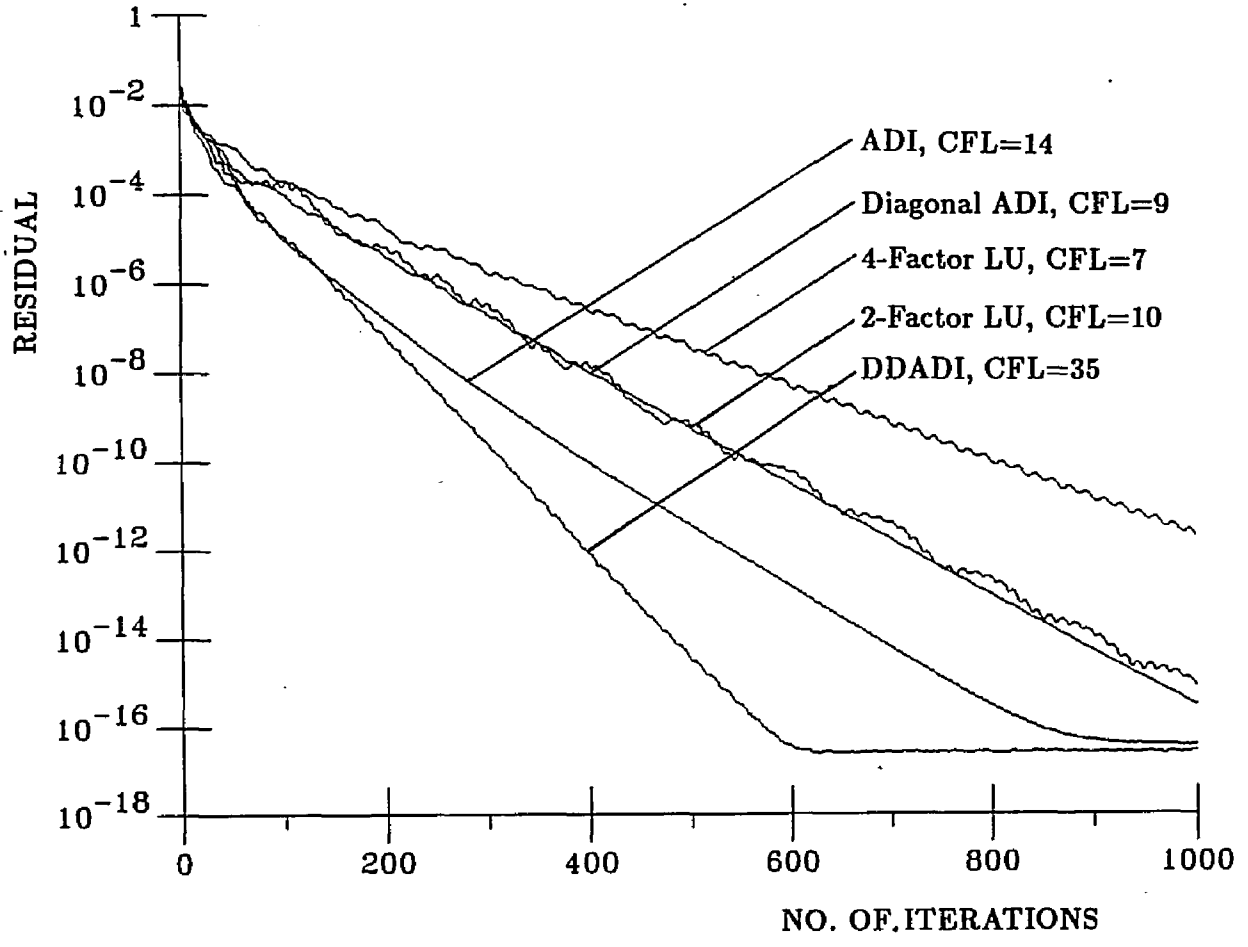

Figure 43: Comparison of Convergence Rates of Various Implicit Schemes.

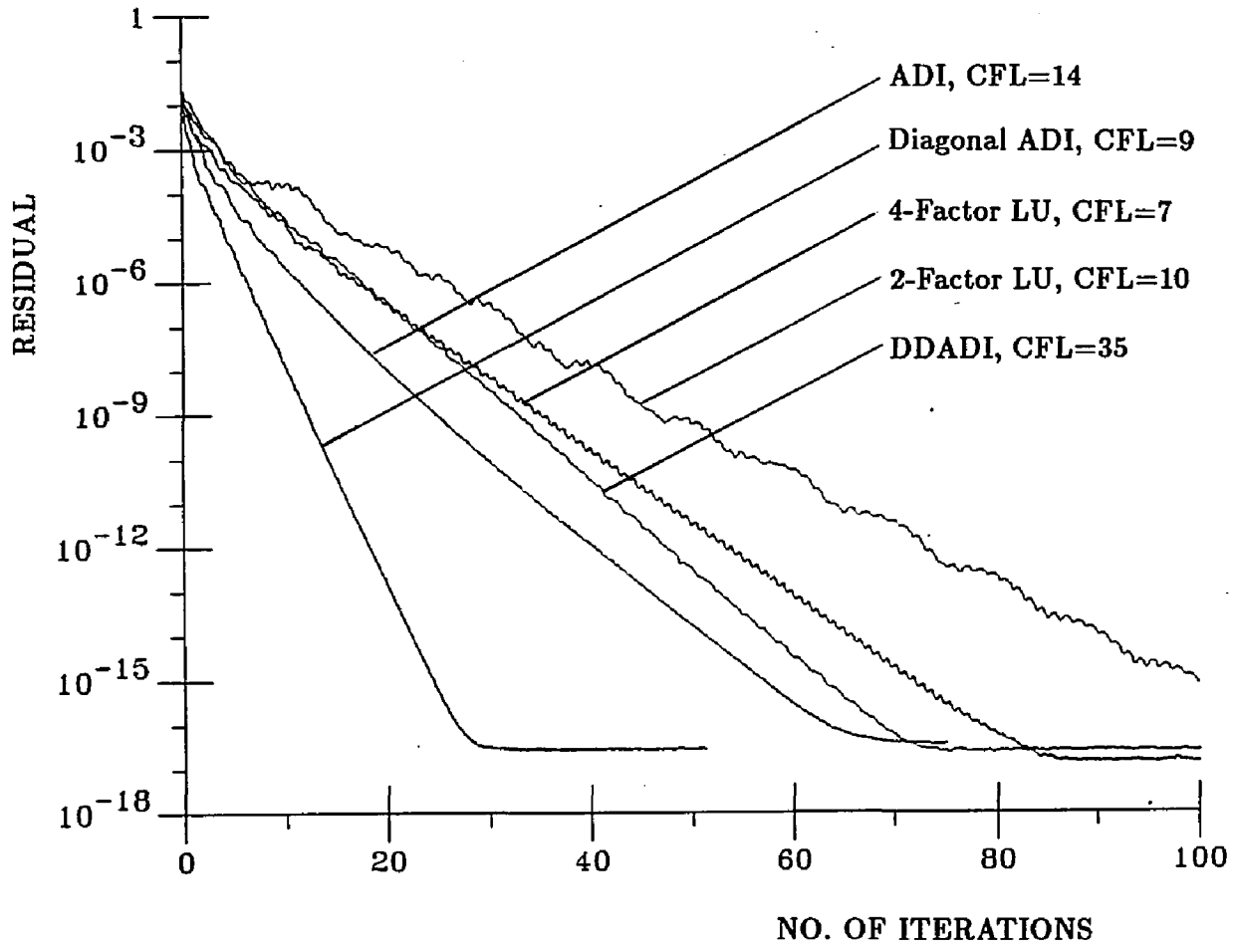

Figure 44: Comparison of CPU Cost of Various Implicit Schemes.

The optimum CFL numbers of the four-factor and two-factor LU schemes are lower than that of the ADI scheme, while the DDADI scheme has a substantially larger optimum CFL number than the ADI scheme. As shown in Fig. 43, the fastest convergence rate is given by the DDADI scheme, and the slowest by the four-factor scheme. The ranking of the convergence rate is almost determined by the optimum CFL number each scheme uses. This ranking is upset by the performances of these scheme on CYBER-205. Figure 44 shows the diagonal ADI scheme gives the fastest convergence rate in terms of CPU time, and the two-factor LU scheme is the slowest. In general, the two-factor and DDADI schemes requires more CPU time per iterations than the others. The major reason is the difficulty encountered in vectorizing these two schemes. The ADI, diagonal ADI, and the four-factor schemes can be vectorized in a similar fashion, which is more efficient than what the two-factor and DDADI schemes can achieve. A summary of the performances of these five schemes is given in Table. 1.

Table 1. Comparison of Performance of Various Implicit Schemes

scheme	ADI	diagonal ADI	4-factor	2-factor	DDADI
optimum CFL	14	9	7	10	35
convergence rate (order per 100 iterations)	1.7	1.25	0.85	1.2	2.4
CYBER205 cost (second per iteration)	0.075	0.026	0.052	0.100	0.120
vectorizability	medium	good	medium	poor	poor

CHAPTER 5

THREE-DIMENSIONAL PROBLEMS

Steady three-dimensional incompressible inviscid flows are considered in this chapter. Previous one- and two-dimensional calculations have given us considerable information concerning the relative efficiency of the implicit schemes and the explicit Runge-Kutta scheme. By and large, the implicit schemes dominate the Runge-Kutta scheme in one dimension, but compete with the Runge-Kutta scheme in two dimensions. Extrapolated to three dimensions, the Runge-Kutta scheme may be expected to be more efficient. Calculations are conducted to confirm this. To demonstrate the applicability of this work on engineering, flows passing through a propeller operating in a wake are solved by a suitable scheme. The analyses in this chapter are mainly aimed at this problem.

5.1 Governing Equations

For rotational propeller problems, it is more convenient to write the governing equations in cylindrical coordinates. Other coordinate systems may cause difficulty in specifying periodic boundary conditions. In the cylindrical coordinate system, the transient incompressible Euler equations can be written as

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{E}}{\partial r} + \frac{\partial \mathbf{F}}{r\partial \theta} + \frac{\partial \mathbf{G}}{\partial z} = \mathbf{H} \tag{5.1}$$

where

$$\mathbf{Q} = r\begin{pmatrix} p \\ u_r \\ u_\theta \\ u_z \end{pmatrix}, \mathbf{E} = r\begin{pmatrix} \beta u_r \\ u_r^2 + p \\ u_r u_\theta \\ u_r u_z \end{pmatrix}, \mathbf{F} = r\begin{pmatrix} \beta u_\theta \\ u_\theta u_r \\ u_\theta^2 + p \\ u_\theta u_z \end{pmatrix}, \mathbf{G} = r\begin{pmatrix} \beta u_z \\ u_z u_r \\ u_z u_\theta \\ u_z^2 + p \end{pmatrix}$$

$$\mathbf{H} = r\begin{pmatrix} 0 \\ (u_\theta^2 + p)/r + \omega^2 r + 2\omega u_\theta \\ -2\omega u_r - u_r u_\theta / r \\ 0 \end{pmatrix}$$

As before, a pseudocompressibility term is added to the continuity equation. For generality, the coordinate system is assumed to rotate around the z-axis with an angular velocity ω.

The right-hand-side of Eq. (5.1), **H**, represents the source vector. In the source vector, $(u_\theta^2 + p)$ and $-u_r u_\theta$ come from the curvature of the θ-axis. The centrifugal force is represented by $\omega^2 r^2$, and the Coriolis force is represented by $2\omega u_\theta r$ and $-2\omega u_r r$. The centrifugal force and the Coriolis force are the results of rotation of the coordinate system.

The flux Jacobians and source Jacobian for Eq. (5.1) are

$$\mathbf{A} = \frac{\partial \mathbf{E}}{\partial \mathbf{Q}} = \begin{pmatrix} 0 & \beta & 0 & 0 \\ 1 & 2u_r & 0 & 0 \\ 0 & u_\theta & u_r & 0 \\ 0 & u_z & 0 & u_r \end{pmatrix}, \qquad \mathbf{B} = \frac{\partial \mathbf{F}}{\partial \mathbf{Q}} = \begin{pmatrix} 0 & 0 & \beta & 0 \\ 0 & u_\theta & u_r & 0 \\ 1 & 0 & 2u_\theta & 0 \\ 0 & 0 & u_z & u_\theta \end{pmatrix}$$

$$\mathbf{C} = \frac{\partial \mathbf{G}}{\partial \mathbf{Q}} = \begin{pmatrix} 0 & 0 & 0 & \beta \\ 0 & u_z & 0 & u_r \\ 0 & 0 & u_z & u_\theta \\ 1 & 0 & 0 & 2u_z \end{pmatrix}, \quad \mathbf{D} = \frac{\partial \mathbf{H}}{\partial \mathbf{Q}} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ \frac{1}{r} & 0 & 2\omega + \frac{2u_\theta}{r} & 0 \\ 0 & -2\omega - \frac{u_\theta}{r} & -\frac{u_r}{r} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

The eigenvalues of the flux Jacobians are

$$\lambda_{1,2,3,4}(\mathbf{A}) = u_r, u_r, u_r \pm \sqrt{u_r^2 + \beta}$$

$$\lambda_{1,2,3,4}(\mathbf{B}) = u_\theta, u_\theta, u_\theta \pm \sqrt{u_\theta^2 + \beta}$$

$$\lambda_{1,2,3,4}(\mathbf{C}) = u_z, u_z, u_z \pm \sqrt{u_z^2 + \beta}$$

By transforming to (ξ, η, ζ) coordinates, one gets the governing equations in the transformed coordinate system (see Appendix),

$$\frac{\partial \hat{\mathbf{Q}}}{\partial t} + \frac{\partial \hat{\mathbf{E}}}{\partial \xi} + \frac{\partial \hat{\mathbf{F}}}{\partial \eta} + \frac{\partial \hat{\mathbf{G}}}{\partial \zeta} = \hat{\mathbf{H}} \tag{5.2}$$

where

$$\hat{\mathbf{Q}} = \frac{1}{J}\mathbf{Q}, \qquad \hat{\mathbf{E}} = \frac{1}{J}\begin{pmatrix} \beta U \\ u_r U + \xi_r p \\ u_\theta U + (\xi_\theta / r) p \\ u_z U + \xi_z p \end{pmatrix}, \qquad \hat{\mathbf{H}} = \frac{1}{J}\mathbf{H}$$

and

$$\hat{\mathbf{A}} = \begin{pmatrix} 0 & \beta\xi_r & \beta\xi_\theta/r & \beta\xi_z \\ \xi_r & U+\xi_r u_r & (\xi_\theta/r)u_r & \xi_z u_r \\ \xi_\theta/r & \xi_r u_\theta & U+(\xi_\theta/r)u_\theta & \xi_z u_\theta \\ \xi_z & \xi_z u_z & (\xi_\theta/r)u_z & U+\xi_z u_z \end{pmatrix}, \qquad \hat{\mathbf{D}} = \mathbf{D}$$

The contravariant velocity, U, is defined as

$$U = u_r\xi_r + u_\theta\frac{\xi_\theta}{r} + u_z\xi_z$$

and the Jacobian of transformation, J, is defined as

$$J = \frac{\partial(\xi,\eta,\zeta)}{r\partial(r,\theta,z)}$$

The rest of the flux vectors, $\hat{\mathbf{F}}$ and $\hat{\mathbf{G}}$, can be obtained by replacing ξ in $\hat{\mathbf{E}}$ by η and ζ, respectively, and so are the flux Jacobians $\hat{\mathbf{B}}$ and $\hat{\mathbf{C}}$.

The eigenvalues of matrix $\hat{\mathbf{A}}$ are given as

$$\lambda_{1,2,3,4}(\hat{\mathbf{A}}) = U, U, U \pm \sqrt{U^2 + \beta[\xi_r^2 + (\xi_\theta/r)^2 + \xi_z^2]}$$

which are similar to those of two-dimensional system except for the presence of two identical eigenvalues. This causes difficulties in deriving the eigenmatrices for MOC boundary conditions [47]. The derivation will be detailed later on.

For solving the propeller flows, two rotational speeds of the coordinate system are of interest. When $\omega = 0$, the coordinate frame is stationary, and the fluid motion is observed from 'absolute coordinates'. When $\omega = \Omega$, where Ω is the rotational speed of the propeller, the fluid motion is described by 'relative coordinates'. Observed from the absolute coordinates, the propeller is rotating at an angular velocity Ω, whereas from the relative coordinates, the propeller appears motionless.

The propeller-flow problems can be solved either by the relative coordinates or by the absolute coordinates. Since we attach a body-fitted grid to the propeller, the motion of the grid depends on the coordinate system on which the observer stands.

In other words, the grid appears stationary on the relative coordinate system, and rotating on the absolute coordinate system. In case the inflow is axisymmetric, both coordinate systems can have steady state solutions if we use the cylindrical coordinates. Steady state solutions do not exist for some absolute coordinate systems such as the Cartesian system. In principle, both system should result in the same steady state solutions. The only difference is how the steady solutions are approached.

Comparing the governing equations in the absolute and relative systems reveals that they give different source vectors. In the cylindrical coordinates, the circumferential velocity component is the only dependent variable that is different for the two systems. They can be related by

$$u_{\theta a} = u_{\theta r} + \Omega r \tag{5.3}$$

where $u_{\theta a}$ and $u_{\theta r}$ are the u_θ in the absolute and relative coordinates, respectively. The source vector in the absolute coordinates is then

$$\mathbf{H}_a = \begin{pmatrix} 0 \\ u_{\theta a}^2 + p \\ -u_r u_{\theta a} \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ u_{\theta r}^2 + p + (\omega r)^2 + 2\omega r u_{\theta r} \\ -\omega r u_r - u_r u_{\theta r} \\ 0 \end{pmatrix} = \mathbf{H}_r + \begin{pmatrix} 0 \\ 0 \\ \omega r u_r \\ 0 \end{pmatrix} \tag{5.4}$$

in which $\mathbf{H}_a$ and $\mathbf{H}_r$ represent the source vectors in the absolute and relative coordinates. Equation (5.4) shows the source vectors in the two coordinate systems differ by $\omega r u_r$ in the third row. Put it in another word, half of the coriolis force in the θ direction has been absorbed into the flux vector terms by using the absolute coordinates. At this point of time we still can not indicate the advantage or disadvantage of using either of the two coordinate systems. Vector stability analysis will be performed to give better insight.

5.2 Implicit Scheme

The two-factor ADI scheme for two dimensions has been proven to be effective in the previous chapter. Extension to a three-factor ADI scheme for three dimensions is straightforward. Just as Eq. (4.13) is used to solve the two-dimensional Euler equations, the three-dimensional Euler equations are similarly solved by

$$\left(I - \Delta t\mathbf{D} + \Delta t\delta_\xi \mathbf{A}\cdot - \frac{\epsilon_i}{8}\frac{1}{J}\delta_{\xi\xi}J\cdot\right)(I - \Delta t\mathbf{D})^{-1}\cdot$$
$$\left(I - \Delta t\mathbf{D} + \Delta t\delta_\eta \mathbf{B}\cdot - \frac{\epsilon_i}{8}\frac{1}{J}\delta_{\eta\eta}J\cdot\right)(I - \Delta t\mathbf{D})^{-1}\cdot$$
$$\left(I - \Delta t\mathbf{D} + \Delta t\delta_\varsigma \mathbf{C}\cdot - \frac{\epsilon_i}{8}\frac{1}{J}\delta_{\varsigma\varsigma}J\cdot\right)\Delta\mathbf{Q}$$
$$= -\Delta t\left(\delta_\xi \mathbf{E} + \delta_\eta \mathbf{F} + \delta_\varsigma \mathbf{G} - \mathbf{H}\right) - \frac{\epsilon_e}{8}\frac{1}{J}(\delta_{\xi\xi\xi\xi} + \delta_{\eta\eta\eta\eta} + \delta_{\varsigma\varsigma\varsigma\varsigma})J\mathbf{Q}^n \qquad (5.5)$$

where the time step, Δt, can be determined in the similar fashion as two dimensions,

$$\Delta t = CFL/\sqrt{\lambda^2_{\xi_{max}} + \lambda^2_{\eta_{max}} + \lambda^2_{\varsigma_{max}}} \qquad (5.6)$$

As before, the 'hats' are dropped for convenience. This three-factor ADI scheme will be shown to be conditionally stable later on. In contrast, the two-factor ADI scheme for the two-dimensional Euler equations is unconditionally stable. The major difference which causes this instability is the additional $O(\Delta t^3)$ temporal error term introduced by the three factors.

5.2.1 Scalar Stability Analysis

To understand the stability characteristics of the three-factor ADI scheme, consider the following three-dimensional, scalar, linear, hyperbolic equation.

$$\frac{\partial u}{\partial t} + \lambda_\xi \frac{\partial u}{\partial \xi} + \lambda_\eta \frac{\partial u}{\partial \eta} + \lambda_\varsigma \frac{\partial u}{\partial \varsigma} = 0 \qquad (5.7)$$

Equation (5.7) is solved by three-factor ADI scheme as follows

$$\left(1 + \Delta t\lambda_\xi \delta_\xi - \frac{\epsilon_i}{8}\delta_{\xi\xi}\right)\left(1 + \Delta t\lambda_\eta \delta_\eta - \frac{\epsilon_i}{8}\delta_{\eta\eta}\right)\left(1 + \Delta t\lambda_\varsigma \delta_\varsigma - \frac{\epsilon_i}{8}\delta_{\varsigma\varsigma}\right)\Delta u$$

$$= -\Delta t \left(\lambda_\xi \delta_\xi + \lambda_\eta \delta_\eta + \lambda_\varsigma \delta_\varsigma\right) u^n - \frac{\epsilon_e}{8} \left(\delta_{\xi\xi\xi\xi} + \delta_{\eta\eta\eta\eta} + \delta_{\varsigma\varsigma\varsigma\varsigma}\right) u^n \qquad (5.8)$$

in which the implicit and explicit artificial viscous terms are added.

To simplify the algebra, symmetric analysis is assumed as before,

$$CFL_\xi = CFL_\eta = CFL_\varsigma = CFL/\sqrt{3}$$

according to Eq. (5.6), and

$$\omega_\xi \Delta\xi = \omega_\eta \Delta\eta = \omega_\varsigma \Delta\varsigma = \theta$$

This symmetric analysis is equivalent to finding the amplification factor of the diagonal line between $(0,0,0)$ and (π,π,π) on the wavenumber domain. This can simplify the algebra dramatically, and the most unstable waves are almost always included in such analysis. The magnitude of the amplification factor thus obtained is rather complex, and is given as

$$|g|^2 = \frac{\text{numerator}}{\text{denominator}} \qquad (5.9)$$

where

$$\text{numerator} = \left\{ \left[1 + \frac{\epsilon_i}{4}(1-C)\right]^3 - \left[1 + \frac{\epsilon_i}{4}(1-C)\right] CFL^2 S^2 - \frac{3}{2}\epsilon_e (1-C)^2 \right\}^2 + \left\{ \sqrt{3}\left[1 + \frac{\epsilon_i}{4}(1-C)\right]^2 CFL\, S - CFL^3 S^3/3\sqrt{3} - \sqrt{3} CFL\, S \right\}^2$$

$$\text{denominator} = \left\{ \left[1 + \frac{\epsilon_i}{4}(1-C)\right]^3 - \left[1 + \frac{\epsilon_i}{4}(1-C)\right] CFL^2 S^2 \right\}^2 + \left\{ \sqrt{3}\left[1 + \frac{\epsilon_i}{4}(1-C)\right]^2 CFL\, S - CFL^3 S^3/3\sqrt{3} \right\}^2$$

in which S and C represent sine and cosine of the wavenumber, θ. Since the numerator and denominator are both positive, the criterion for stability is

$$\text{numerator} \leq \text{denominator}$$

This criterion results in

$$-3\epsilon_e(1-C)^2\left[1+\frac{\epsilon_i}{4}(1-C)\right]^3+3\epsilon_e(1-C)^2\left[1+\frac{\epsilon_i}{4}(1-C)\right]CFL^2S^2+\frac{9}{4}\epsilon_e^2(1-C)^4$$

$$-6\left[1+\frac{\epsilon_i}{4}(1-C)\right]^2CFL^2S^2+\frac{2}{3}CFL^4S^4+3CFL^2S^2\leq 0 \qquad (5.10)$$

Note that Eq. (5.10) contains four parameters, CFL, ϵ_e, ϵ_i, and θ.

First consider the case without any implicit or explicit viscosity. In such case, the scheme is stable if

$$\frac{2}{3}CFL^4S^4-3CFL^2S^2\leq 0$$

or

$$CFL^2S^2\leq\frac{9}{2} \qquad (5.11)$$

which indicates that CFL limit is $\sqrt{4.5}$, or approximately 2.12, and that the mid-wavenumber wave is the most unstable mode. Notice this result holds only when $CFL\,S$ is non-zero. Zero $CFL\,S$ represents the highest and the lowest-wavenumber waves. These two wave modes are always neutrally stable when no viscosity is added.

Next consider the effect of explicit viscosity. When $\epsilon_i = 0$, Eq. (5.10) becomes

$$\frac{2}{3}CFL^4S^4-3CFL^2S^2-3\epsilon_e(1-C)^2+3CFL^2S^2\epsilon_e(1-C)^2+\frac{9}{4}\epsilon_e^2(1-C)^2\leq 0 \quad (5.12)$$

Five terms occur in Eq. (5.12). The first two terms have been seen in the analysis for zero artificial viscosity. The remaining three terms are from the explicit artificial viscosity. The third term is negative and the fourth and fifth terms are positive. Negative terms have a stablizing effect; positive terms have a destablizing effect. Among these three terms, the fourth is the largest for mid-wavenumber wave ($S = 1$). At this wavenumber, the fourth term is larger than the third unless CFL is less than 1 which is too small to be interesting. Even if the implicit viscosity is added, the explicit viscosity is still harmful to the stability of the mid-wavenumber

wave. However, small amount of explicit viscosity could be needed to remove the odd-even decoupling.

The presence of implicit viscosity can increase stability. To demonstrate this, consider the case with the implicit viscosity but without the explicit viscosity. Equation (5.10) then becomes

$$-6\left[1+\frac{\epsilon_i}{4}(1-C)\right]^2+\frac{2}{3}CFL^2S^2+3\leq 0 \tag{5.13}$$

Again, Eq. (5.13) holds for non-zero CFL S only, since CFL^2S^2 has been divided through. The implicit viscosity has no effect on stability when CFL S and ϵ_e are both zero. In other words, the implicit viscosity alone can not remove the odd-even decoupling. Away from the lowest- and highest-wavenumber, the effect of implicit viscosity is included in the first term of Eq. (5.13). It is always negative and thus stablizing. The left-hand-side of Eq. (5.13) can be differentiated once to obtain the wavenumber of the most unstable wave (i.e., the wavenumber with maximum $|g|$). This wavenumber satisfies

$$C=\frac{\frac{9}{4}\epsilon_i^2+9\epsilon_i}{\frac{9}{4}\epsilon_i^2+4CFL^2}$$

or

$$S=\frac{\sqrt{16CFL^4+18\epsilon_i^2CFL^2-\frac{81}{2}\epsilon_i^3-81\epsilon_i^2}}{\frac{9}{4}\epsilon_i^2+4CFL^2}$$

This wavenumber is $\pi/2$ when ϵ_i is zero, and approaches 0 as ϵ_i increases. Substituting C and S into Eq. (5.13) gives

$$32CFL^6+(-144-144\epsilon_i+18\epsilon_i^2)CFL^4-81\epsilon_i^3CFL^2+\frac{729}{16}\epsilon_i^4=0 \tag{5.14}$$

Notice that only equal sign is considered in Eq. (5.14) because only the most unstable wave is under investigation. For a given ϵ_i, maximum CFL for stability can be found by solving Eq. (5.14). As ϵ_i approaches infinity, Eq. (5.14) approaches

$$18CFL^4-81\epsilon_iCFL^2+\frac{729}{16}\epsilon_i^2=0 \tag{5.15}$$

It is obvious that the relationship between CFL^2 and ϵ_i approaches linear when ϵ_i approaches infinity. The curve of maximum CFL versus ϵ_i is shown in Fig. 45.

In summary, the scalar stability analysis of three-factor ADI scheme results in the following conclusions:

(1) Maximum CFL for stability is $\sqrt{4.5}$ without artificial viscosity.

(2) Explicit viscosity can remove odd-even decoupling, but has negative effect to mid-wavenumber waves for CFL of interest.

(3) Implicit viscosity stabilize mid-wavenumber waves, but can not remove odd-even decoupling.

(4) Ideal amplification factor can be obtained by adding both viscosities.

These conclusions are verified by Fig. 46 which shows the amplification factor for various combinations of the implicit and explicit artificial viscosities. Four curves are shown in Fig. 46, all for $CFL = \sqrt{4.5}$. The curve for $\epsilon_i = \epsilon_e = 0$ shows neutral stability at mid- and highest-wavenumber. Adding the explicit viscosity results in instability as shown by the curve for $\epsilon_e = 0.2$ and $\epsilon_i = 0$. The explicit viscosity has the effect of damping high-wavenumber waves and destablizing mid-wavenumber waves. In contrast, the implicit viscosity stabilizes mid-wavenumber waves and leaves high-wavenumber waves neutrally stable, as shown by the curve for $\epsilon_e = 0$ and $\epsilon_i = 1$. An ideal combination is shown by the amplification factor for $\epsilon_e = 0.2$ and $\epsilon_i = 1$.

Recall that the above results are for a linear scalar equation, Eq. (5.7). The stability characteristics of Eq. (5.2) for the three-factor ADI differ from the above in that Eq. (5.2) has more than one eigenvalue in each direction, and that it has source terms. However, except for the source term effect, the stability of a vector system is usually qualitatively the same as that of a scalar equation. To understand the stability of Eq. (5.2), vector stability analysis is invoked. The following section

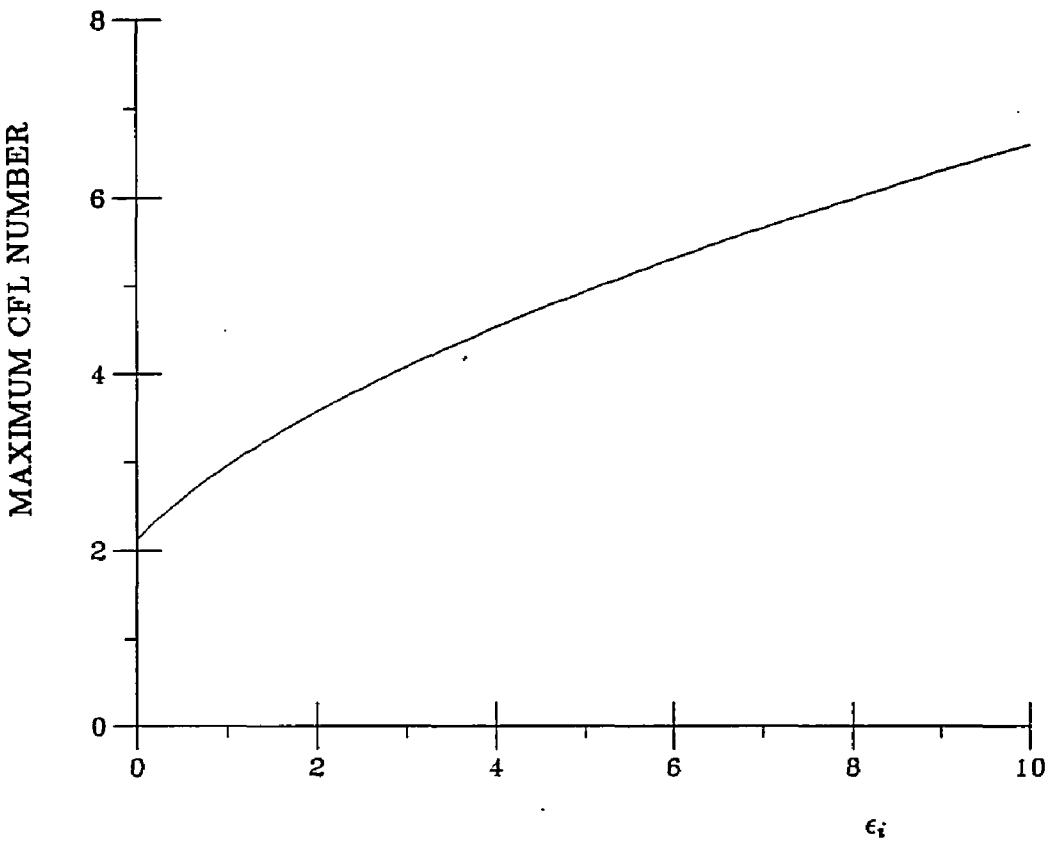

Figure 45: Maximum CFL Number of Three-Dimensional ADI Scheme. No Explicit Viscosity.

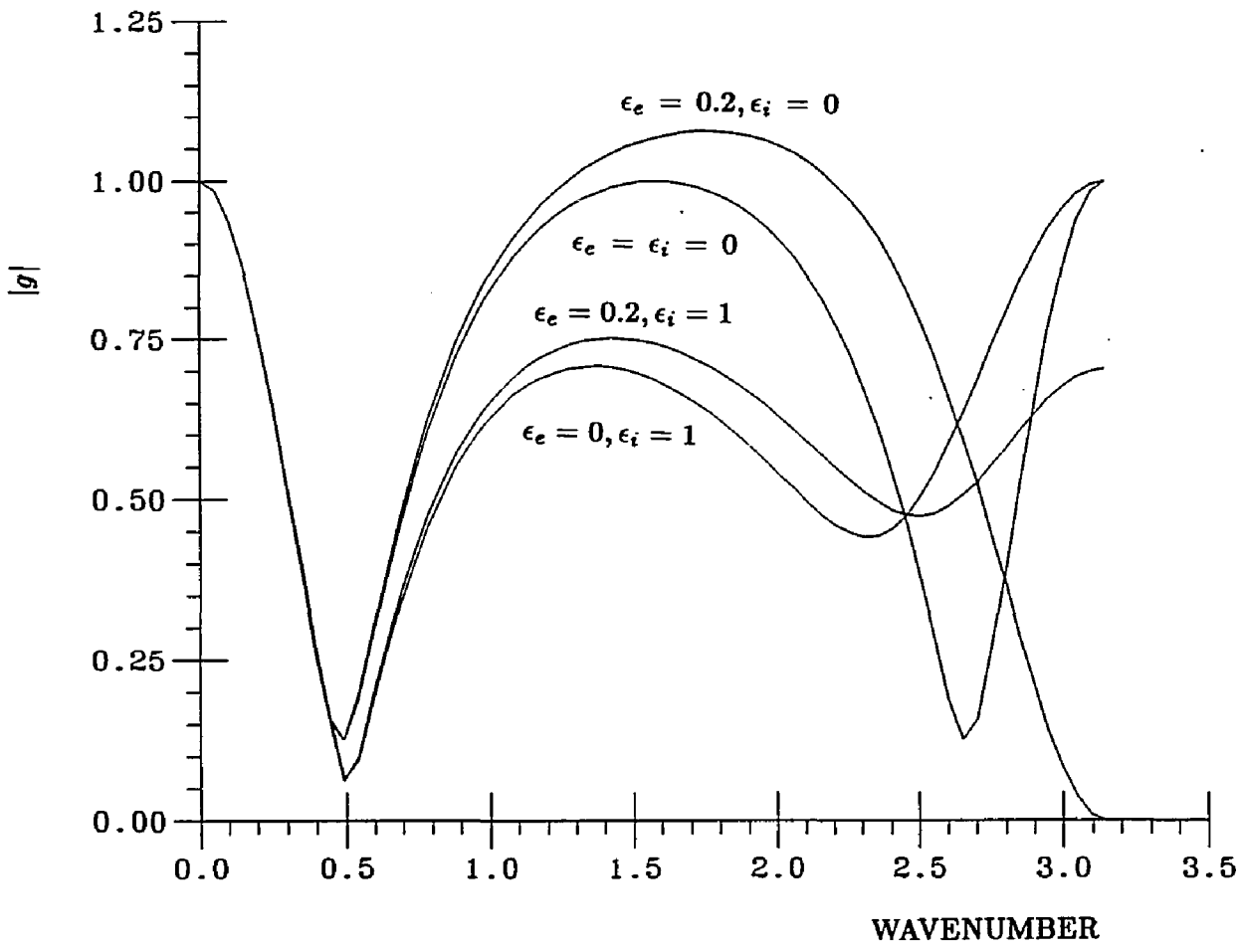

Figure 46: Amplification Factor of Three-Dimensional ADI Scheme for Scalar Hyperbolic Equation. $CFL = \sqrt{4.5}$.

addresses this in detail.

5.2.2 Vector Stability Analysis

For studying the stability characteristics of a system of equations, both the vector and the scalar analyses are important. There is no doubt that the vector stability analysis on the particular system is more realistic than the scalar analysis on a mathematically similar equation. However, due to its mathematical complexity, the vector stability analysis usually can only be performed on a computer case by case. Specific conclusions such as those drawn from the scalar analysis in the previous section are quite difficult to obtain from the vector analysis. The philosophy adopted here is that the scalar analysis provides the guild lines for stability and that the results should be confirmed by the vector analysis. Also, the effect of the source terms can not be predicted in the scalar analysis and has to be analyzed in the vector analysis.

To perform vector stability analysis, the amplification matrix has to be found. The amplification matrix for Eq. (5.5) can be obtained by

$$\mathbf{L}_{\text{imp}}(\mathbf{G}-\mathbf{I}) = \mathbf{L}_{\text{exp}}$$

or more specifically

$$\mathbf{G} = \mathbf{I} + \mathbf{L}_{\text{imp}}^{-1}\mathbf{L}_{\text{exp}} \tag{5.16}$$

where $\mathbf{L}_{\text{imp}}$ is the implicit operator and $\mathbf{L}_{\text{exp}}$ is the explicit operator in the Fourier domain. $\mathbf{L}_{\text{imp}}$ contains five operators,

$$\mathbf{L}_{\text{imp}} = \mathbf{L}_\xi(\mathbf{I}-\Delta t\mathbf{D})^{-1}\mathbf{L}_\eta(\mathbf{I}-\Delta t\mathbf{D})^{-1}\mathbf{L}_\varsigma$$

where

$$\mathbf{L}_\xi = \left(\mathbf{I} - \Delta t\mathbf{D} + i\Delta t\mathbf{A}S_\xi + \frac{\epsilon_i^{(2)}}{4}(1-C_\xi)\mathbf{I}\right)$$

$$\mathbf{L}_\eta = \left(\mathbf{I} - \Delta t\mathbf{D} + i\Delta t\mathbf{B}S_\eta + \frac{\epsilon_i^{(2)}}{4}(1 - C_\eta)\mathbf{I}\right)$$

$$\mathbf{L}_\varsigma = \left(\mathbf{I} - \Delta t\mathbf{D} + i\Delta t\mathbf{C}S_\varsigma + \frac{\epsilon_i^{(2)}}{4}(1 - C_\varsigma)\mathbf{I}\right)$$

and the explicit operator, $\mathbf{L}_{\text{exp}}$, is

$$\mathbf{L}_{\text{exp}} = \mathbf{D} - i\Delta t\left(\mathbf{A}S_\xi + \mathbf{B}S_\eta + \mathbf{C}S_\varsigma\right) - \frac{\epsilon_e^{(4)}}{2}\left[(1 - C_\xi)^2 + (1 - C_\eta)^2 + (1 - C_\varsigma)^2\right]$$

The stability is determined by the magnitudes of eigenvalues of matrix G. The spectral radius must be less or equal to one to ensure stability.

Equation (5.16) is used to study the stability characteristics of the three-factor ADI scheme. Consider a cell located at $r = 1$ with its faces perpendicular to three axes, r, θ, and z. In such case, all metric coefficients are zero except ξ_r, η_θ, and ς_z. We further assume the grid spacing to be

$$\Delta r = 0.1, \qquad r\Delta\theta = 0.1, \qquad \Delta z = 0.2 \tag{5.17}$$

and the absolute velocity to be

$$u_r = 0.1, \qquad u_\theta = 0.3, \qquad u_z = 1.0 \tag{5.18}$$

which are typical in propeller calculations.

First consider the case with $\omega = \Omega = 0$, where ω and Ω are the rotational speeds of the coordinates and the body, respectively. Notice that this does not imply the absence of source terms. The source vector still contains terms due to the curvature of the θ-coordinate. The amplification-factor contours for selected ς-wavenumbers are shown in Fig. 47 for $CFL = 4$ with no artificial viscosity added ($\epsilon_e = \epsilon_i = 0$). Here the amplification factor is defined as the largest magnitude of the eigenvalues of matrix G, and the CFL is defined by Eq. (5.6). For this case, the three-factor ADI scheme is stable. However, the mid-wavenumber wave at the

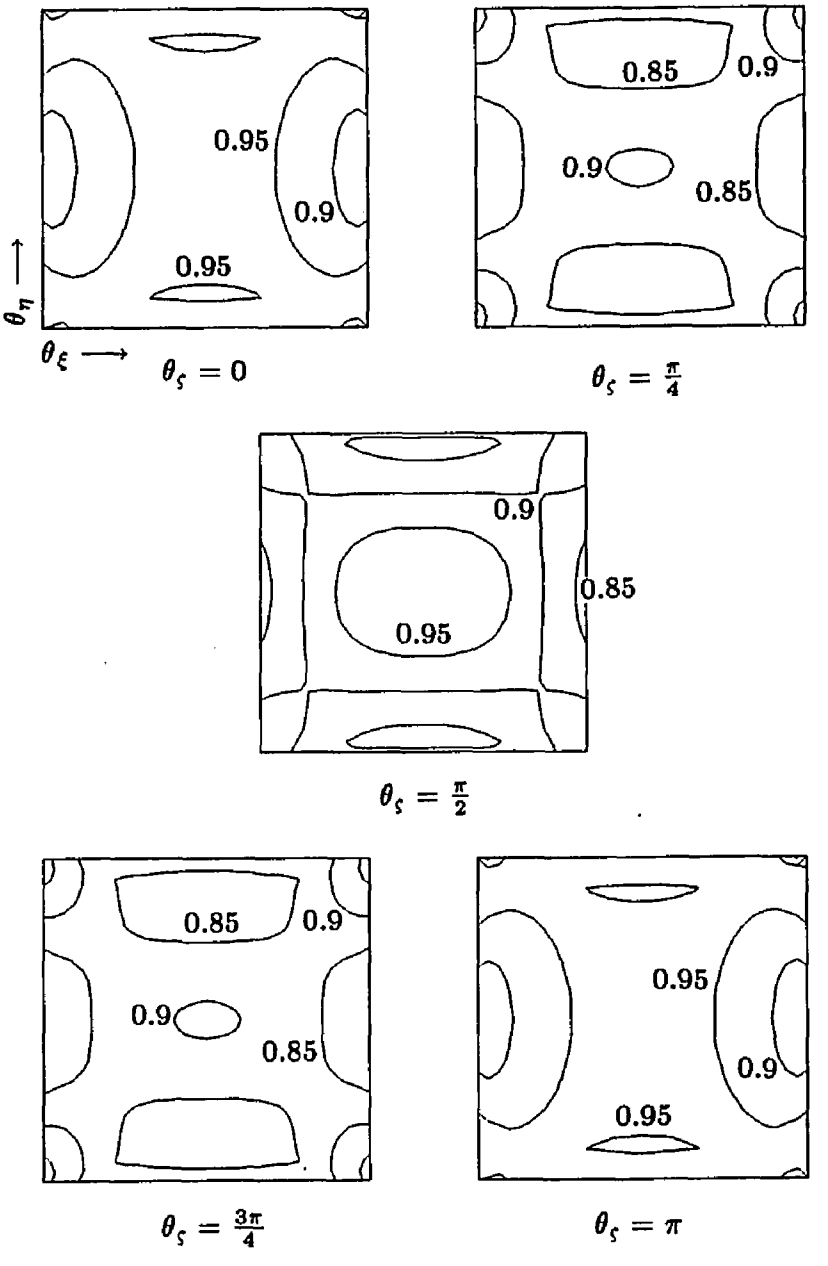

Figure 47: Amplification Factor of Three-Dimensional ADI Scheme for Vector System. $CFL = 4$, $\epsilon_e = \epsilon_i = 0$, $\omega = \Omega = 0$.

center of the Fourier domain ($\omega_\xi \Delta\xi = \omega_\eta \Delta\eta = \omega_\varsigma \Delta\varsigma = \pi/2$) almost crosses the stability criterion. Slight increase of the CFL number can result in instability. The CFL limit in this case is approximately twice the CFL limit based on the scalar analysis. This is a result of the unequal eigenvalues of the flux vectors in the three directions. In other words, unequal eigenvalues can relieve the CFL constraint. In fact, in the limit when the eigenvalues in one direction shrink to zero, the stability analysis returns to two dimensions. The most restrictive CFL limit appears when the three directions have the eigenvalues of the same magnitudes. Choi [53] has shown that in such case the CFL limit (about 2.56) is approximately the same as the scalar CFL limit.

The amplification factor of the highest-wavenumber wave in Fig. 47 is one. This indicates that odd-even decoupling may occur due to lack of explicit viscosity. According to the results of the scalar stability analysis, adding the explicit viscosity alone could cause instability of the mid-wavenumber waves. The implicit viscosity is required to reduce the amplification factor around the mid-wavenumber. A combination of $\epsilon_e = 0.2$ and $\epsilon_i = 1$ appears to work well for the scalar equation. The same combination is also checked for the vector system, and the amplification factor is shown in Fig. 48. As can be seen, the ADI scheme is stable on the vector system, and in general, the implicit and explicit viscosities affect the amplification factor of the vector system in a similar fashion as they do to the scalar equation. This example and many others indicate the qualitative similarity between the stability characteristics of the scalar equation and the vector system.

Next we consider the stability characteristics of the ADI scheme in solving problems involving a rotating body (e. g., a propeller). For comparison, the stability analysis is performed under the same assumptions as the previous case, that is, Eq. (5.17) and (5.18), and the CFL number and the viscosities are also set to be

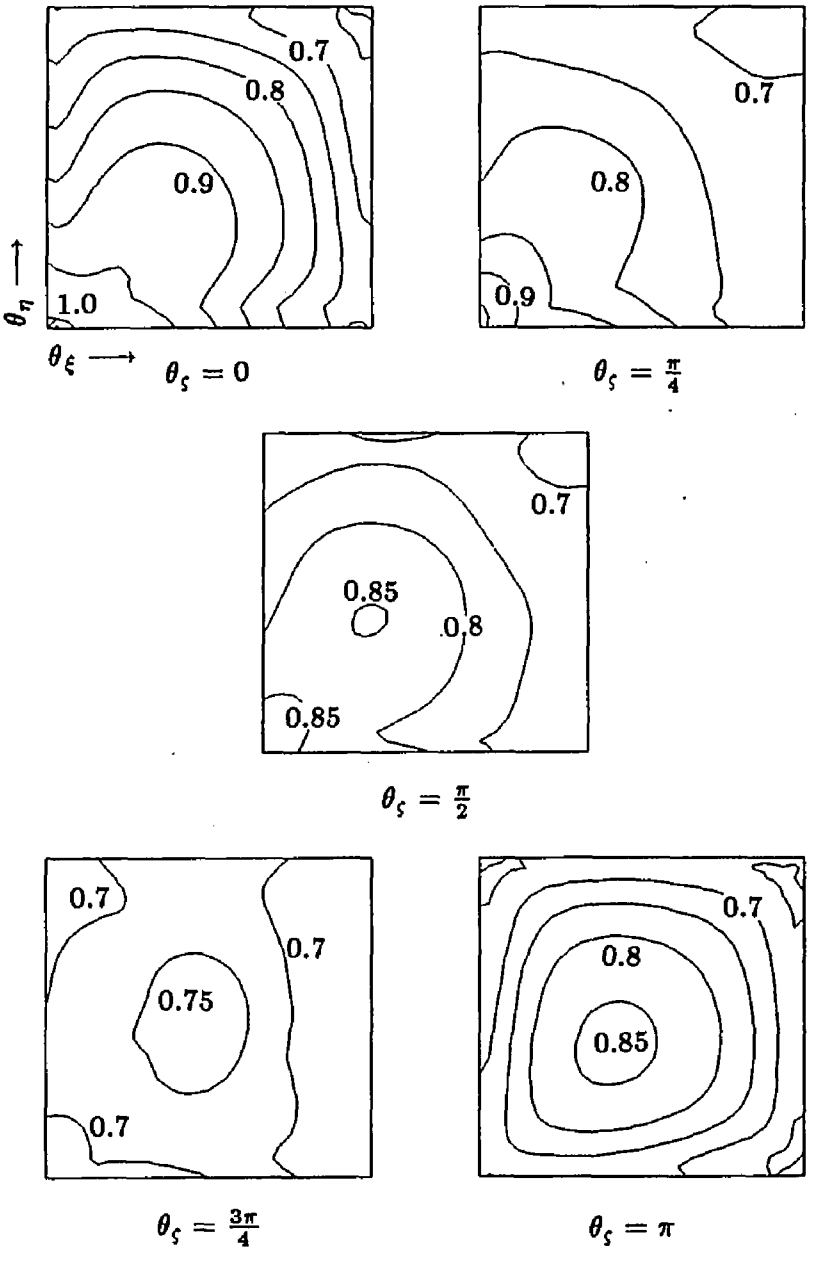

Figure 48: Amplification Factor of Three-Dimensional ADI Scheme for Vector System. $CFL = 4$, $\epsilon_e = 0.2$, $\epsilon_i = 1$, $\omega = \Omega = 0$.

the same ($CFL = 4, \epsilon_e = 0.2, \epsilon_i = 1$). As mentioned earlier, this type of problems can be solved by either relative or absolute coordinates. The circumferential velocity in Eq. (5.18) is for the absolute coordinates. On the relative coordinates the circumferential velocity is $0.3 + \omega r$. The amplification factor for a rotation speed of 60 rpm is shown in Fig. 49 and Fig. 50 for the relative and absolute coordinates, respectively. They both indicate that the scheme is stable regardless of the coordinate system used. In general, however, the amplification factor in either Fig. 49 or Fig. 50 is slightly larger than that in Fig. 48 for non-rotating body. This may indicate a smaller dissipation rate in solving flows around a rotating body. By comparing Fig. 49 and Fig. 50, one actually find very little difference between the amplification factors for different coordinate systems. Earlier derivation shows the difference between using the relative and the absolute coordinates is a transfer of half of the Coriolis force in the θ-momentum equation into the flux vector. Here we see this transfer of source term has nearly no effect on stability. Therefore, as far as the stability is concerned, one is free to choose either relative or absolute coordinates.

5.2.3 Boundary Conditions

The boundary conditions suitable for the two-dimensional ADI scheme can be extended to the three-dimensional ADI scheme in a straightforward manner. For instance, the boundary condition in ξ-direction can be written as

$$\left(\mathbf{S} + \Delta t \mathbf{L}\mathbf{M}_{\xi}^{-1}\delta_{\xi}\mathbf{A}\cdot\right)\Delta\mathbf{Q} = \mathbf{L}\mathbf{M}_{\xi}^{-1}\left(\mathbf{I} + \Delta t\delta_{\eta}\mathbf{B}\cdot\right)^{-1}\left(\mathbf{I} + \Delta t\delta_{\varsigma}\mathbf{C}\cdot\right)^{-1} RHS \quad (5.19)$$

which is extended from Eq. (4.23) for two dimensions. However, the derivation of the three-dimensional eigenmatrices, $\mathbf{M}_{\xi}$, $\mathbf{M}_{\eta}$, and $\mathbf{M}_{\varsigma}$, is not as straightforward as that of two-dimensional ones. The difficulty lies in finding the first two eigenvectors associated with the repeated eigenvalue. The following formulation is derived by

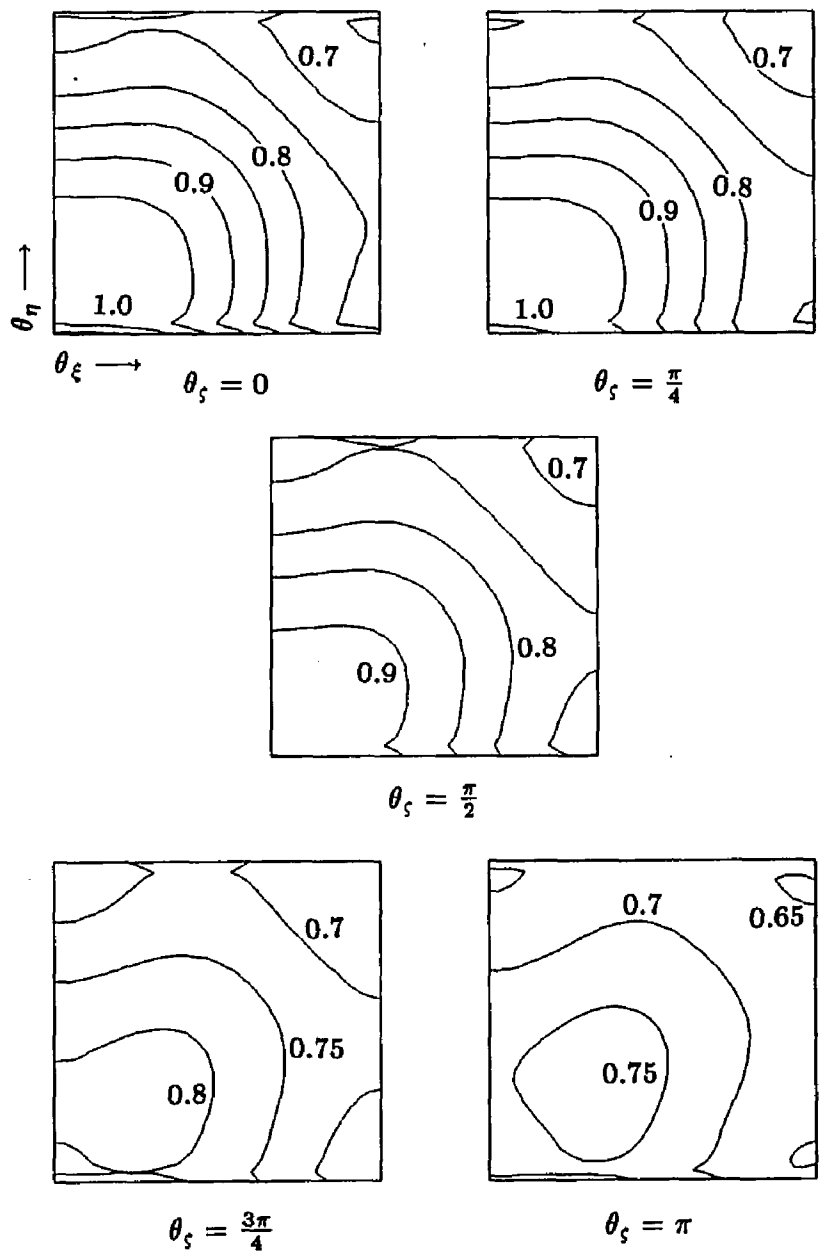

Figure 49: Amplification Factor of Three-Dimensional ADI Scheme for Vector System. $CFL = 4$, $\epsilon_e = 0.2$, $\epsilon_i = 1$, $\omega = \Omega = 60$rpm (Relative Coordinates).

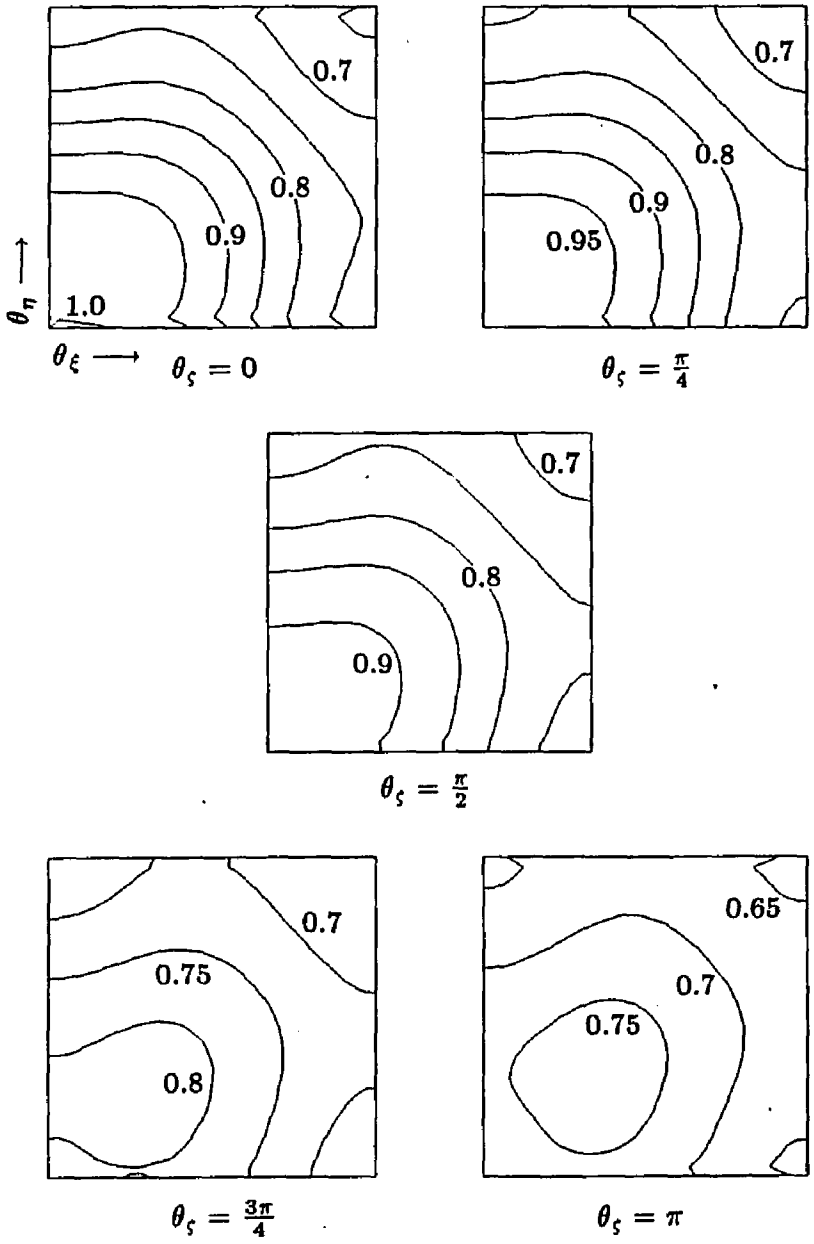

Figure 50: Amplification Factor of Three-Dimensional ADI Scheme for Vector System. $CFL = 4$, $\epsilon_e = 0.2$, $\epsilon_i = 1$, $\omega = 0$, $\Omega = 60$rpm (Absolute Coordinates).

Rogers, Chang, and Kwak [47].

For $\mathbf{M}_\xi$, The form of the first two eigenvectors can be written as

$$(\mathbf{A} - U\mathbf{I})\mathbf{X} = \begin{pmatrix} -U & \beta\xi_x & \beta\xi_y & \beta\xi_z \\ \xi_x & u\xi_x & u\xi_y & u\xi_z \\ \xi_y & v\xi_x & v\xi_y & v\xi_z \\ \xi_z & w\xi_x & w\xi_y & w\xi_z \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = 0$$

This can be reduced to

$$\begin{pmatrix} 0 & \xi_x & \xi_y & \xi_z \\ \xi_x & 0 & 0 & 0 \\ \xi_y & 0 & 0 & 0 \\ \xi_z & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = 0$$

Thus the first two eigenvectors are given by

$$x_1 = 0 \tag{5.20a}$$

$$\xi_x x_2 + \xi_y x_3 + \xi_z x_4 = 0 \tag{5.20b}$$

Since there are three unknown and only one equation for the last three elements of these eigenvectors, many possible combinations can be made up. However, it is important that any two eigenvectors in $\mathbf{M}_\xi$ are not linearly dependent. Should any two be linearly dependent, $\mathbf{M}_\xi$ is singular, and $\mathbf{M}_\xi^{-1}$ is nonexistent. Hence the two eigenvectors determined by Eq. (5.20b) must be chosen carefully to avoid any possible linear dependency.

To construct suitable eigenvectors, consider the left-hand-side of Eq. (5.20b) as a dot product.

$$(\nabla\xi) \cdot \vec{X} = 0 \tag{5.21}$$

where

$$\nabla\xi = \xi_x\vec{i} + \xi_y\vec{j} + \xi_z\vec{k}$$

$$\vec{X} = x_2\vec{i} + x_3\vec{j} + x_4\vec{k}$$

and $\vec{i}$, $\vec{j}$, and $\vec{k}$ are the unit vectors in x-, y-, and z-directions. In other words, one must find two linearly independent vectors that are perpendicular to the vector,

$\nabla\xi$, in three-dimensional space. Figure 51 shows the three-dimensional schematic of the problem. The vector $\nabla\xi$ is normal to a constant ξ surface. It is sufficient to find two non-parallel vectors which are tangential to the constant ξ surface. The obvious choices are

$$\vec{X}_1 = x_\eta \vec{i} + y_\eta \vec{j} + z_\eta \vec{k}$$

and

$$\vec{X}_2 = x_\varsigma \vec{i} + y_\varsigma \vec{j} + z_\varsigma \vec{k}$$

These two vectors satisfy Eq. (5.21),

$$(\nabla\xi) \cdot \vec{X}_1 = \xi_x x_\eta + \xi_y y_\eta + \xi_z z_\eta = \xi_\eta = 0$$

$$(\nabla\xi) \cdot \vec{X}_2 = \xi_x x_\varsigma + \xi_y y_\varsigma + \xi_z z_\varsigma = \xi_\varsigma = 0$$

and $\vec{X}_1$ and $\vec{X}_2$ are linearly independent as long as the Jacobian of transformation

$$J = \frac{\partial(\xi, \eta, \varsigma)}{\partial(x, y, z)}$$

remains finite.

Finally, the first two eigenvectors are given as

$$\mathbf{X}_1 = \begin{pmatrix} 0 \\ x_\eta \\ y_\eta \\ z_\eta \end{pmatrix}, \qquad \mathbf{X}_2 = \begin{pmatrix} 0 \\ x_\varsigma \\ y_\varsigma \\ z_\varsigma \end{pmatrix}$$

The other eigenvectors are relatively easier to find. The eigenmatrix, $\mathbf{M}_\xi$, is found to be

$$\mathbf{M}_\xi = \frac{1}{2C_\xi m_\xi} \begin{pmatrix} \mathrm{M}_{\xi 11} & \mathrm{M}_{\xi 12} & \mathrm{M}_{\xi 13} & \mathrm{M}_{\xi 14} \\ \mathrm{M}_{\xi 21} & \mathrm{M}_{\xi 22} & \mathrm{M}_{\xi 23} & \mathrm{M}_{\xi 24} \\ \mathrm{M}_{\xi 31} & \mathrm{M}_{\xi 32} & \mathrm{M}_{\xi 33} & \mathrm{M}_{\xi 34} \\ \mathrm{M}_{\xi 41} & \mathrm{M}_{\xi 42} & \mathrm{M}_{\xi 43} & \mathrm{M}_{\xi 44} \end{pmatrix} \tag{5.22}$$

where

$$\mathrm{M}_{\xi 11} = 0$$

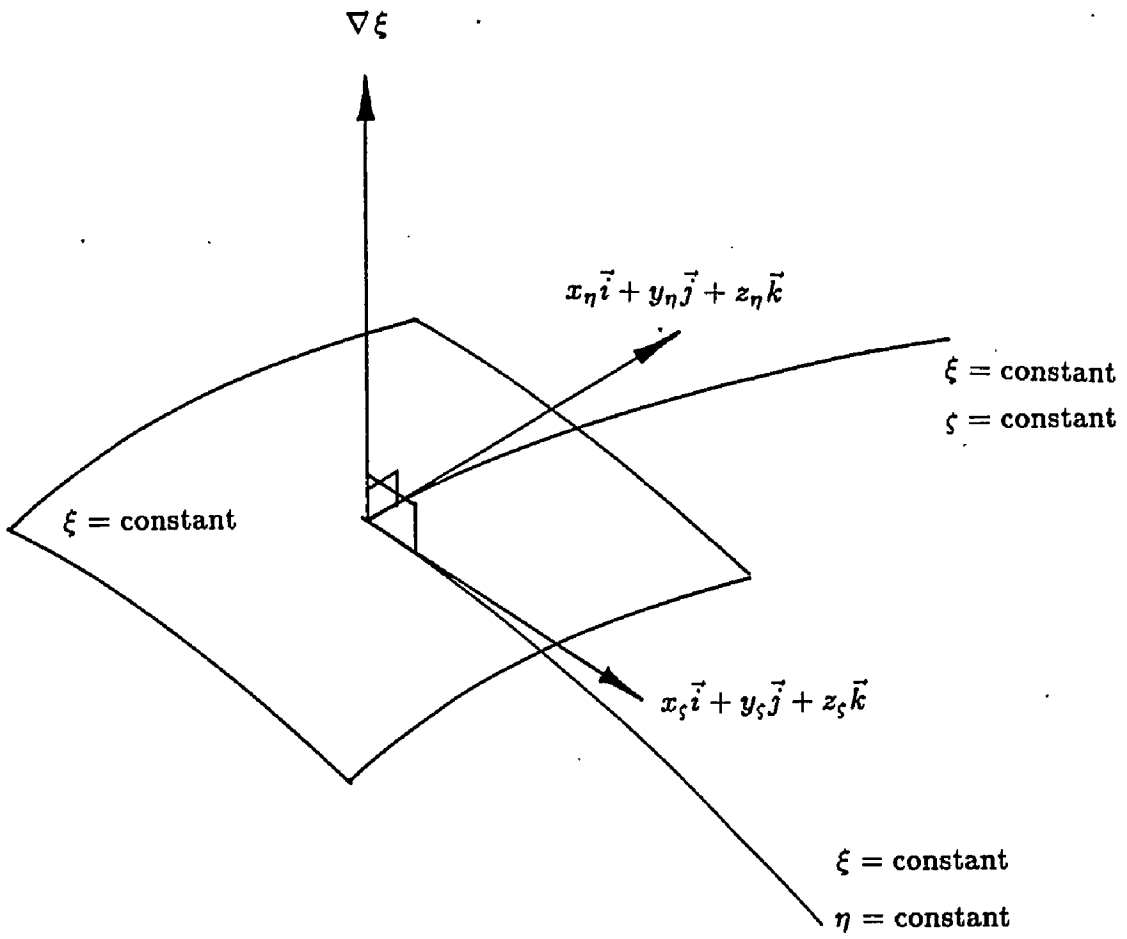

Figure 51: Three-Dimensional Relation between $\nabla\xi$, $\vec{x}_1$, and $\vec{x}_2$.

$$\mathrm{M}_{\xi 12} = 0$$

$$\mathrm{M}_{\xi 13} = -(U - C_\xi)C_\xi$$

$$\mathrm{M}_{\xi 14} = -(U + C_\xi)C_\xi$$

$$\mathrm{M}_{\xi 21} = 2m_\xi x_\eta$$

$$\mathrm{M}_{\xi 22} = 2m_\xi x_\varsigma$$

$$\mathrm{M}_{\xi 23} = um_\xi - \xi_x(U - C_\xi)$$

$$\mathrm{M}_{\xi 24} = -um_\xi + \xi_x(U + C_\xi)$$

$$\mathrm{M}_{\xi 31} = 2m_\xi y_\eta$$

$$\mathrm{M}_{\xi 32} = 2m_\xi y_\varsigma$$

$$\mathrm{M}_{\xi 33} = vm_\xi - \xi_y(U - C_\xi)$$

$$\mathrm{M}_{\xi 34} = -vm_\xi + \xi_y(U + C_\xi)$$

$$\mathrm{M}_{\xi 41} = 2m_\xi z_\eta$$

$$\mathrm{M}_{\xi 42} = 2m_\xi z_\varsigma$$

$$\mathrm{M}_{\xi 43} = wm_\xi - \xi_z(U - C_\xi)$$

$$\mathrm{M}_{\xi 44} = -wm_\xi + \xi_z(U + C_\xi)$$

and

$$m_\xi = \xi_x^2 + \xi_y^2 + \xi_z^2$$

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

$$C_\xi = \sqrt{U^2 + \beta m_\xi}$$

Its inverse matrix, $\mathbf{M}_\xi^{-1}$, is given as

$$\mathbf{M}_\xi^{-1} = \begin{pmatrix} \mathrm{M}_{\xi 11}^{-1} & \mathrm{M}_{\xi 12}^{-1} & \mathrm{M}_{\xi 13}^{-1} & \mathrm{M}_{\xi 14}^{-1} \\ \mathrm{M}_{\xi 21}^{-1} & \mathrm{M}_{\xi 22}^{-1} & \mathrm{M}_{\xi 23}^{-1} & \mathrm{M}_{\xi 24}^{-1} \\ \mathrm{M}_{\xi 31}^{-1} & \mathrm{M}_{\xi 32}^{-1} & \mathrm{M}_{\xi 33}^{-1} & \mathrm{M}_{\xi 34}^{-1} \\ \mathrm{M}_{\xi 41}^{-1} & \mathrm{M}_{\xi 42}^{-1} & \mathrm{M}_{\xi 43}^{-1} & \mathrm{M}_{\xi 44}^{-1} \end{pmatrix} \tag{5.23}$$

where

$$\mathrm{M}_{\xi 11}^{-1} = x_\varsigma(w\xi_y - v\xi_z) + y_\varsigma(u\xi_z - w\xi_x) + z_\varsigma(v\xi_x - u\xi_y)$$

$$\mathrm{M}_{\xi 12}^{-1} = z_\varsigma(Uv + \beta\xi_y) - y_\varsigma(Uw + \beta\xi_z)$$

$$\mathrm{M}_{\xi 13}^{-1} = x_\varsigma(Uw + \beta\xi_z) - z_\varsigma(Uu + \beta\xi_x)$$

$$M^{-1}_{\xi 14} = y_\varsigma(Uu + \beta\xi_x) - x_\varsigma(Uv + \beta\xi_y)$$

$$M^{-1}_{\xi 21} = x_\eta(v\xi_z - w\xi_y) + y_\eta(w\xi_x - u\xi_z) + z_\eta(u\xi_y - v\xi_x)$$

$$M^{-1}_{\xi 22} = y_\eta(Uw + \beta\xi_z) - z_\eta(Uv + \beta\xi_y)$$

$$M^{-1}_{\xi 23} = z_\eta(Uu + \beta\xi_x) - x_\eta(Uw + \beta\xi_z)$$

$$M^{-1}_{\xi 24} = x_\eta(Uv + \beta\xi_y) - y_\eta(Uu + \beta\xi_x)$$

$$M^{-1}_{\xi 31} = -\frac{1}{J}m_\xi$$

$$M^{-1}_{\xi 32} = -\frac{1}{J}(U + C_\xi)\xi_x$$

$$M^{-1}_{\xi 33} = -\frac{1}{J}(U + C_\xi)\xi_y$$

$$M^{-1}_{\xi 34} = -\frac{1}{J}(U + C_\xi)\xi_z$$

$$M^{-1}_{\xi 41} = \frac{1}{J}m_\xi$$

$$M^{-1}_{\xi 42} = \frac{1}{J}(U - C_\xi)\xi_x$$

$$M^{-1}_{\xi 43} = \frac{1}{J}(U - C_\xi)\xi_y$$

$$M^{-1}_{\xi 44} = \frac{1}{J}(U - C_\xi)\xi_z$$

The eigenmatrices and their inverse matrices for η- and ς-directions can be obtained by replacing ξ in $\mathbf{M}_\xi$ and $\mathbf{M}_\xi^{-1}$ by either η or ς. Note that the above derivation is based on the (x, y, z) coordinate system. However, all the formulas can be applied to cylindrical coordinate system by the following modifications,

$$x \longrightarrow r$$

$$y \longrightarrow r\theta$$

$$u \longrightarrow u_r$$

$$v \longrightarrow u_\theta$$

$$w \longrightarrow u_x$$

5.3 Explicit Scheme

Extension of Runge-Kutta scheme to three-dimensional problems is straightforward. The time-marching procedure suitable for one dimension such as Eq. (3.25)

is readily applicable to three dimensions. The only difference is the right-hand-side operator, P. For the propeller flow problem, the right-hand-side contains space-difference operators in three directions and a source vector.

The boundary conditions in three dimensions can still be treated based on MOC. Equation (3.27) is appropriate for one-, two-, or three-dimensional problems. The inverse eigenmatrices in three directions, $\mathbf{M}_\xi^{-1}$, $\mathbf{M}_\eta^{-1}$, and $\mathbf{M}_\zeta^{-1}$ were derived earlier for the implicit scheme (see Sec. 5.2.3). For a rotating propeller in an incompressible flow, three boundary conditions need to be specified on an inflow boundary since there are three incoming characteristics. One possible set of boundary conditions is the velocity components in the three directions. At the outflow boundary, pressure can be specified.

Stability analysis of the implicit scheme uncovers the restrictive stability limit it suffers. The explicit scheme, on the other hand, remains approximately the same stability characteristics when extended to three dimensions. As a matter of fact, if one defines

$$CFL = \frac{\lambda_\xi \Delta t}{\Delta \xi} + \frac{\lambda_\eta \Delta t}{\Delta \eta} + \frac{\lambda_\zeta \Delta t}{\Delta \zeta}$$

and performs the symmetric stability analysis on Eq. (5.7), then the CFL limit would be the same as that of one dimension. Or we may say the maximum time-step size is reduced to one third from one dimension to three dimensions.

The residual-smoothing method for three dimensions follows that for two dimensions. The finite-difference equation for three-dimensional residual-smoothing can be written as

$$(1 - \epsilon_\xi \delta_{\xi\xi})(1 - \epsilon_\eta \delta_{\eta\eta})(1 - \epsilon_\zeta \delta_{\zeta\zeta})\bar{P} = P \tag{5.24}$$

which is extended from Eq. (4.56) for two dimensions. In Eq. (5.24), P is the right-hand-side operator, and $\bar{P}$ is used to update the dependent variables at the

new time level in lieu of P. Once again, the wave mode which excites instability is to be found first, then the CFL limit with residual-smoothing employed can be obtained. Assume that

$$\epsilon_\xi = \epsilon_\eta = \epsilon_\varsigma$$

and define

$$\epsilon \equiv \epsilon_\xi + \epsilon_\eta + \epsilon_\varsigma$$

The unstable wave mode is found to satisfy

$$C = \frac{\sqrt{1+\frac{4}{3}\epsilon+\frac{100}{9}\epsilon^2}-(1+\frac{2}{3}\epsilon)}{\frac{8}{3}\epsilon}$$

and

$$S = \frac{\sqrt{2(1+\frac{2}{3}\epsilon)\sqrt{1+\frac{4}{3}\epsilon+\frac{100}{9}\epsilon^2}-2-\frac{8}{3}\epsilon-\frac{40}{9}\epsilon^2}}{\frac{8}{3}\epsilon}$$

At this wave mode the gain of maximum CFL number is

$$\text{gain} = \frac{\epsilon\left[(5+\frac{10}{3}\epsilon)-\sqrt{1+\frac{4}{3}\epsilon+\frac{100}{9}\epsilon^2}\right]^3}{24\sqrt{2(1+\frac{2}{3}\epsilon)\sqrt{1+\frac{4}{3}\epsilon+\frac{100}{9}\epsilon^2}-2-\frac{8}{3}\epsilon-\frac{40}{9}\epsilon^2}} \tag{5.25}$$

The gain curve as a function of ϵ is shown in Fig. 52.

The scalar stability analysis addressed above gives quantitative description of the stability characteristics of a linear hyperbolic equation solved by the Runge-Kutta scheme. Similar stability behavior is expected for a hyperbolic system such as Eq. (5.2). To verify this, vector stability analysis is also performed. In order to compare the stability results with those of the ADI scheme, consistent cell size and velocity components (Eqs. (5.17) and (5.18)) are assumed.

Guided by the scalar stability analysis, the vector stability analysis are carried out based on the following parameters,

$$CFL = 7, \qquad \epsilon = 1, \qquad \epsilon_e = 3.5 \tag{5.26}$$

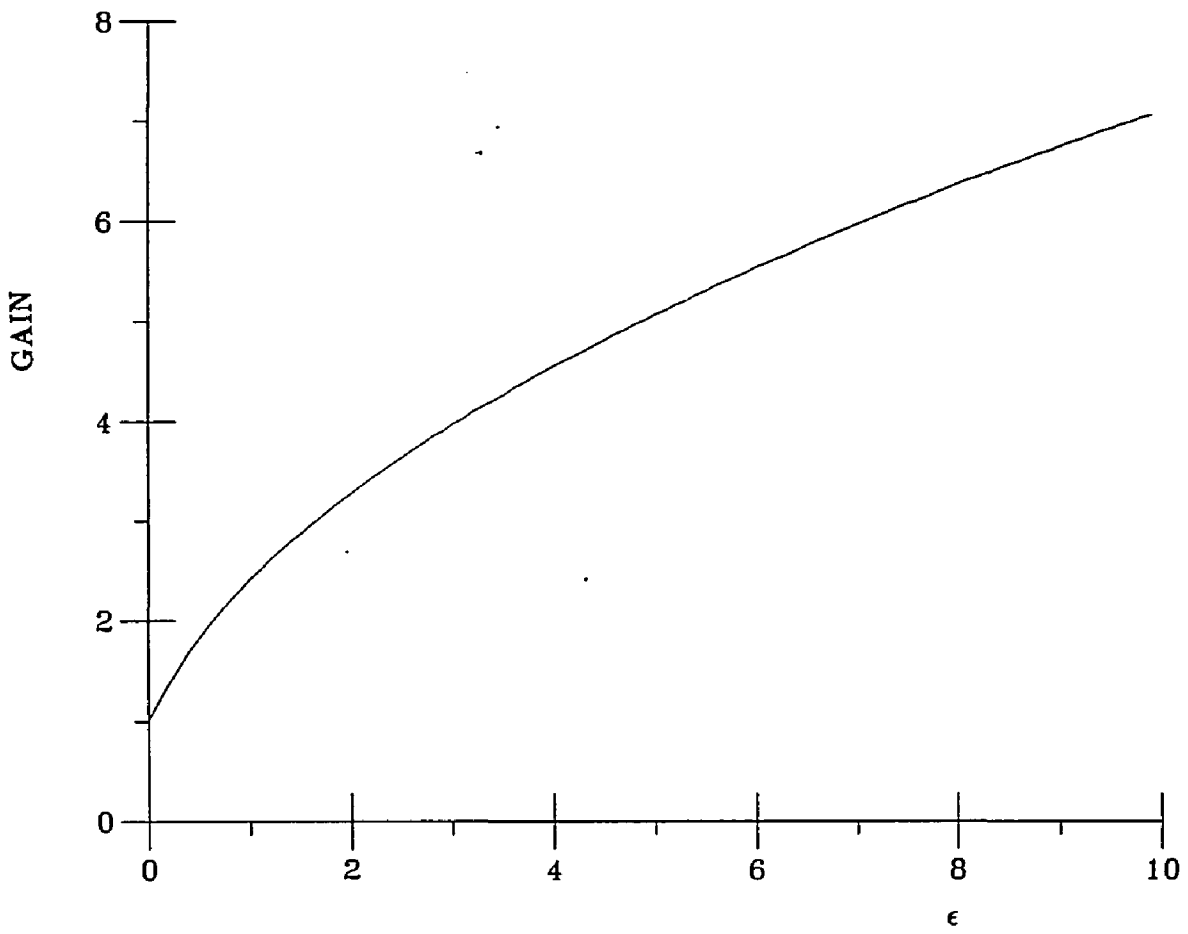

Figure 52: CFL Number Gain Due to Residual-Smoothing for Three-Dimensional Runge-Kutta Scheme.

The amplification factor for a fixed body and fixed coordinates ($\omega = \Omega = 0$) is shown in Fig. 53. Figure 53 shows that the Runge-Kutta scheme is stable in such case, and the pattern of the amplification factor is similar to that of the scalar stability analysis (see Fig. 9). The amplification factor for a body rotating at 60 rpm is shown in Fig. 54 and Fig. 55 for the relative coordinates ($\omega = \Omega = 60\text{rpm}$) and the absolute coordinates ($\omega = 0, \Omega = 60\text{rpm}$). As we see, both coordinate systems yield almost identical amplification factor as the case for a fixed body. This implies that the Runge-Kutta scheme is less sensitive to the rotation of the body than the ADI scheme, regardless of the coordinate system chosen.

5.4 Results and Discussion

5.4.1 Efficiency Comparisons of ADI and Runge-Kutta schemes

Efficiency is of major concern for a numerical scheme especially in three dimensions. As before, we will compare the efficiency of ADI and Runge-Kutta schemes based on the number of iterations, and the required CPU time to reach a specific convergence level. These comparisons are made on a $7 \times 33 \times 65$ mesh for a hull-fin combination as shown in Fig. 56. Comparisons of the convergence rates of the Runge-Kutta and the ADI scheme in terms of the number of iterations are shown in Fig. 57, and comparisons based on the CYBER-205 CPU time are given in Fig. 58.

On the basis of the number of iterations, the ADI scheme and the Runge-Kutta scheme with residual-smoothing converge at essentially identical rates as shown in Fig. 57. Both schemes converge at a rate of approximately five orders of magnitude per 1000 iterations. Without residual-smoothing, the Runge-Kutta scheme is slowed down by a factor of more than two. Recall that in two dimensions, the ADI scheme converges at a substantially faster rate than the Runge-Kutta

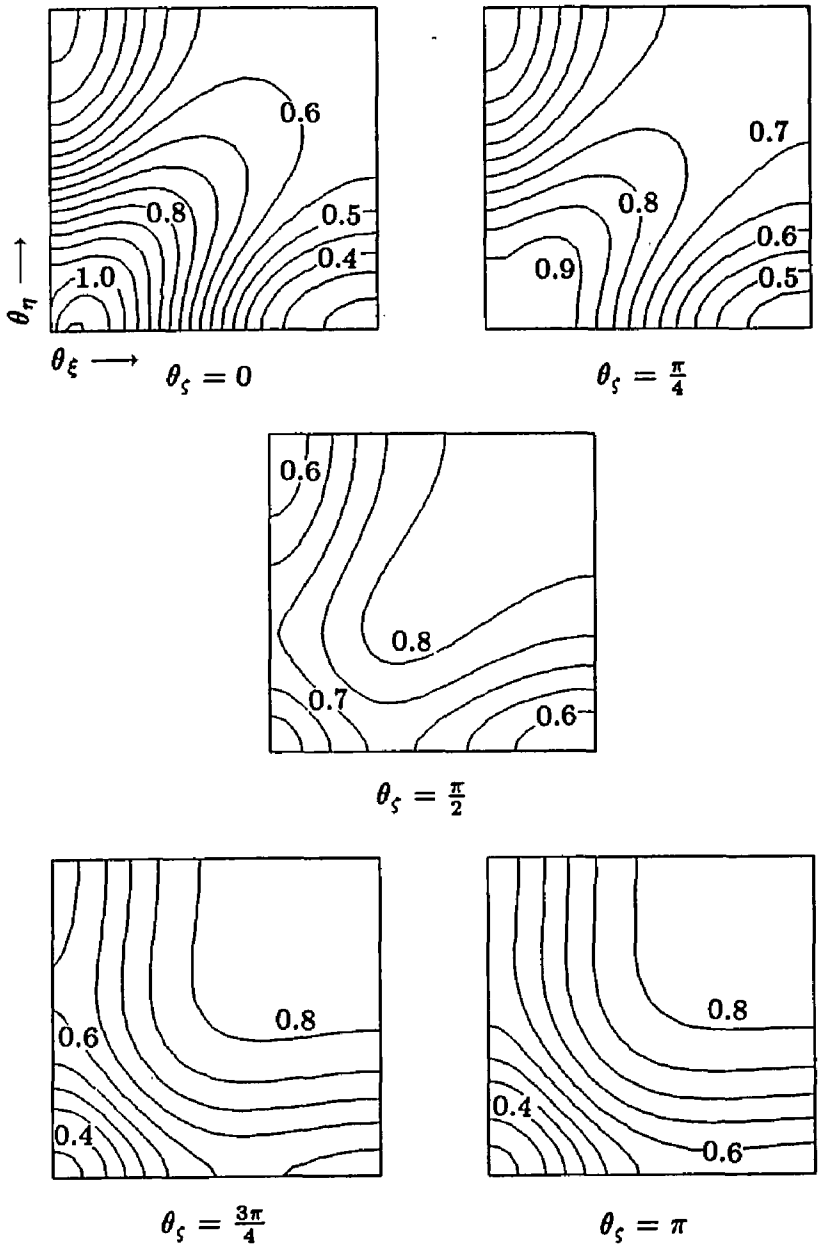

Figure 53: Amplification Factor of Three-Dimensional Runge-Kutta Scheme for Vector System. $CFL = 7$, $\epsilon_e = 3.5$, $\epsilon = 1$, $\omega = \Omega = 0$.

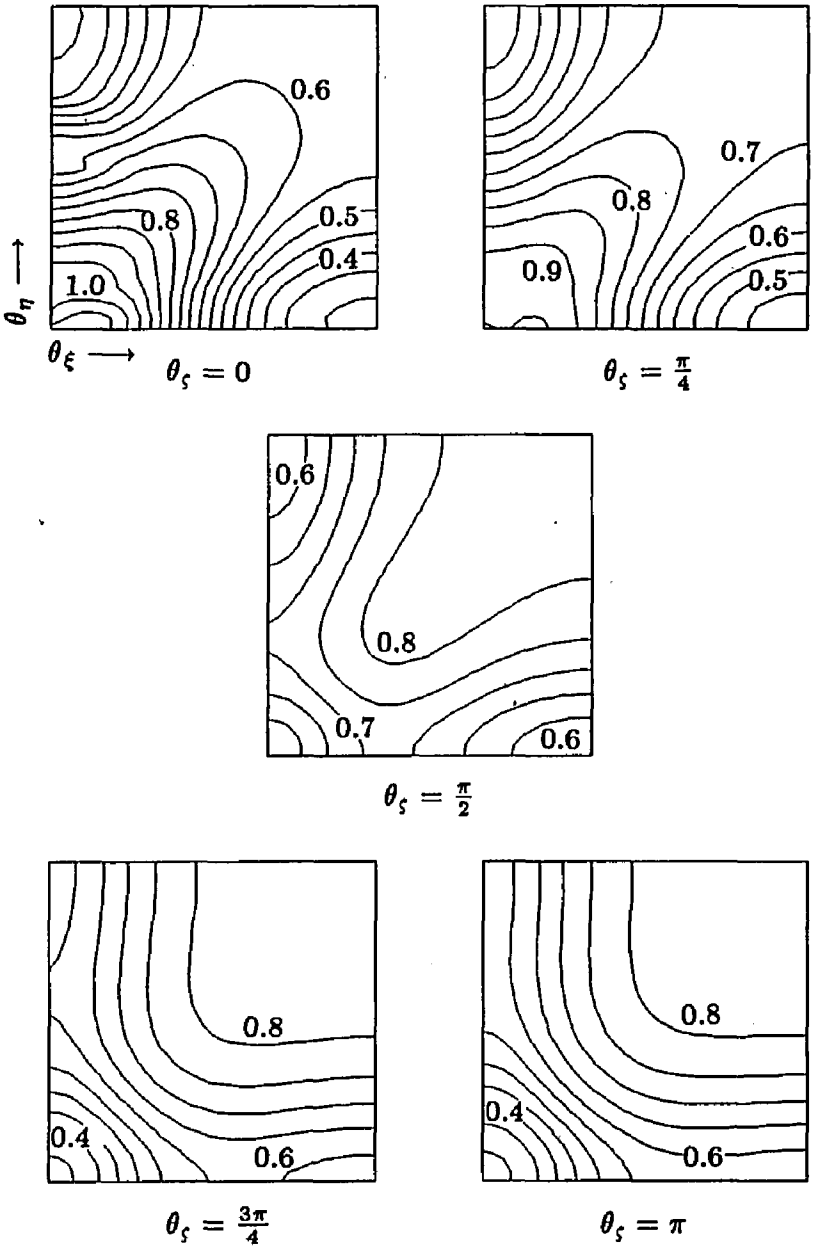

Figure 54: Amplification Factor of Three-Dimensional Runge-Kutta Scheme for Vector System. $CFL = 7$, $\epsilon_e = 3.5$, $\epsilon = 1$, $\omega = \Omega = 60$rpm (Relative Coordinates).

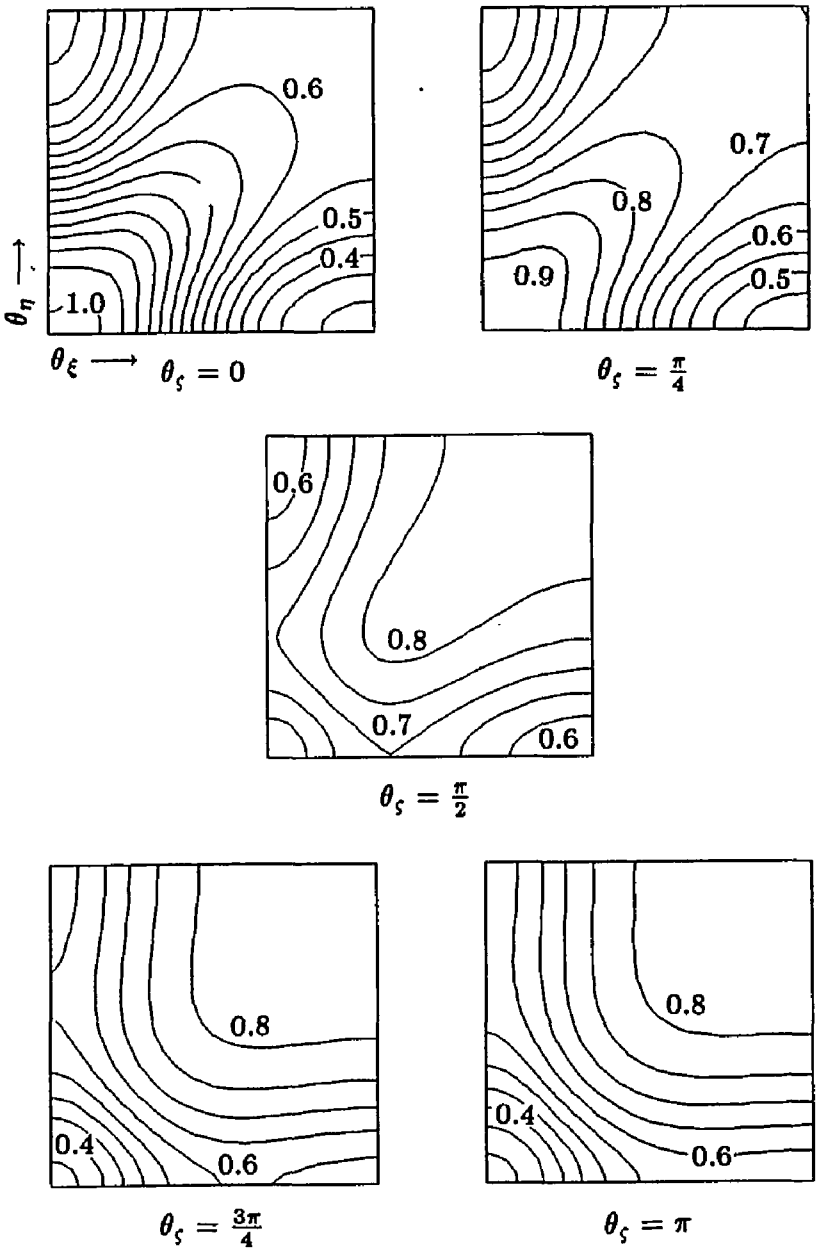

Figure 55: Amplification Factor of Three-Dimensional Runge-Kutta Scheme for Vector System. $CFL = 7$, $\epsilon_e = 3.5$, $\epsilon = 1$, $\omega = 0$, $\Omega = 60$rpm (Absolute Coordinates).

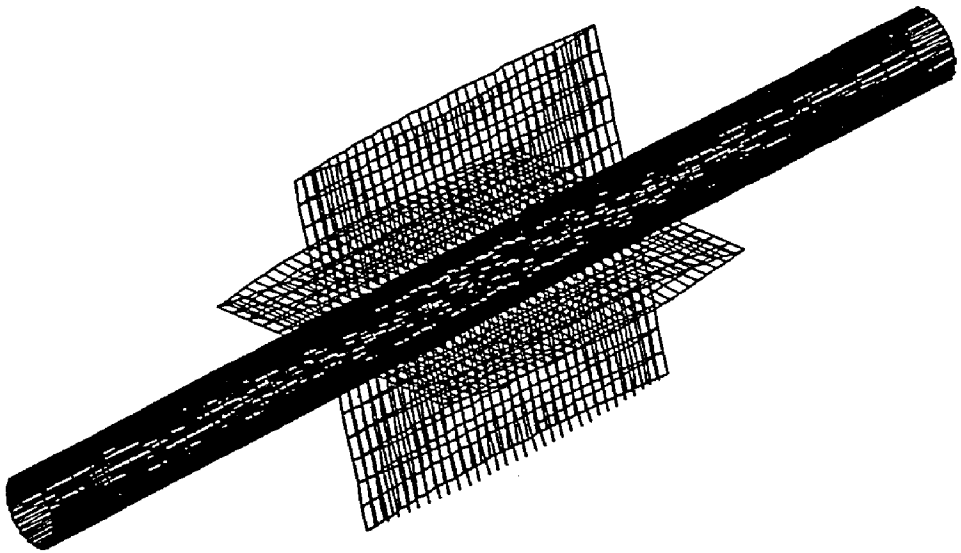

Figure 56: Hull-Fin Combination with Surface Grid. Grid size is $7 \times 33 \times 65$.

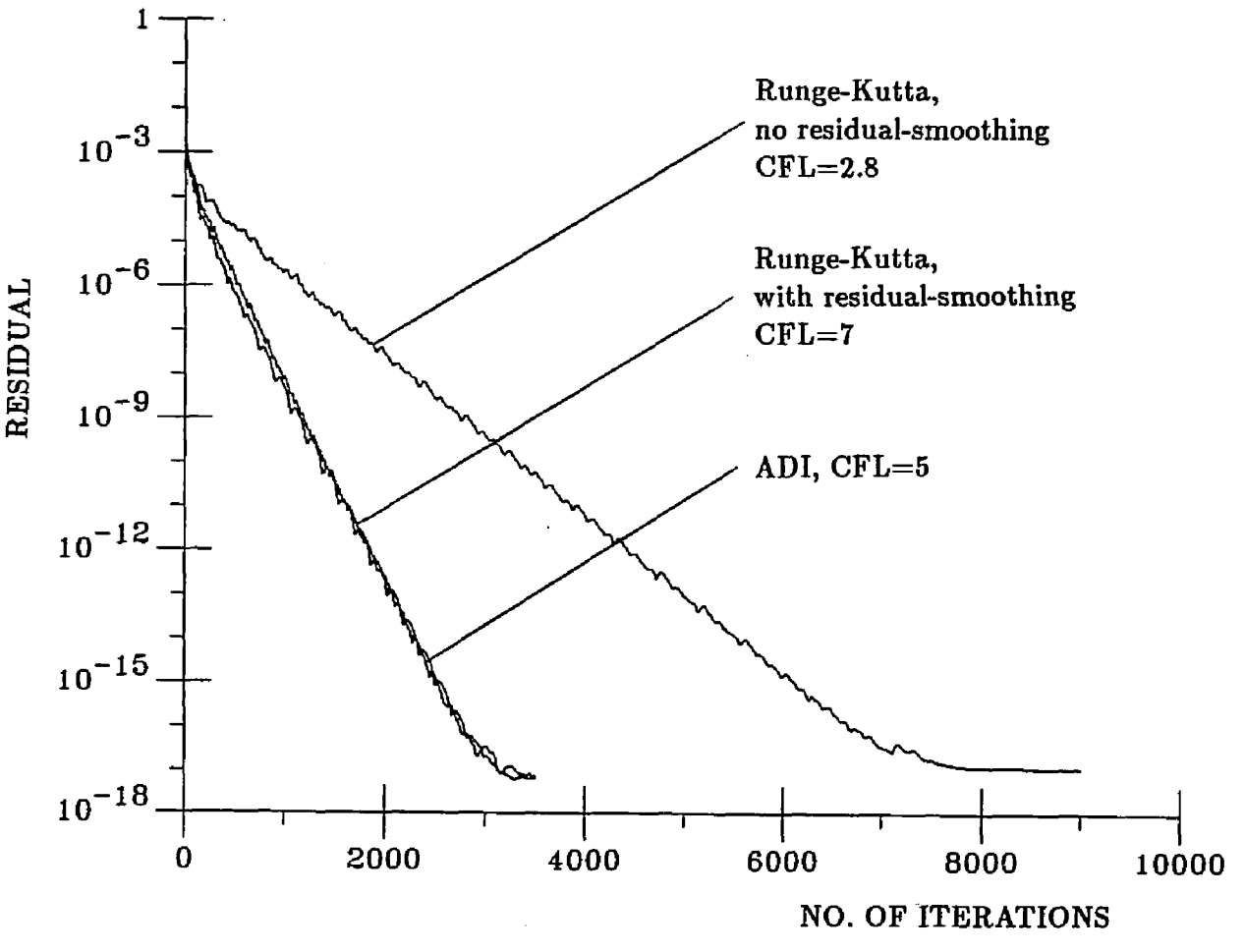

Figure 57: Comparison of Convergence Rates of Runge-Kutta and ADI Scheme for Flow through Hull-Fin Combination.

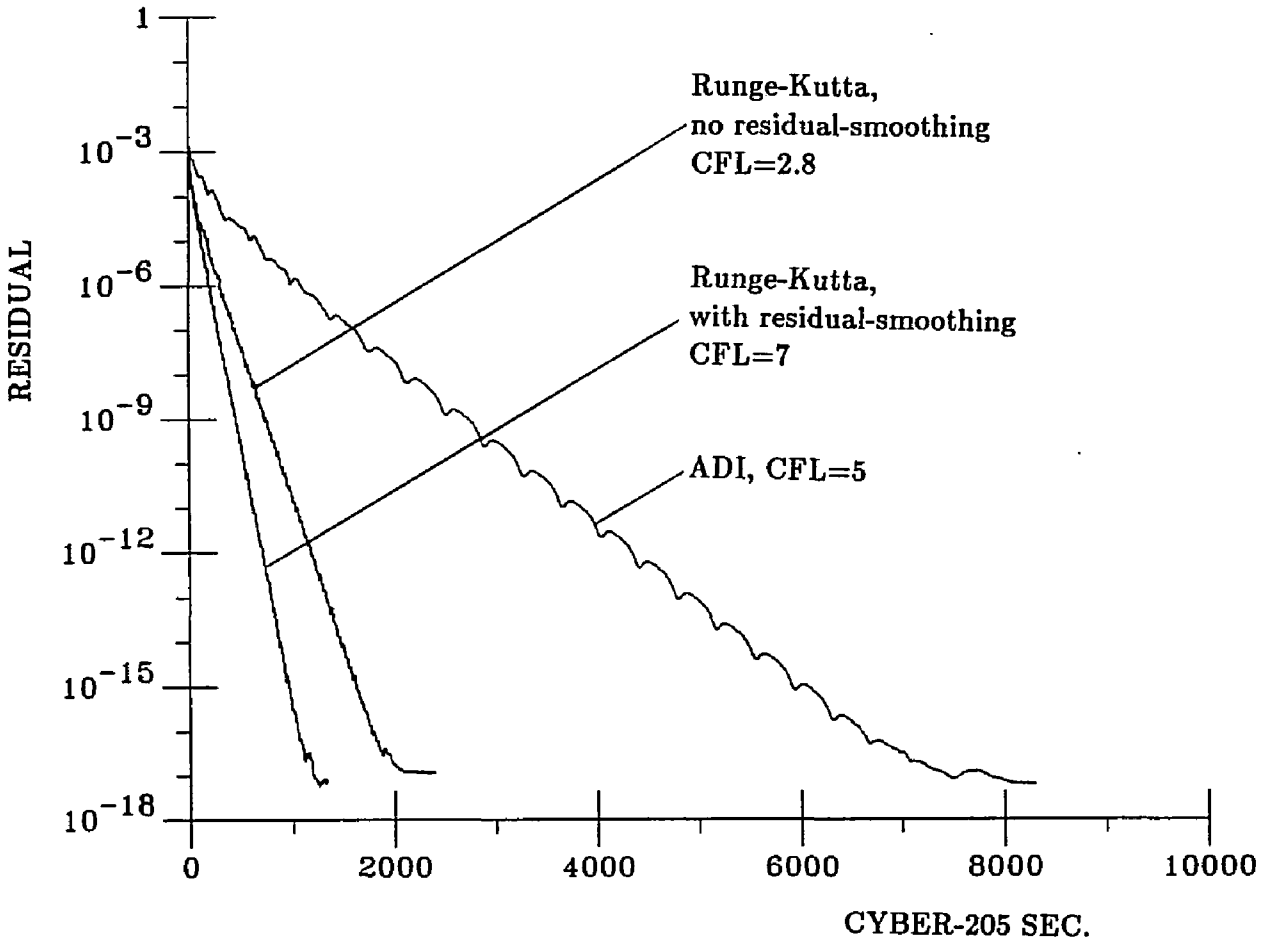

Figure 58: Comparison of CPU Cost of Runge-Kutta and ADI Scheme for Flow through Hull-Fin Combination.

scheme. The primary reason is not because of an improvement of the Runge-Kutta scheme in three dimensions, but because of a significant reduction of the CFL limit of the ADI scheme in three dimensions. As previous stability analysis shows, the ADI scheme essentially becomes conditionally stable in three dimensions, while the Runge-Kutta scheme remains the same limiting CFL number, which is 2.8. By means of residual-smoothing, the CFL limit can be brought up to 7, as shown in Fig. 57. The optimum CFL number of the ADI scheme for this case is experimentally found to be 5. This CFL number is twice as large as the limiting CFL number predicted by the stability analysis based upon three equal velocity components. The reason is that the eigenvalues in the r-direction are significantly smaller than those in other directions because u_r is small. If the magnitudes of the velocity components were closer, the CFL limit of the ADI scheme would be even more restrictive.

A comparison on the computation costs for these two schemes is shown in Fig. 58. Quite obviously, the Runge-Kutta scheme is much more economical than the ADI scheme in solving the three-dimensional Euler equations. The CPU requirement of the Runge-Kutta is about one fifth of the ADI scheme. Also notice that the advantage of using residual-smoothing becomes less because it requires more computational work. The superiority of the Runge-Kutta scheme comes from its fewer computational counts per time step and its higher level of vectorizability on CYBER-205.

5.4.2 Propeller Calculations

In this section, we shall apply the schemes studied so far to solve the inviscid flow around a marine propeller in the wake of a ship-hull. This represents one of the practical applications of numerically solving the Euler equations in engineering. Most of the contemporary marine propeller design procedures utilize the lifting-

surface method [54-59] which is based on the potential theory. This method can solve the propeller flowfields efficiently. However, the potential theory is limited to irrotational flows. Generally, a propeller flowfield is strongly rotational because the propeller is operated in the ship wake. The most direct way to assess the performance of a propeller in such flowfields is by solving the Euler equations.

Previous study reveals the superiority of the Runge-Kutta scheme over the ADI scheme in solving the Euler equations. We therefore use the Runge-Kutta scheme to perform a series of computation on a propeller pictured in Fig. 59. This particular 5-blade propeller has a relatively high pitch and large area. The sections are NACA-66 airfoils with linearly varying thickness, and the hub-to-tip ratio is 0.2. This propeller is assumed to be operated inside a shroud.

Calculations for this marine propeller are carried out on a mesh of $21 \times 21 \times 61$ grid points. (21 radially, 21 circumferencially, and 61 axially) Figure 60 shows the helical, body-fitted grid system generated by an algebraic grid generator. In principle, because the present algorithm is applied to a general coordinate, any arbitrary grid system can be utilized for computation as long as it is body-fitted. In practice, the computation is most efficient and robust when the dependent variables change uniformly across the grid points. The grid generator utilized here is designed to meet such requirement. For this grid system, the fully vectorized Euler solver runs at a speed of 3 iterations per second on CYBER-205. The calculations presented have converged at least four orders of magnitude.

The boundary conditions for these computations are based on the theory of characteristics as discussed above. The characteristic theory enforces specification of a boundary condition for each incoming characteristic line. For the upstream boundary, three characteristic lines are incoming, and three velocity components are specified. The downstream boundary requires only one boundary condition,

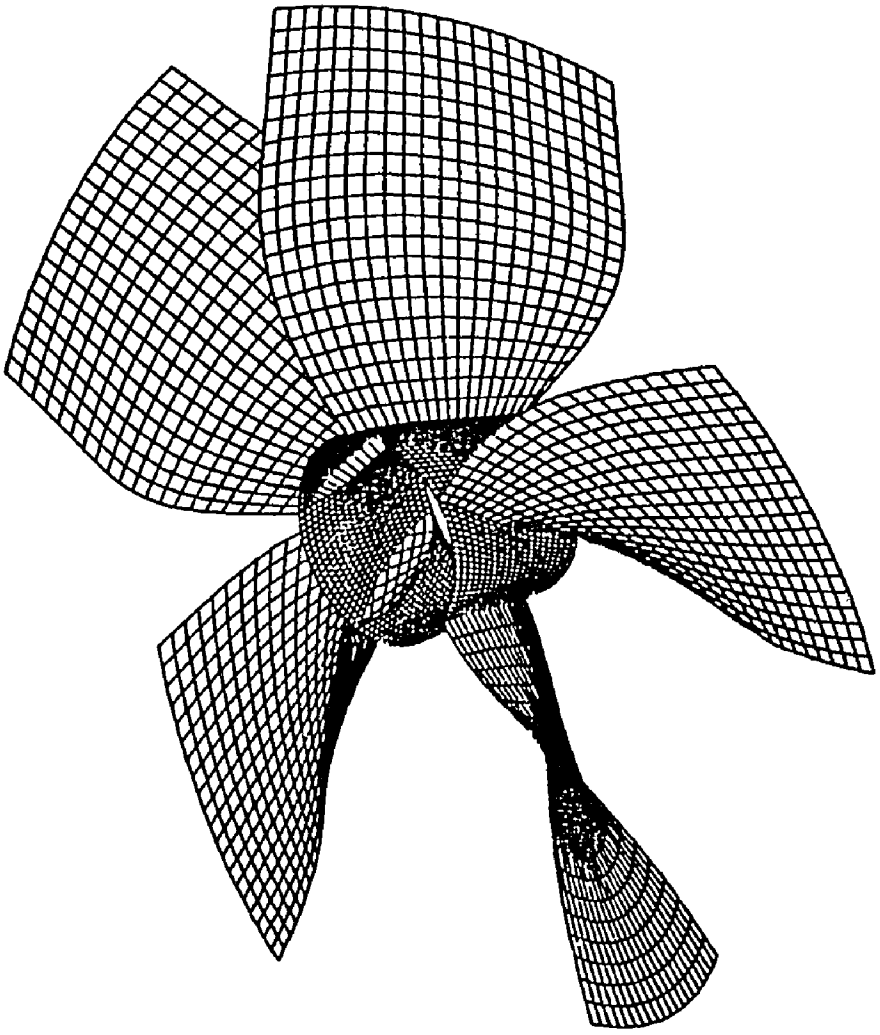

Figure 59: Test Propeller with Surface Grid.

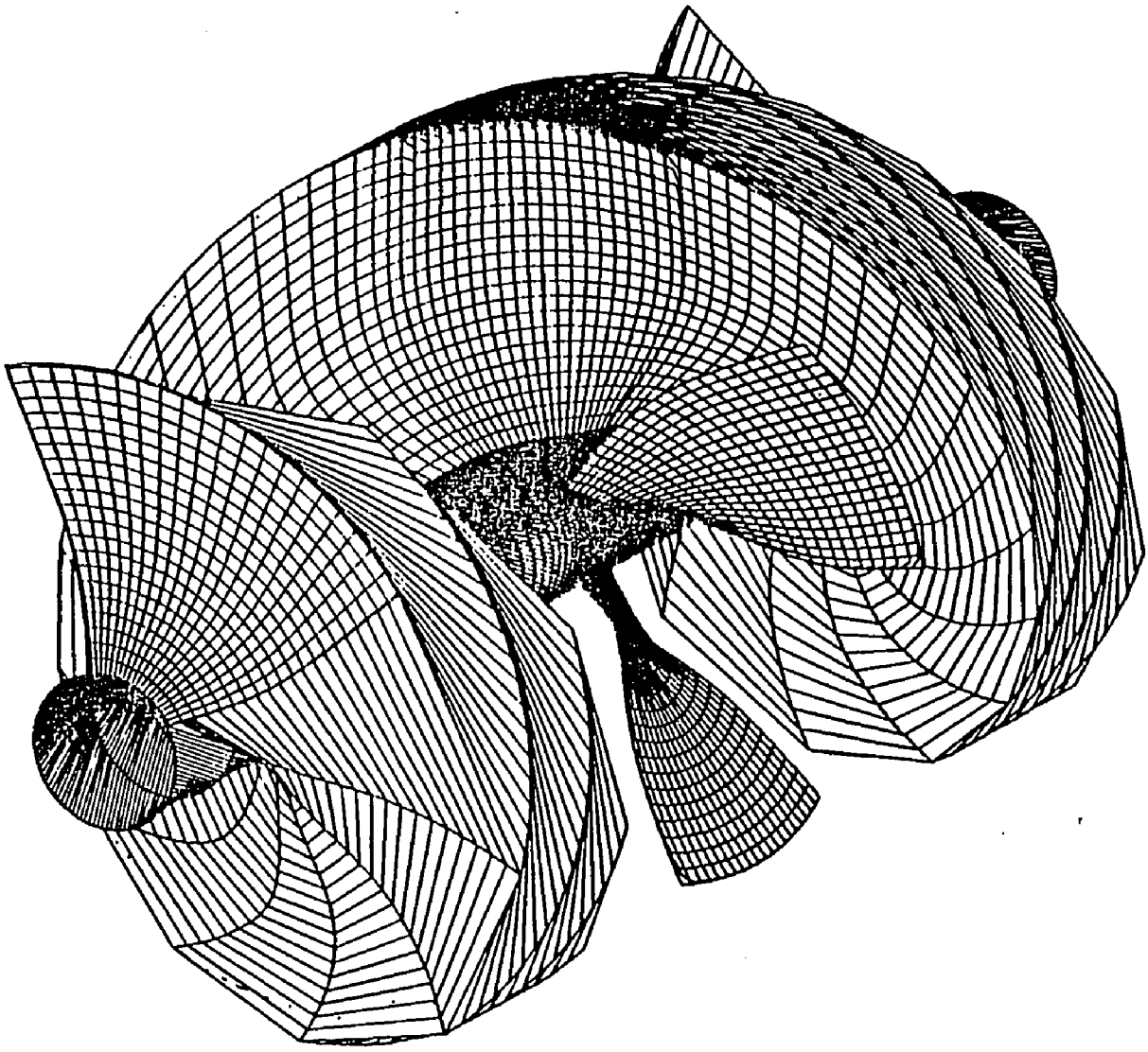

Figure 60: Grid for Computing Flow through Test Propeller. Grid Size is 21 × 21 × 61.

and the pressure is specified. On the wall boundaries such as the blades, the hub, and the shroud, zero normal velocity is enforced. The boundary conditions requires inversions of 4×4 matrices for each boundary point. When using a vector machine, these inversions can become bottle-necked if they are not vector processed. Vectorizing the boundary points is more difficult because their storage addresses may not meet the vectorization requirement for the CYBER-205. In our Euler solver, this has been overcome by utilizing the special gathering and scattering technique to vector process these inversions.

Overall performance data obtained by solving the Euler equations are presented for a number of advance ratios in Fig. 61 through Fig. 63. The advance ratio is defined as

$$J = \frac{U_0}{nD}$$

where n is the number of revolution per second, D is the propeller-tip diameter, and U_0 is the ship speed. The results for both irrotational and rotational flows are shown in the same figures for comparison. The irrotational flows are formed by imposing uniform inflows, while the rotational flows are formed by specifying a sheared inflow at far upstream. The sheared velocity profile is give by

$$u_z = U_0 \left[1 - 0.5 \left(\frac{r_{tip} - r}{r_{tip} - r_{hub}} \right)^2 \right]$$

$$u_r = u_\theta = 0$$

which gives a boundary-layer-like profile with the velocity 1.0 at the tip and 0.5 at the hub.

The thrust coefficient (K_T) and torque coefficient (K_Q) versus advance ratio are shown in Fig. 61 and Fig. 62, respectively. These coefficients are defined by

$$K_T = \frac{T}{\rho n^2 D^4}, \qquad K_Q = \frac{Q}{\rho n^2 D^5}$$

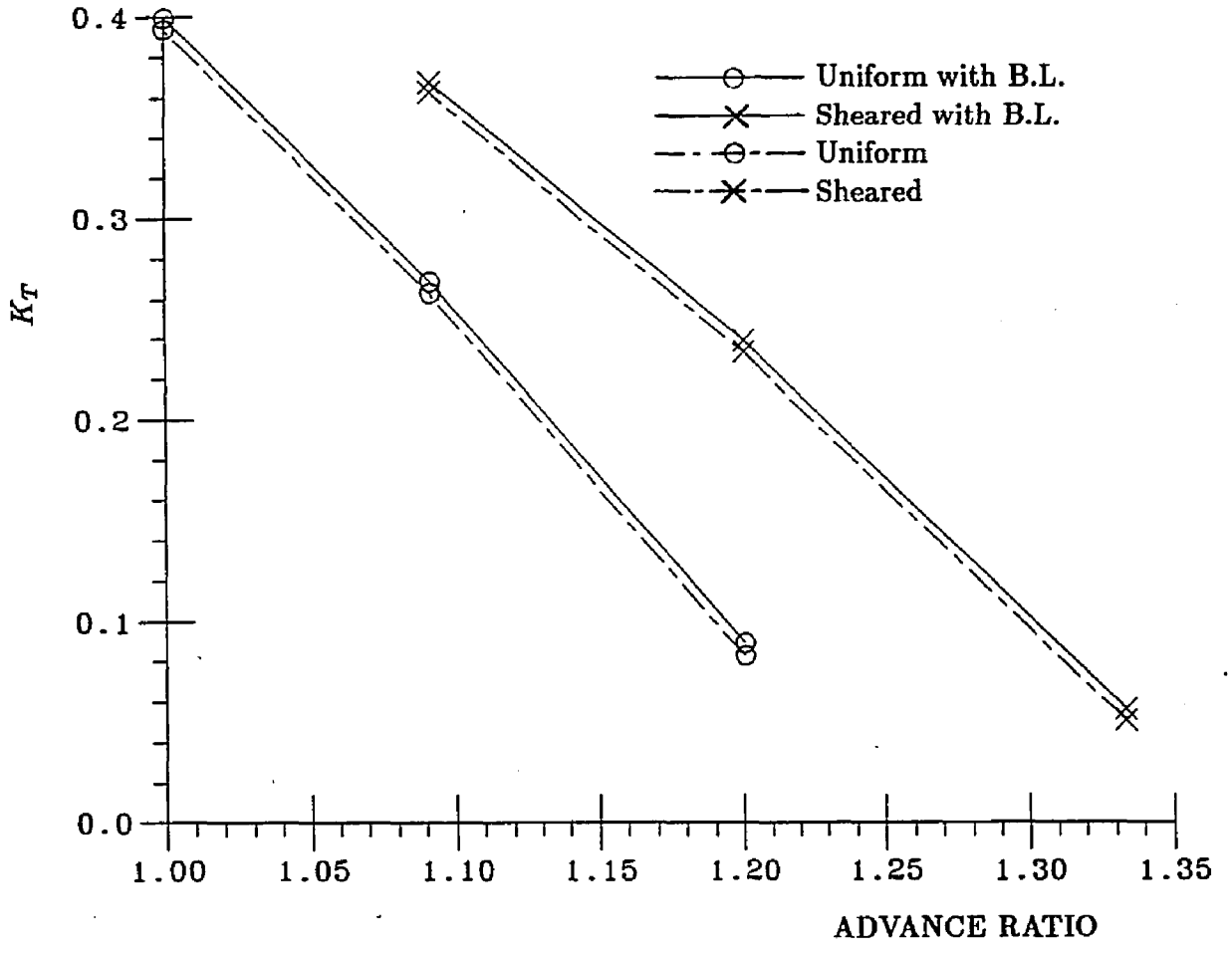

Figure 61: Computed Thrust Coefficient of Test Propeller.

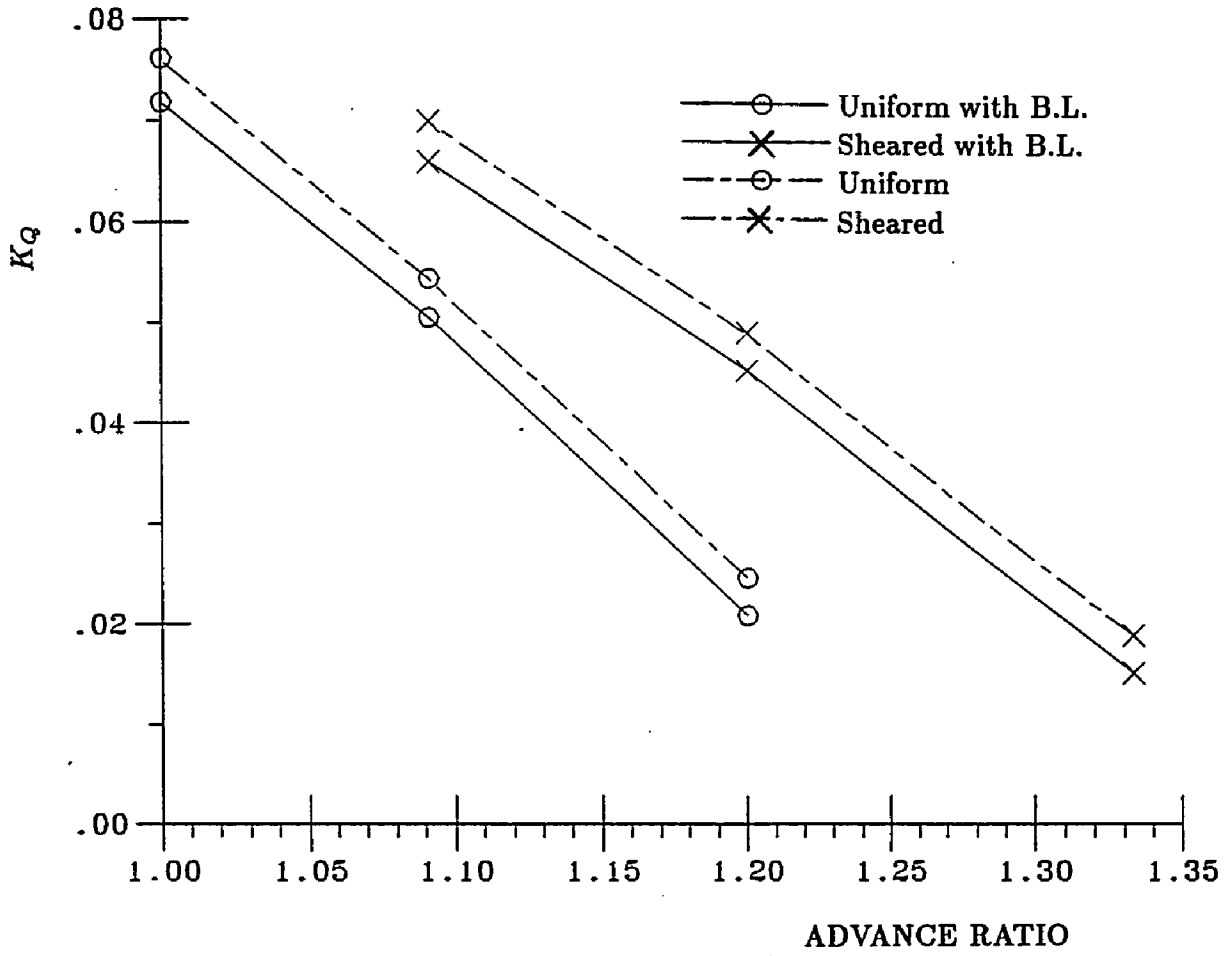

Figure 62: Computed Torque Coefficient of Test Propeller.

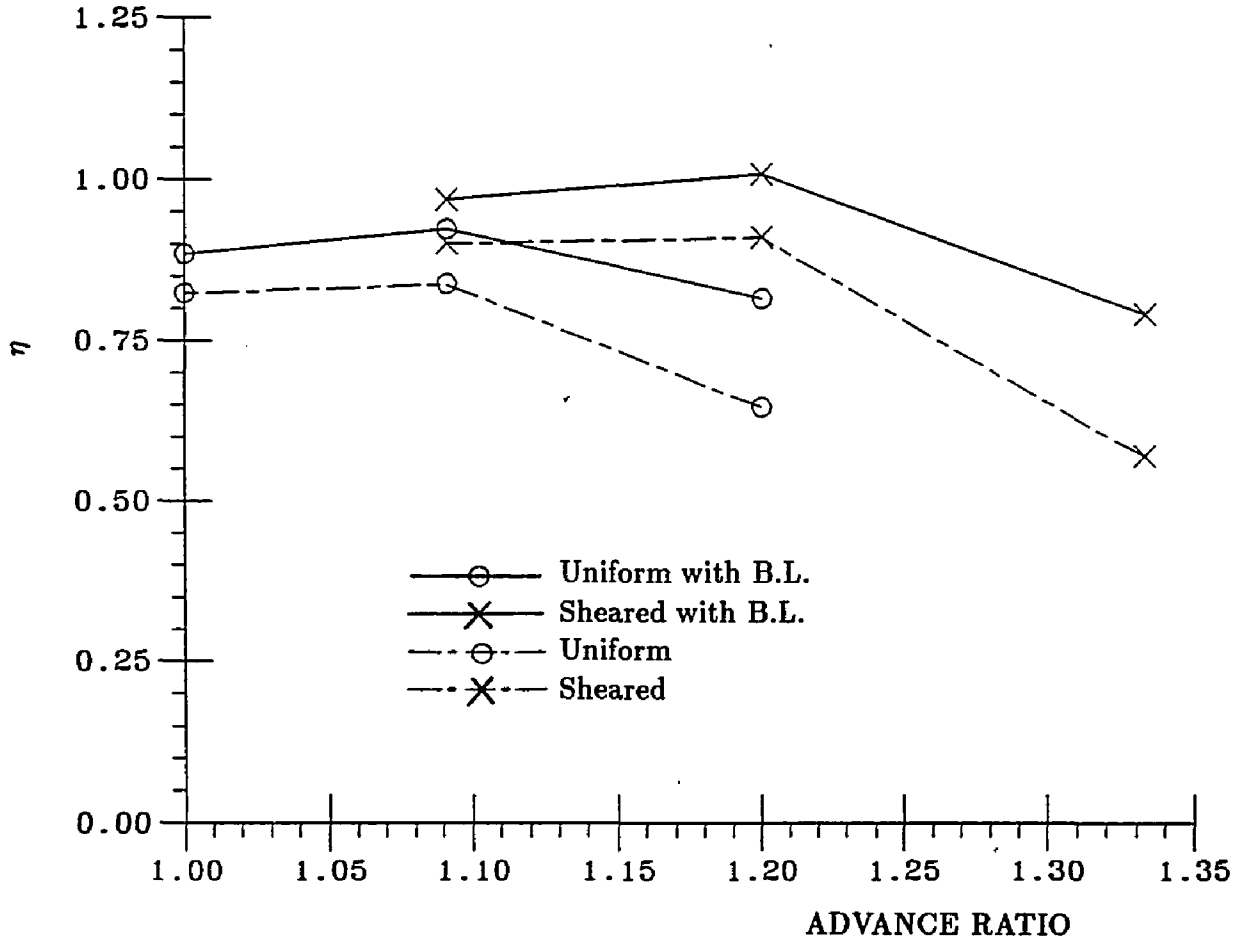

Figure 63: Computed Efficiency of Test Propeller.

where T, Q, and ρ are the thrust, torque, and the density. Both coefficients are relatively high due to large pitch and area. The shroud also contributes to the high coefficients by prohibit communication of the pressure field over the tip. The velocity defect in the sheared inflow alters the angle of attack and significantly increase the thrust and torque level for the same advance ratio.

The effects of the viscous boundary layers are also estimated by adding the drag of a turbulent, flat plate boundary layer, and the estimated coefficients are expressed as the dashed lines. As expected, the boundary layers increase the torque coefficients but decrease the thrust coefficients. The net result is a decrease in the efficiency.

The calculated efficiencies for the irrotational and rotational flows are shown in Fig. 63. The efficiency is defined by

$$\eta = \frac{J}{2\pi}\frac{K_T}{K_Q}$$

As observed, the cases shown in Fig. 63 are around the maximum efficiencies for both types of flows. The designed efficiency is approximately 0.8 for the irrotational flow at advance ratio 1.1 and 0.9 for rotational flow at advance ratio 1.2, both with blade boundary-layers considered. Notice here that for the rotational flow, we define the efficiency by the ship speed not the averaged inflow speed.

To depict the details of the solution, the pressure distribution and the streamlines for selected sections are presented. The cases shown are for an advance ratio of 1.2. Both irrotational and rotational flows are presented for comparison. Figure 64 and 65 show the pressure distribution on the pressure and suction sides scaled by,

$$\frac{1}{2}\rho U_R^2 = \frac{1}{2}\rho\left[U_0^2 + (2\pi nr)^2\right]$$

for irrotational and rotational flows. As can be seen, the rotational flow case gains a larger overall pressure difference between the pressure and suction sides from the

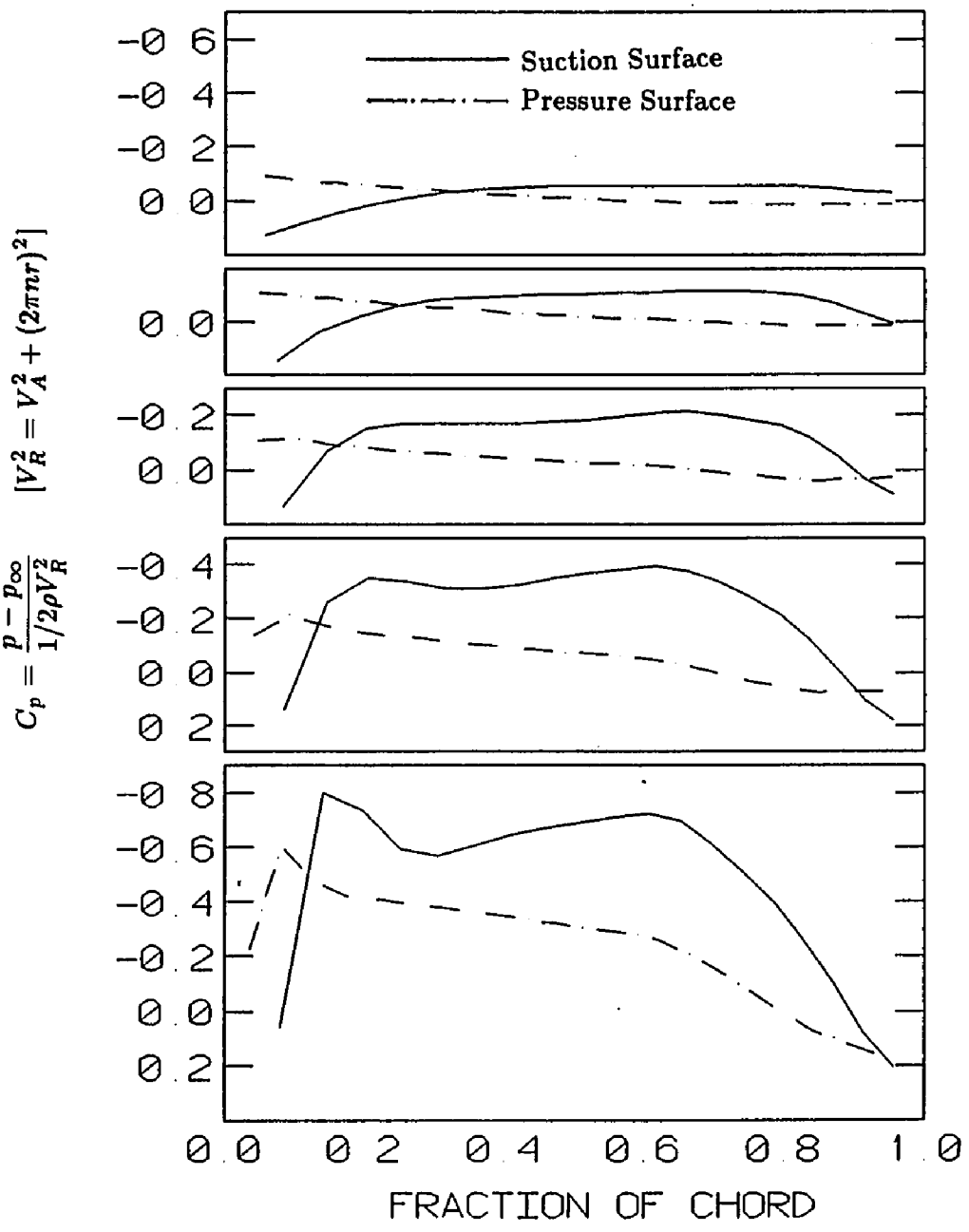

Figure 64: Pressure Distribution of Test Propeller. Uniform Inflow, Advance ratio=1.2.

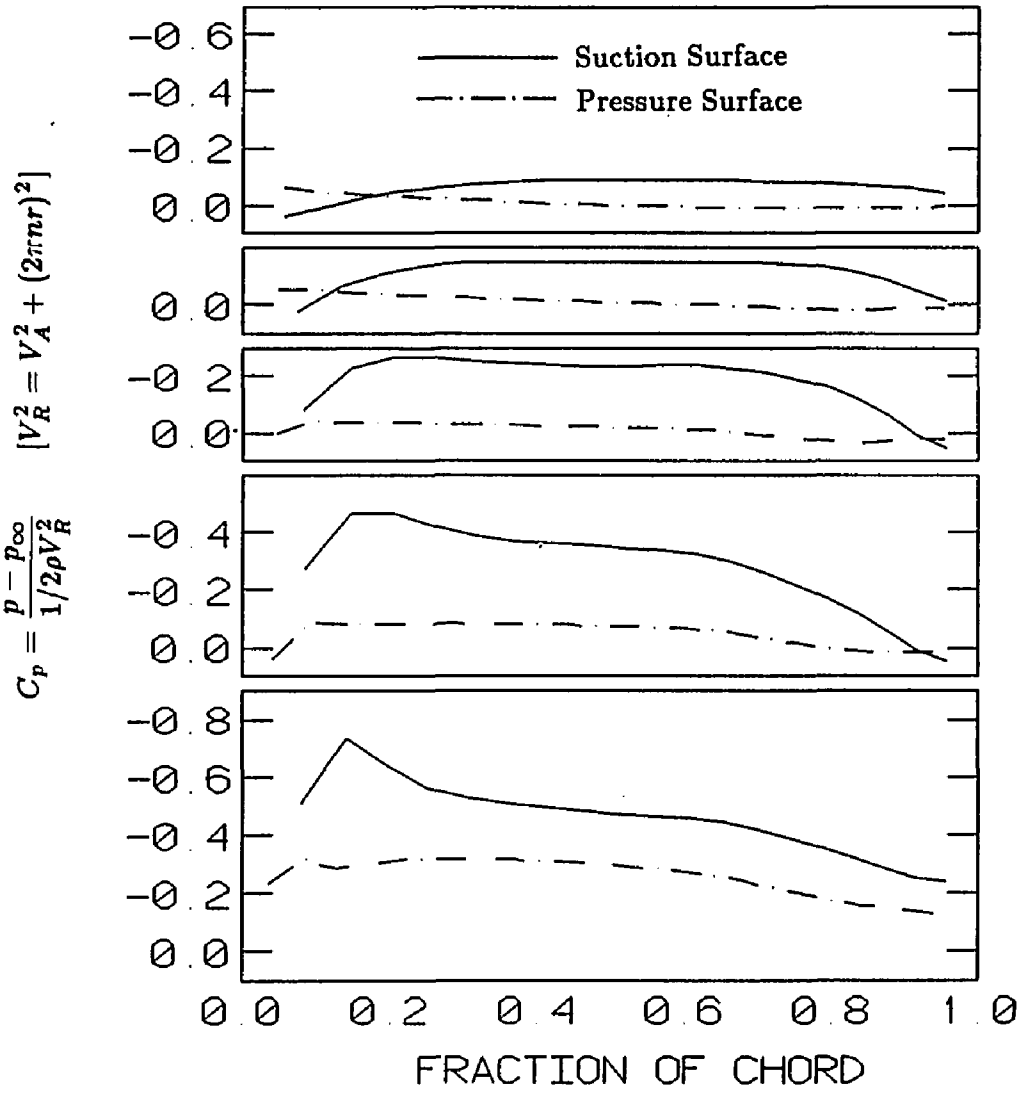

Figure 65: Pressure Distribution of Test Propeller. Sheared Inflow, Advance ratio=1.2.

changes of the angle of attack. However, near the hub section the pressure difference is less because of the smaller inflow velocity. As a result, the patters of the pressure contour for two types of flows appear to be quite different as shown in Fig. 66 and Fig. 67. The pressure contour for the irrotational flow look more ordered with the low pressure zone located on the blade-root. The low pressure region moves up into the blade center because of the smaller inflow velocity on the hub. This difference indicates a significant migration of the load distribution, and can cause an impact on propeller design.

Figure 68 and 69 show projections of the streamlines on an axisymmetric surface for the blade tip-, hub-, and mid-section corresponding to Fig. 66 and 67. Three-dimensional effect can be visualized from some of the streamlines such as those on the hub section of the rotation flow. From these streamlines the downstream vortex sheets can be located precisely. This can be used to benefit current lifting-surface computations. These streamlines also indicate the smoothness of the solution. As observed, the streamlines are quite smooth except around the leading edge where the singular points locate. Local grid refinement and the use of C-mesh in the vicinity of the blades would give improved resolution here.

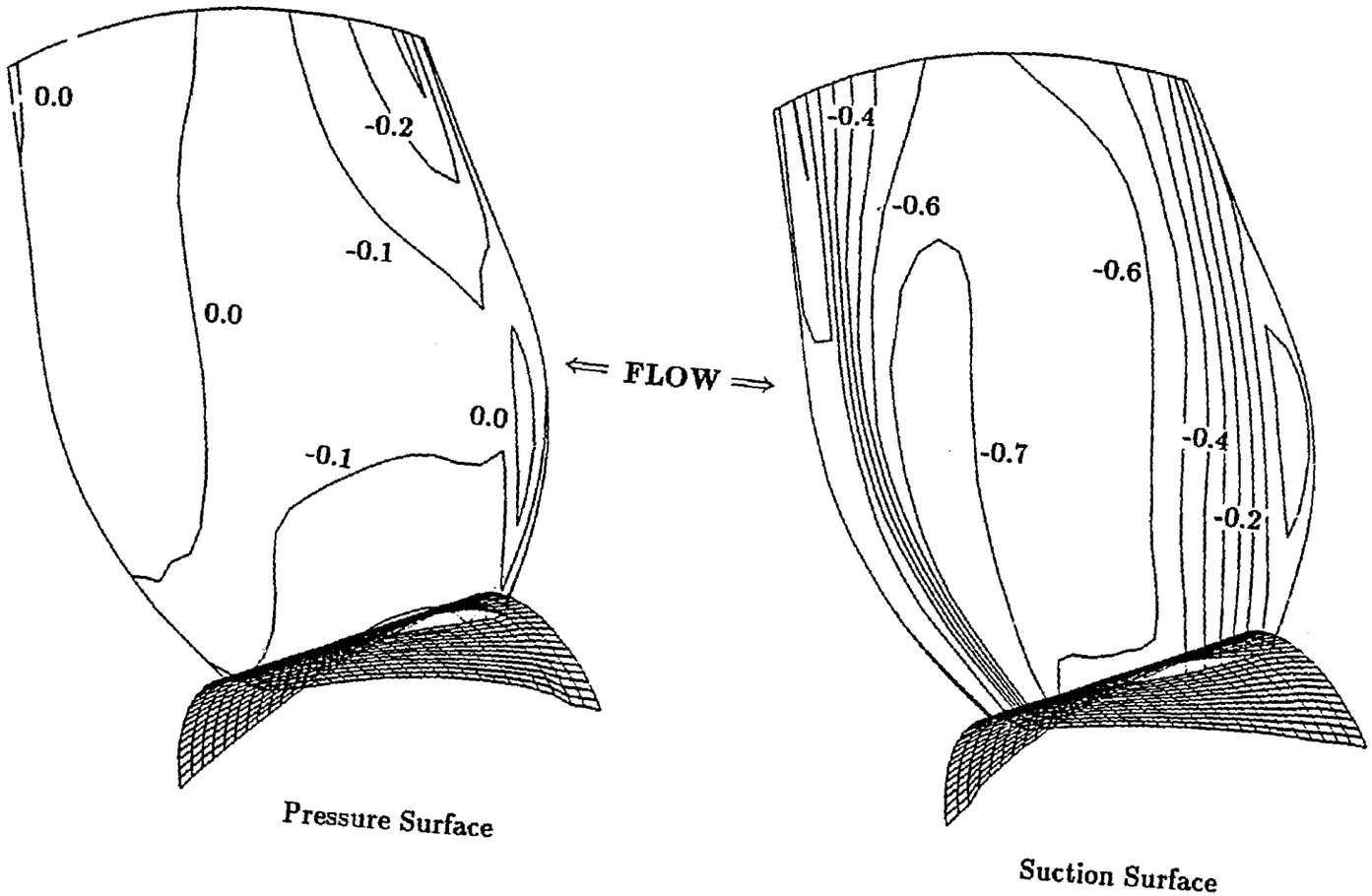

Figure 66: Pressure Contour on Blade Surface. Uniform Inflow with Advance ratio=1.2.

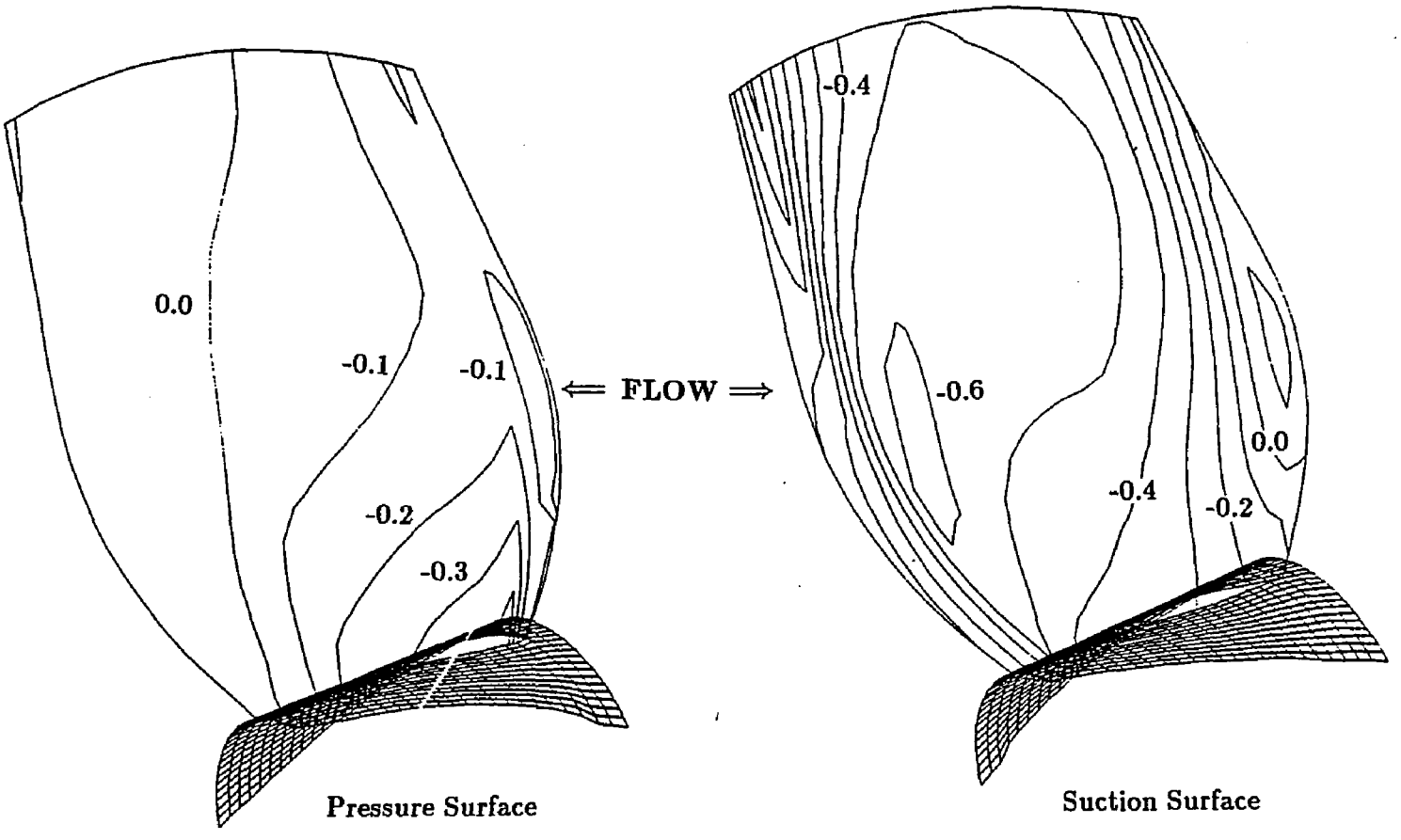

Figure 67: Pressure Contour on Blade Surface. Sheared Inflow with Advance ratio=1.2.

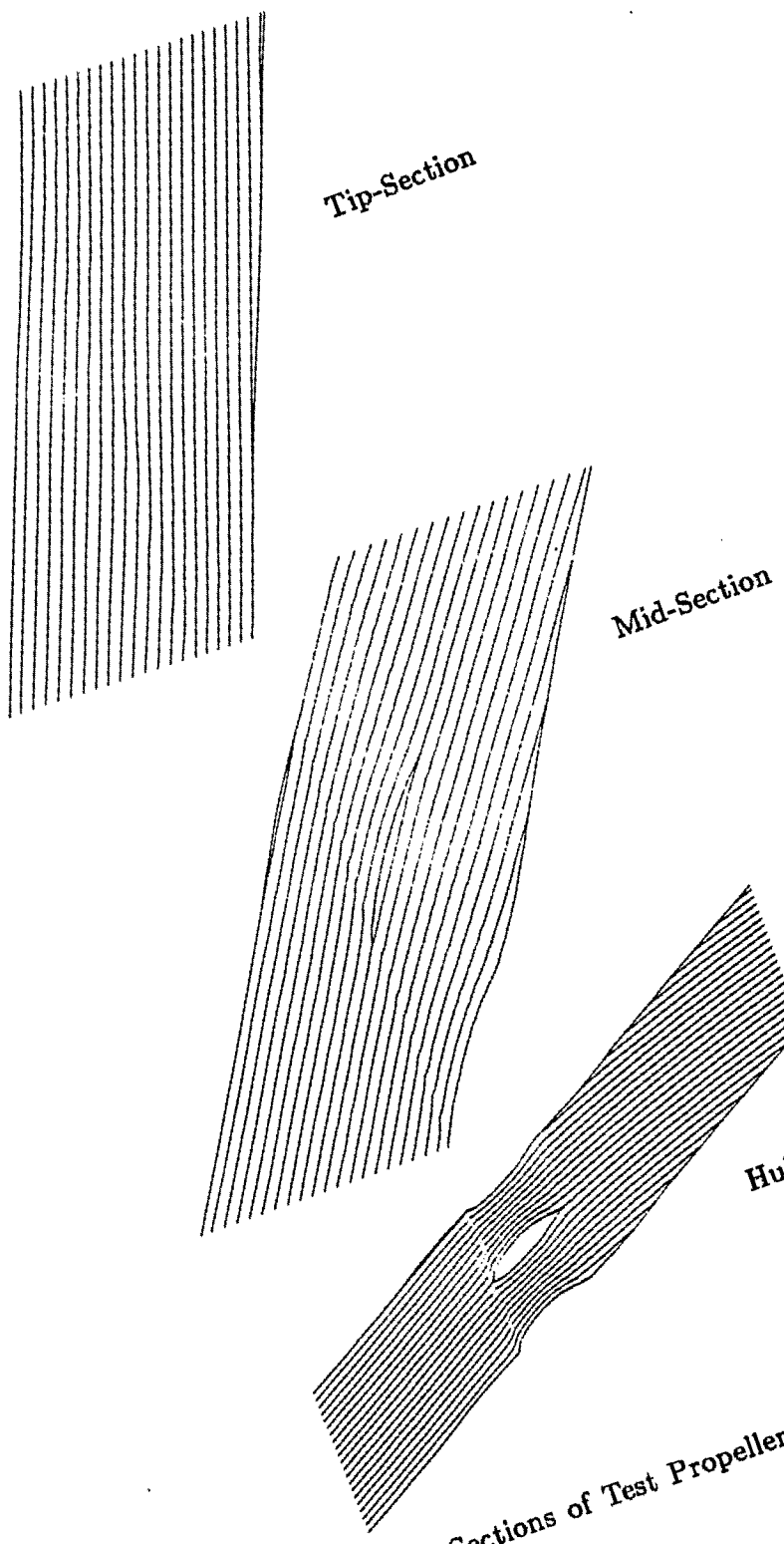

Figure 68: Streamlines of Sections of Test Propeller. Uniform Inflow with Advance ratio=1.2.

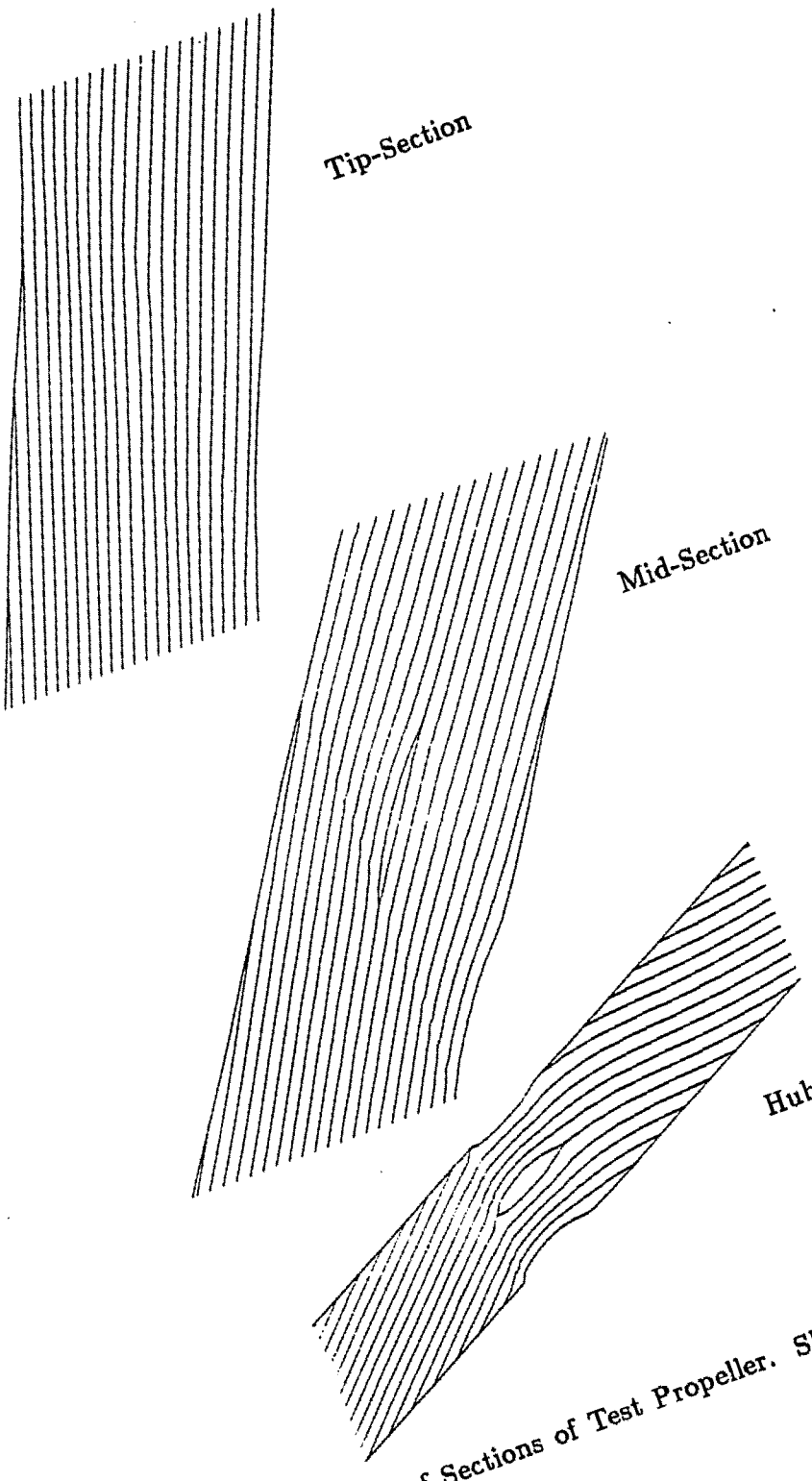

Figure 69: Streamlines of Sections of Test Propeller. Sheared Inflow with Advance ratio=1.2.

CHAPTER 6
SUMMARY

The Euler equations (compressible and incompressible) have been computed successfully by using advanced time-marching schemes such as implicit ADI and explicit Runge-Kutta schemes. Central-difference has been utilized to maintain the second-order accuracy. The numerical difficulty inherent to the parabolic nature of the incompressible system has been overcome by introducing a time derivative of pressure to the continuity equation. The resulting system is hyperbolic in time and is mathematically similar to its compressible-flow counterpart. Both systems are solved by the same time-marching procedures, and has been shown to be well-behaved by numerous computations.

Boundary conditions have been formulated based on the method of characteristics (MOC). The MOC boundary procedure has been shown to be accurate and stable. Another boundary treatment known as the explicit boundary procedure is also presented for comparison. One widely-used set of explicit boundary conditions for compressible flows is addressed in detail. Analogous explicit boundary conditions for incompressible flows are also formulated. The MOC boundary conditions are compared with these explicit boundary conditions in regard to their effect on the convergence and the solution quality. The results show that the MOC boundary conditions work better for both.

The convergence of these schemes is predicted by studying their stability characteristics. Stability analyses show that the Runge-Kutta scheme is conditionally stable with a CFL limit of $2\sqrt{2}$, regardless of the number of dimensions. As predicted by the stability analysis, the Runge-Kutta scheme can be accelerated by using the so-called residual-smoothing method. This acceleration method has been applied to one-, two-, and three-dimensional Runge-Kutta scheme and proven to

be effective. The implicit scheme, on the other hand, experiences dramatic change in its stability characteristics from one to three dimensions. The implicit scheme is unconditionally stable in one dimension; with virtually no limit in the CFL number and the convergence rate. In two dimensions, the stability analysis still predicts the implicit ADI scheme to be unconditionally stable but, suggests the existence of a maximum convergence rate for an optimum CFL number. The implicit ADI scheme becomes conditionally stable in three dimensions due to a larger temporal error. The limiting CFL number is reduced to $\sqrt{4.5}$ as the stability analysis predicts.

The convergence rates of the ADI and the Runge-Kutta schemes have been compared extensively in this work. We have made these comparisons based on number of iterations and CPU-time required on scalar and vector machines. In one dimension, the implicit scheme is overwhelmingly superior to the explicit scheme; the CPU-time cost for the implicit scheme is about one order of magnitude less than that for the explicit scheme. For two-dimensional cases, the implicit scheme is still more efficient. However, the need for approximate-factorization in multi-dimensional implicit scheme greatly reduces its efficiency and the explicit scheme becomes more favorable. In three dimensions, the Runge-Kutta scheme becomes more efficient than the ADI scheme. The restrictive CFL limit and the larger block-size of the three-dimensional ADI scheme give the implicit scheme a major decline in efficiency. Although the Runge-Kutta scheme converges at about the same rate as the ADI scheme, it is less expensive because of its fewer computational counts per step and higher vectorizability.

The application of solving the Euler equations by these schemes is demonstrated by computing flows passing a marine propeller. This is motivated by the need to solve rotational flows in order to understand the physics of the propeller/ship-hull

interaction. Successful calculations have been noted by several runs for different flow conditions on a five-blade propeller. Rotational flows are formed by imposing upstream sheared inflows which simulate the velocity profiles of the ship-hull boundary-layers. The results are compared with irrotational flows for overall performance data and flow details. The comparisons show significant change in the pressure distribution on the blade surfaces caused by the sheared inflow, and suggest that the rotationality should not be ignored.

Besides the ADI and the Runge-Kutta schemes, some substitutional methods such as the diagonal ADI scheme and various LU schemes are also explored. Central-difference is taken to formulate the diagonal ADI scheme, while the upwind-difference of Steger-Warming is utilized to design the implicit operators of the LU schemes. Interest is focused on the efficiencies of these implicit schemes as opposed to the ADI scheme. Comparisons of their efficiencies are made in terms of number of iterations and CPU time on the CYBER-205. The results show that the DDADI scheme, a member of the LU schemes, gives the most rapid convergence based on number of iterations, while the diagonal ADI converges the fastest based on the CPU time. It is also found that some LU schemes can not be vectorized as efficiently as the ADI scheme, but some could be more efficient than the ADI scheme in three dimensions.

BIBLIOGRAPHY

1. Ross, D. (1976). *Mechanics of Underwater Noise*, Paragamon Press, New York.
2. Strasberg, M. (1977). Propeller Cavitation Noise after 35 Years of study, *Proceedings of Symposium on Noise and Fluid Engineering*, ASME Annual Meeting, pp. 89-100.
3. Roache, P. J. (1982). *Computational Fluid Dynamics*, Hermosa Publishers, Albuquerque, New Mexico, pp. 9-10.
4. Harlow, F. H. and Welch, J. E. (1965). Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface, *Phys. Fluids*, vol. 8, pp. 2182-2189.
5. Raithby, G. D. and Schneider, G. E. (1979) Numerical solution of Problems in Incompressible Fluid Flow: Treatment of the Velocity-Pressure Coupling. *Numer. Heat Transfer*, vol. 2, pp. 417-440.
6. Chorin, A. J. (1967). A Numerical Method for Solving Incompressible Viscous Flow Problems, *J. Comp. Phys.*, vol. 2, pp. 12-26.
7. Lax, P. D. and Wendroff, B. (1960). Systems of Conservation Laws, . *Comm. Pure Appl. Math.*, vol. 13, pp. 217-237.
8. Beam, R. M. and Warming, R. F. (1976). An Implicit Finite-Difference Algorithm for Hyperbolic Systems in Conservation Law Form, . *J. Comp. Phys.*, vol. 22, pp. 87-110.
9. Beam, R. M. and Warming, R. F. (1978). An Implicit Factored Scheme for the Compressible Navier-Stokes Equations, *AIAA J.*, vol. 16, pp. 393-401.
10. Jameson, A., Schmidt, W., and Turkel, E. (1981). Numerical Solution of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time Stepping Schemes, AIAA paper 81-1259, AIAA 14th Fluid and Plasma Dynamics Conference, Palo Alto, June 1981.
11. MacCormack, R. W. (1969). The Effect of Viscosity in Hypervelocity Impact Cratering, AIAA paper 69-354, Cincinnati, Ohio.
12. Lax, P. D. (1954). Weak Solutions of Nonlinear Hyperbolic Equations and Their Numerical Computation, *Comm. Pure Appl. Math.*, vol. 7, pp. 159-193.
13. Lax, P. D. and Richtmyer, R. D. (1956). Survey of the Stability of Linear Finite Difference Equations, *Comm. Pure Appl. Math.*, vol. 9, pp. 267-293.
14. Lax, P. D. and Wendroff, B. (1964). Difference Schemes for Hyperbolic Equations with High Order of Accuracy, *Comm. Pure. Appl. Math.*, vol. 17, pp. 381-398.

15. Richtmyer, R. D. and Morton, K. W. (1967). *Finite Difference Methods for Initial Value Problems*, InterScience, New York.

16. Moretti, G. and Abbett, M. (1966). A Time-Dependent Computational Method for Blunt Body Flows, *AIAA J.*, vol. 4, pp. 2136-2141.

17. Briley, W. R. and McDonald, H. (1980). On the Structure of Use of Linearized Block Implicit Schemes, *Journal of Computational Physics*, vol. 34, pp. 54-77.

18. Steger, J. L. and Kutler. P. (1977). Implicit Finite Difference Procedures for the Computation of Vortex Wakes, *AIAA J.*, vol. 2, pp. 12-26.

19. Choi, D. and Merkle, C. L. (1984). Fully Implicit Iterative Solution of Incompressible Flows, Proceeding of the 5th IMACS International Symposium on Computer Methods for Partial Differential Equations, Lehigh University, Pa., pp. 15-22

20. Choi, D. and Merkle, C. L. (1984). Application of Time-Iterative Schemes to Incompressible Flow, AIAA paper 84-1638, AIAA 17th Fluid and Plasma Dynamics Conference, Snowmass, Colorado, June 1984.

21. Chang, J. L. C. and Kwak, D. (1984). On the Method of Pseudo Compressibility for Numerically Solving Incompressible Flows, AIAA paper 84-0252, AIAA 22nd Aerospace Science Meeting, Reno, Nevada, Jan. 1984.

22. Kwak, D. and Chang, J. L. C. and Chakravarthy, S. R. (1984). An Incompressible Navier-Stokes Flow Solver in Three-Dimensional Curvilinear Coordinate System Using Primitive Variables, AIAA paper 84-0253, AIAA 22nd Aerospace Science Meeting, Reno, Nevada, Jan. 1984.

23. Merkle. C. L. and Tsai, Y. L. (1986). Application of Runge-Kutta Schemes to Incompressible Flows, AIAA paper 86-0553, AIAA 24th Aerospace Sciences Meeting, Reno, Nevada, Jan. 1986.

24. Tsai, Y. L., Merkle. C. L., and Huang, T. T. (1988). Euler Equation Analysis of the Propeller-Wake Interaction, 17th Symposium of Naval Hydrodynamics, Holland.

25. Pulliam, T. H. Steger, J. L. (1985). Recent Improvements in Efficiency, Accuracy, and Convergence for Implicit Approximate Factorization Algorithms, AIAA paper 85-0360, AIAA 23rd Aerospace Sciences Meeting, Reno, Nevada, Jan. 1985.

26. Giesing, J. P. (1964). Extension of the Douglas Neumann Program to Problems of Lifting Infinite Cascades, Report No. LB-31653, Douglas Aircraft Company, Revised 2.

27. John, F. (1982). *Partial Differential Equations*, Springer-Verlag, New York, pp. 46-52.

28. Rai, M. M. and Chaussee, D. S. (1983). New Implicit Schemes and Implicit boundary conditions, AIAA paper 83-0123, 21th Aerospace Sciences Meeting, Reno, Nevada, Jan. 1983.

29. Chakravarthy, S. R. (1982). Euler Equations-Implicit Schemes and Implicit Boundary Conditions, *AIAA J.*, vol. 21, pp. 1565-1571.

30. Vichnevetsky, R. and Bowles, J. B. (1982). *Fourier Analysis of Numerical Approximations of Hyperbolic Equations*, SIAM Studies in Applied Mathematics.

31. Smith, G. D. (1978). *Numerical Solution of Partial Differential Equations: Finite Difference Methods*, 2nd edition, Oxford University Press.

32. Courant, R., Friedrichs, K. O., and Lewy, H. (1928). Uber die Partiellen Differenzengleichungen der Mathemaschen Physik, *Mathematische Annalen*, vol. 100, pp. 32-74. (Translated to: On the Partial Difference Equations of Mathematical Physics, IBM J. Res. Dev., vol. 11, pp. 215-234, 1967)

33. Carnahan, B., Luther, H. A., and Wilkes, J. O. (1969). *Applied Numerical Method*, Wiley, pp. 361-366.

34. Jameson, A. and Baker, T. J. (1983). Solution of the Euler Equations for Complex Configurations, *Proc. AIAA 6th Computational Fluid Dynamics Conference*, Danvers, Ma, pp.293-302.

35. Anderson, D. A., Tannehill, J. C., and Pletcher, R. H. (1984). *Computational Fluid Mechanics and Heat Transfer*, McGRAW-HILL, pp. 242-243.

36. Rizzi, A. and Eriksson, L. A. (1984). Computation of Flow around Wings Based on the Euler Equations, *J. Fluid Mech.*, vol. 148, pp. 45-71.

37. Vinokur, M. (1974). Conservation Equations of Gas-Dynamics in Curvilinear Coordinate Systems, *J. Comp. Phys.*, vol. 14, pp. 105-125.

38. Warming, R. F. and Beam, R. M. (1977). On the Construction and Application of Implicit Factored Schemes for Conservation Laws, Symposium on Computational Fluid Dynamics, New York. See *SIAM-AMS Proceedings*, vol. 11, 1978, pp. 85-129.

39. Steger, J. L. (1978). Implicit Finite-Difference Simulation of Flow about Arbitrary Two-Dimensional Geometries, *AIAA J.*, vol. 16, no. 7, July 1978, pp. 679-686.

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

41. Jameson, A. and Mavriplis, D. (1986). Finite Volume Solution of the Two-Dimensional Euler Equations on a Regular Triangular Mesh, *AIAA J.*, vol. 24, no. 4, April 1986, pp. 611-618.

42. Ni, R. H. (1982). A Multiple-Grid Scheme for Solving the Euler Equations, *AIAA J.*, vol. 20, no. 11, Nov. 1982, pp. 1565-1571.

43. Chima, R. V. and Johnson, G. M. (1983). Efficient Solution of the Euler and Navier-Stokes Equations with a Vectorized Mutiple-Grid Algorithm, AIAA paper 83-1893, AIAA 6th Computational Fluid Dynamics Conference, Danvers, Ma., July 1983.

44. Barth, T. J. and Steger, J. L. (1985). A Fast Efficient Implicit Scheme for the Gasdynamic Equations Using a Matrix Reduction Technique, AIAA paper 85-0439, AIAA 23rd Aerospace Sciences Meeting, Reno, Nevada, Jan. 1985.

45. Thompson, J. F., Warsi, Z. U. A., and Mastin, C. W. (1985). *Numerical Grid Generation*, North-Holland, New York.

46. Sorenson, R. L. (1980). A computer Program to Generate Two-Dimensional Grids About Airfoils and Other Shapes by the Use of Poisson's Equations, NASA TM-81198.

47. Rogers, S. E., Chang, J. L. C., and Kwak, D. (1986). A Diagonal Algorithm for the Method of Pseudocompressibility, AIAA paper 86-1060, AIAA/ASME 4th Fluid Mechanics, Plasma dynamics, and Lasers Conference, Atlanta, Ga., May 1986.

48. Young, D. and Gregory, R., A Survey of Numerical Mathematics, vol. 2, pp. 1068, Addison & Wesley.

49. Steger, J. L. and Warming, R. F. (1981). Flux Vector Splitting of the Inviscid Gasdynamic Equations with Application to Finite Difference Methods, *J. Comput. Phys.*, vol. 40, no. 2, 1981, pp.263-293.

50. Obayashi, S. and Kuwahara, K. (1984). LU Factorization of an Implicit Scheme for the Compressible Navier-Stokes Equations, AIAA paper 84-1670.

51. Obayashi, S. and Fujii, K. (1985). Computation of Three-Dimensional Viscous Transonic Flows with the LU Factored Scheme, AIAA paper 85-1510, AIAA 7th Computational Fluid Dynamics Conference, Cincinnati, Ohio, July 1985.

52. Lombard, C. K., Venkatapathy, E., and Bardina, J. (1984). Universal Single Level Implicit Algorithm for Gasdynamics, AIAA paper 84-1533, AIAA 17th Fluid Mechanics, Plasma dynamics, and Lasers Conference, Snowmass, Colorado, June 1984.

53. Choi, D. (1985). Computation of Incompressible Flow by a Coupled Implicit Scheme, Ph.D. dissertation, Department of Mechanical Engineering, Pennsylvania State University, University Park.

54. Bristow, D. R. (1980). Development of Panel Methods of Subsonic Analysis and Design, NASA Report 3234, 1980.

55. Greeley, D. S. and Kerwin, J. E. (1980). Numerical Methods for Propeller Design and Analysis in Steady Flow, SNAME, vol. 90, 1982.

56. Yang, C-I, Jesup, S. D. (1987). Marine Propeller Analysis with Panel Method, AIAA/SAE/ASME/ASEE 23rd Joint Propulsion Conference, 1987.

57. Kerwin, J. E., Kinnas, S. A., Lee, J-T, and Shih, W-Z (1987). A Surface Panel Method for the Hydrodynamic Analysis of Ducted Propellers, Annual Meeting of the SNAME, 1987.

58. Kerwin, J. E. and Lee, C. S. (1978). Prediction of Steady and Unsteady Marine Propeller Performance by Numerical Lifting Surface Theory, SNAME, 1978.

59. Tsaknoas, S., Jacobs, W. R., and Ali, M. R. (1972). An Exact Linear Lifting Surface Theory for a Marine Propeller in a Non-Uniform Flow Field, Davidson Lab., Stevens Institute of Technology, Rep. SIT-DL-72-1509, Feb., 1972.

APPENDIX

THREE-DIMENSIONAL COORDINATE TRANSFORMATION

One conservative system of equations written in the (x, y, z, t) coordinate such as

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{E}}{\partial x} + \frac{\partial \mathbf{F}}{\partial y} + \frac{\partial \mathbf{G}}{\partial z} = \mathbf{H} \tag{A.1}$$

can be transformed to a general coordinate $(\xi, \eta, \varsigma, \tau)$ with the following functional relationship with the original coordinate,

$$\xi = \xi(x, y, z, t), \qquad \eta = \eta(x, y, z, t), \qquad \varsigma = \varsigma(x, y, z, t), \qquad \tau = \tau(t) \tag{A.2}$$

$$x = x(\xi, \eta, \varsigma, \tau), \qquad y = y(\xi, \eta, \varsigma, \tau), \qquad z = z(\xi, \eta, \varsigma, \tau), \qquad t = t(\tau) \tag{A.3}$$

Consider the relation between the derivatives with respect to the independent variables of the two systems, (x, y, z, t) and $(\xi, \eta, \varsigma, \tau)$. By the chain rule,

$$\frac{\partial}{\partial \xi} = x_\xi \frac{\partial}{\partial x} + y_\xi \frac{\partial}{\partial y} + z_\xi \frac{\partial}{\partial z} + t_\xi \frac{\partial}{\partial t} \tag{A.4}$$

where the subscripts represent differentiation. The last term of Eq. (A.4) vanishes because the differentiation is carried out for fixed τ, which is directly related to t. Equation (A.4) and other derivative operators in $(\xi, \eta, \varsigma, \tau)$ system then become

$$\begin{aligned}
\frac{\partial}{\partial \xi} &= x_\xi \frac{\partial}{\partial x} + y_\xi \frac{\partial}{\partial y} + z_\xi \frac{\partial}{\partial z} \\
\frac{\partial}{\partial \eta} &= x_\eta \frac{\partial}{\partial x} + y_\eta \frac{\partial}{\partial y} + z_\eta \frac{\partial}{\partial z} \\
\frac{\partial}{\partial \varsigma} &= x_\varsigma \frac{\partial}{\partial x} + y_\varsigma \frac{\partial}{\partial y} + z_\varsigma \frac{\partial}{\partial z} \\
\frac{\partial}{\partial \tau} &= x_\tau \frac{\partial}{\partial x} + y_\tau \frac{\partial}{\partial y} + z_\tau \frac{\partial}{\partial z} + t_\tau \frac{\partial}{\partial \tau}
\end{aligned} \tag{A.5}$$

Reversely,

$$\begin{aligned}
\frac{\partial}{\partial x} &= \xi_x \frac{\partial}{\partial \xi} + \eta_x \frac{\partial}{\partial \eta} + \varsigma_x \frac{\partial}{\partial \varsigma} \\
\frac{\partial}{\partial y} &= \xi_y \frac{\partial}{\partial \xi} + \eta_y \frac{\partial}{\partial \eta} + \varsigma_y \frac{\partial}{\partial \varsigma} \\
\frac{\partial}{\partial z} &= \xi_z \frac{\partial}{\partial \xi} + \eta_z \frac{\partial}{\partial \eta} + \varsigma_z \frac{\partial}{\partial \varsigma} \\
\frac{\partial}{\partial t} &= \xi_t \frac{\partial}{\partial \xi} + \eta_t \frac{\partial}{\partial \eta} + \varsigma_t \frac{\partial}{\partial \varsigma} + \tau_t \frac{\partial}{\partial \tau}
\end{aligned} \tag{A.6}$$

Applying the operators in Eq. (A.5) to ξ gives

$$\begin{aligned}
1 &= x_\xi \xi_x + y_\xi \xi_y + z_\xi \xi_z \\
0 &= x_\eta \xi_x + y_\eta \xi_y + z_\eta \xi_z \\
0 &= x_\varsigma \xi_x + y_\varsigma \xi_y + z_\varsigma \xi_z \\
0 &= x_\tau \xi_x + y_\tau \xi_y + z_\tau \xi_z + t_\tau \xi_t
\end{aligned} \tag{A.7}$$

The metric coefficients ξ_x, ξ_y, and ξ_z can now be obtained by solving Eq. (A.7). The result is

$$\begin{aligned}
\xi_x &= (y_\eta z_\varsigma - y_\varsigma z_\eta)/J' \\
\xi_y &= (z_\eta x_\varsigma - z_\varsigma x_\eta)/J' \\
\xi_z &= (x_\eta y_\varsigma - x_\varsigma y_\eta)/J'
\end{aligned} \tag{A.8}$$

where J' is inverse of the Jacobian of transformation,

$$J = \partial(\xi, \eta, \varsigma)/\partial(x, y, z) = \frac{1}{\partial(x, y, z)/\partial(\xi, \eta, \varsigma)} = \frac{1}{J'} \tag{A.9}$$

The rest metric coefficients can be obtained in the similar fashion by operating Eq. (A.5) on η, ς, and τ. By the same token, operating Eq. (A.6) on x, y, z, and t gives the reverse metric coefficients. To summarize, the metric coefficients are given by

$$\begin{pmatrix} \xi_x & \xi_y & \xi_z & \xi_t \\ \eta_x & \eta_y & \eta_z & \eta_t \\ \varsigma_x & \varsigma_y & \varsigma_z & \varsigma_t \\ \tau_x & \tau_y & \tau_z & \tau_t \end{pmatrix}$$

$$= \frac{1}{J'} \begin{pmatrix} y_\eta z_\varsigma - y_\varsigma z_\eta & z_\eta x_\varsigma - z_\varsigma x_\eta & x_\eta y_\varsigma - x_\varsigma y_\eta & -\frac{\partial(x,y,z)}{\partial(\eta,\varsigma,\tau)t_\tau} \\ y_\varsigma z_\xi - y_\xi z_\varsigma & z_\varsigma x_\xi - z_\xi x_\varsigma & x_\varsigma y_\xi - x_\xi y_\varsigma & -\frac{\partial(x,y,z)}{\partial(\varsigma,\xi,\tau)t_\tau} \\ y_\xi z_\eta - y_\eta z_\xi & z_\xi x_\eta - z_\eta x_\xi & x_\xi y_\eta - x_\eta y_\xi & -\frac{\partial(x,y,z)}{\partial(\xi,\eta,\tau)t_\tau} \\ 0 & 0 & 0 & \frac{J'}{t_\tau} \end{pmatrix} \qquad (A.10)$$

The reverse metric coefficients are obtained by reversing (x, y, z, t) and $(\xi, \eta, \varsigma, \tau)$. They are also given for completion,

$$\begin{pmatrix} x_\xi & x_\eta & x_\varsigma & x_\tau \\ y_\xi & y_\eta & y_\varsigma & y_\tau \\ z_\xi & z_\eta & z_\varsigma & z_\tau \\ t_\xi & t_\eta & t_\varsigma & t_\tau \end{pmatrix}$$

$$= \frac{1}{J} \begin{pmatrix} \eta_y \varsigma_z - \eta_z \varsigma_y & \varsigma_y \xi_z - \varsigma_z \xi_y & \xi_y \eta_z - \xi_z \eta_y & -\frac{\partial(\xi,\eta,\varsigma)}{\partial(y,z,t)\tau_t} \\ \eta_z \varsigma_x - \eta_x \varsigma_z & \varsigma_z \xi_x - \varsigma_x \xi_z & \xi_z \eta_x - \xi_x \eta_z & -\frac{\partial(\xi,\eta,\varsigma)}{\partial(z,x,t)\tau_t} \\ \eta_x \varsigma_y - \eta_y \varsigma_x & \varsigma_x \xi_y - \varsigma_y \xi_x & \xi_x \eta_y - \xi_y \eta_x & -\frac{\partial(\xi,\eta,\varsigma)}{\partial(x,y,t)\tau_t} \\ 0 & 0 & 0 & \frac{J}{\tau_t} \end{pmatrix} \qquad (A.11)$$

Cyclic property of Eq. (A.10) gives the following useful relations,

$$\frac{\partial}{\partial \xi}\left(\frac{\xi_x}{J}\right) + \frac{\partial}{\partial \eta}\left(\frac{\eta_x}{J}\right) + \frac{\partial}{\partial \varsigma}\left(\frac{\varsigma_x}{J}\right) = 0 \qquad (A.12a)$$

$$\frac{\partial}{\partial \xi}\left(\frac{\xi_y}{J}\right) + \frac{\partial}{\partial \eta}\left(\frac{\eta_y}{J}\right) + \frac{\partial}{\partial \varsigma}\left(\frac{\varsigma_y}{J}\right) = 0 \qquad (A.12b)$$

$$\frac{\partial}{\partial \xi}\left(\frac{\xi_z}{J}\right) + \frac{\partial}{\partial \eta}\left(\frac{\eta_z}{J}\right) + \frac{\partial}{\partial \varsigma}\left(\frac{\varsigma_z}{J}\right) = 0 \qquad (A.12c)$$

$$\frac{\partial}{\partial \xi}\left(\frac{\xi_t}{J}\right) + \frac{\partial}{\partial \eta}\left(\frac{\eta_t}{J}\right) + \frac{\partial}{\partial \varsigma}\left(\frac{\varsigma_t}{J}\right) + \tau_t \frac{\partial}{\partial \tau}\left(\frac{1}{J}\right) = 0 \qquad (A.12d)$$

The system of equations, Eq. (A.1), are ready for transformation. Divide Eq. (A.1) by J, apply the chain rule, and make use of Eqs.(A.12a-d) to get

$$\begin{aligned} &\frac{\partial(\mathbf{Q}\tau_t/J)}{\partial \tau} + \frac{\partial}{\partial \xi}\left(\frac{\mathbf{Q}\xi_t + \mathbf{E}\xi_x + \mathbf{F}\xi_y + \mathbf{G}\xi_z}{J}\right) \\ &\quad + \frac{\partial}{\partial \eta}\left(\frac{\mathbf{Q}\eta_t + \mathbf{E}\eta_x + \mathbf{F}\eta_y + \mathbf{G}\eta_z}{J}\right) \\ &\quad + \frac{\partial}{\partial \varsigma}\left(\frac{\mathbf{Q}\varsigma_t + \mathbf{E}\varsigma_x + \mathbf{F}\varsigma_y + \mathbf{G}\varsigma_z}{J}\right) + \frac{\mathbf{Q}}{J(t_\tau)^2} t_{\tau\tau} = 0 \end{aligned} \qquad (A.13)$$

In practice, $t_{\tau\tau}$ is made zero by specifying a linear functional relation between t and τ. Here we let t equal to τ. By defining transformed dependent variables, flux vectors, and source vector, Eq. (A.13) becomes the governing equation in the (ξ, η, ζ, τ) coordinate system,

$$\frac{\partial \hat{\mathbf{Q}}}{\partial \tau} + \frac{\partial \hat{\mathbf{E}}}{\partial \xi} + \frac{\partial \hat{\mathbf{F}}}{\partial \eta} + \frac{\partial \hat{\mathbf{G}}}{\partial \zeta} = \hat{\mathbf{H}} \tag{A.14}$$

where

$$\hat{\mathbf{Q}} = \frac{1}{J}\mathbf{Q}$$

$$\hat{\mathbf{E}} = \frac{1}{J}\left(\mathbf{E}\xi_x + \mathbf{F}\xi_y + \mathbf{G}\xi_z + \mathbf{Q}\xi_t\right)$$

$$\hat{\mathbf{F}} = \frac{1}{J}\left(\mathbf{E}\eta_x + \mathbf{F}\eta_y + \mathbf{G}\eta_z + \mathbf{Q}\eta_t\right)$$

$$\hat{\mathbf{G}} = \frac{1}{J}\left(\mathbf{E}\zeta_x + \mathbf{F}\zeta_y + \mathbf{G}\zeta_z + \mathbf{Q}\zeta_t\right)$$

$$\hat{\mathbf{H}} = \frac{1}{J}\mathbf{H}$$

VITA

Ying-Liang Peter Tsai was born on April 8, 1959 in Kaohsiung, Taiwan, The Republic of China. He received the degree of Bachelor of Science in Mechanical Engineering in June, 1980 from the National Taiwan University and the Degree of Master of Science in Mechanical Engineering from The Pennsylvania State University in May, 1988.

The author was an Ordnance Officer of the Army of The Republic of China from August, 1980 to May, 1982. Subsequent to his military service, the author worked as a research assistant in the Energy Research Lab. of The Republic of China. He was a graduate assistant in the Department of Mechanical Engineering of the Pennsylvania State University from August, 1983 to May, 1988.